DrivAerNet++: A Large-Scale Multimodal Car Dataset with Computational Fluid Dynamics Simulations and Deep Learning Benchmarks

Mohamed Elrefaie[*]
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
mohamed.elrefaie@mit.edu

Florin Morar
Morphing and Optimization Solutions
BETA CAE SYSTEMS USA, Inc
Farmington Hills, MI 48334 USA
fmorar@beta-cae.us

Angela Dai
Department of Computer Science
Technical University of Munich
Garching, 85748 Germany
angela.dai@tum.de

Faez Ahmed
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
faez@mit.edu

Abstract

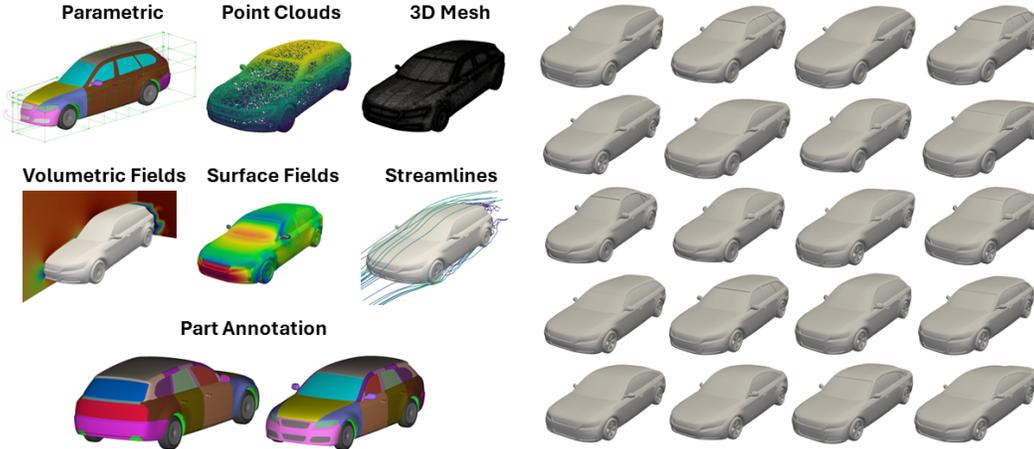
We present DrivAerNet++, the largest and most comprehensive multimodal dataset for aerodynamic car design. DrivAerNet++ comprises 8,000 diverse car designs modeled with high-fidelity computational fluid dynamics (CFD) simulations. The dataset includes diverse car configurations such as fastback, notchback, and estateback, with different underbody and wheel designs to represent both internal combustion engines and electric vehicles. Each entry in the dataset features detailed 3D meshes, parametric models, aerodynamic coefficients, and extensive flow and surface field data, along with segmented parts for car classification and point cloud data. This dataset supports a wide array of machine learning applications including data-driven design optimization, generative modeling, surrogate model training, CFD simulation acceleration, and geometric classification. With more than 39 TB of publicly available engineering data, DrivAerNet++ fills a significant gap in available resources, providing high-quality, diverse data to enhance model training, promote generalization, and accelerate automotive design processes. Along with rigorous dataset validation, we also provide ML benchmarking results on the task of aerodynamic drag prediction, showcasing the breadth of applications supported by our dataset. This dataset is set to significantly impact automotive design and broader engineering disciplines by fostering innovation and improving the fidelity of aerodynamic evaluations. Dataset and code available at: `https://github.com/Mohamedelrefaie/DrivAerNet`

1 Introduction

Car design is a complex and iterative process requiring close collaboration between designers, who focus on aesthetics, and engineers, who ensure the design meets performance constraints. One of the key challenges is achieving a balance between aesthetic appeal and aerodynamic efficiency, which directly impacts fuel consumption. With stricter fuel consumption regulations for internal combustion engine (ICE) cars and increased range requirements for battery-powered electric vehicles

---

[*]Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.

(a) Data modalities of DrivAerNet++. Top row: different data representations; middle row: CFD simulation results; bottom row: annotated car components.

(b) Selected samples from DrivAerNet++ showing diversity in shape with different car designs (fastback, estateback, and notchback), wheels configurations, and underbody configurations.

Figure 1: Data modalities and shape variations in the DrivAerNet++ dataset.

(BEVs) [48, 9, 46], ensuring efficient car aerodynamics has become crucial. As a result, there is significant interest in developing machine learning methods for modeling car aerodynamics.

Data-driven approaches can significantly shorten the process needed before obtaining performance estimates, which typically involves generating the 3D mesh, ensuring watertightness and simulation readiness, performing CFD meshing, defining the solver and boundary conditions, running the CFD, and postprocessing the results. By streamlining these steps, data-driven methods enhance efficiency and expedite the design process. This enables designers to explore various ideas with real-time, accurate performance estimates, ultimately enhancing outcomes with greater design freedom. Recent advances in geometric deep learning methods [59, 58, 57, 1, 41, 68, 63, 7] have demonstrated their ability to estimate performance values from CFD rapidly, facilitating interactive design modifications. However, these methods are often restricted to simple problems due to a lack of public datasets, which limits their broader applicability.

Existing datasets frequently focus on simpler 2D cases [8, 66, 23, 6, 70, 41, 31] or simplified 3D models [7, 57, 45, 69, 60, 63, 68], often excluding critical components like wheels, mirrors, and underbodies. As highlighted by [34], including these elements significantly affects aerodynamic performance, resulting in a notable increase in drag. This was evidenced by increases in the drag value of approximately 142% in CFD simulations and 120% in wind tunnel experiments. This underscores the crucial role of comprehensive 3D modeling in achieving accurate aerodynamic assessments. In addition, around 25% of a passenger car's aerodynamic drag is attributed directly or indirectly to its wheels [10]. Furthermore, many large-scale datasets [63, 69] lack experimental validation of the baseline simulation with physical wind tunnel tests and validation of the individual convergence of each simulation.

There is a notable lack of public, large-scale, and multimodal car datasets, which hinders progress in data-driven design. This stands in contrast to other fields, where standardized datasets such as ImageNet [19], ObjectNet3D [75], ModelNet [74], and ScanNet [17] have driven significant advancements.

We address these challenges with the DrivAerNet++[2] dataset (see Figure 1). The DrivAerNet++ dataset represents a significant advancement over its predecessor, the original DrivAerNet dataset [24] with the integration of 4,000 diverse car shapes to its collection. This enhancement doubles the dataset's volume to a total of 8,000 industry-standard car designs and notably elevates the simulation fidelity with a more complex cell structure (24M cells, as opposed to 8-16M in the original dataset). Furthermore, DrivAerNet++ expands its utility by incorporating detailed 3D flow field data, parametric data, aerodynamic performance coefficients, and part annotations. The dataset encompasses a

---

[2]https://github.com/Mohamedelrefaie/DrivAerNet

2

wide range of geometries and configurations, covering most conventional car design categories and including both detailed underbodies for traditional ICE cars and smooth underbodies for electric cars.

2 Related Work

Large-scale, diverse, and high-fidelity datasets are essential for advancing deep learning methods for CFD and engineering design, offering standardized data that aid in the development, validation, and comparison of new methodologies. Emerging datasets like AirfRANS [8], BubbleML [32], Lagrangebench [67], and BLASTNet [14] have significantly contributed to the machine learning community in fluid mechanics by providing comprehensive data for training and benchmarking. In engineering design, datasets such as AircraftVerse [15] offer detailed and diverse configurations of aerial vehicle designs, helping engineers validate new design strategies and ensure high performance. However, there still exists no large-scale dataset for 3D shapes that combines both high-fidelity CFD simulations with engineering design specifically tailored for car aerodynamic design.

Table 1: A comprehensive comparison of various aerodynamics datasets, highlighting key aspects such as dataset size, the inclusion of aerodynamic coefficients (C_d, C_l), velocity (**u**), pressure (p), wall-shear stress (τ_w) fields, wheels/underbody modeling, parametric study capability, number of design parameters, shape variation, experimental validation data, multimodality, and open-source availability. *M* refers to 3D meshes, *PC* to point clouds, *P* to parametric data, *A* to part annotations, and *C* to CFD data.[†]Dataset is based on ShapeNet [12]. [*]There exists experimental data from wind tunnel tests for validation.

Dataset	Size	Aerodynamics Data					Wheels/Underbody Modeling	Parametric	Design Parameters	Shape Variation	Experimental Validation	Modalities	Open-source
		C_d	C_l	**u**	p	τ_w							
Usama et. al 2021 [70]	500	✔	✗	✗	✗	✗	✗	✔	40 (2D)	✗	✗	P	✗
Li et. al 2023 [45][*]	551	✔	✗	✗	✔	✔	✗	✔	6 (3D)	✗	✗	M,P,C	✗
Rios et. al 2021 [60][†]	600	✔	✔	✗	✗	✗	✗	✗	-	✗	✗	M,PC	✗
Li et. al 2023 [45][†]	611	✔	✗	✗	✔	✔	✗	✗	-	✔	✗	M,C	✗
Umetani et. al 2018 [69][†]	889	✔	✗	✔	✔	✗	✗	✗	-	✔	✗	M,C	✔
Gunpinar et. al 2019 [31]	1,000	✔	✗	✗	✗	✗	✗	✔	21 (2D)	✗	✗	P	✗
Jacob et. al 2021 [39][*]	1,000	✔	✔	✔	✗	✔	✔	✔	15 (3D)	✗	✔	M,C,P	✗
Trinh et. al 2024 [68]	1,121	✗	✗	✔	✔	✗	✗	✗	-	✗	✗	M,C	✗
Remelli et. al 2020 [57][†]	1,400	✗	✗	✗	✔	✗	✗	✗	-	✔	✗	M,C	✗
Baque el al. 2018 [7]	2,000	✔	✗	✗	✗	✗	✗	✔	21 (3D)	✗	✗	M,P	✗
Song et. al 2023 [63][†]	2,474	✔	✗	✗	✗	✗	✗	✗	-	✔	✗	M	✔
DrivAerNet [24][*]	4,000	✔	✔	✔	✔	✔	✔	✔	50 (3D)	✗	✔	M,PC,C,P	✔
DrivAerNet++ (Ours)[*]	8,000	✔	✔	✔	✔	✔	✔	✔	26-50 (3D)	✔	✔	M,PC,C,P,A	✔

The comparison presented in Table 1 supports our motivation by highlighting the lack of open-source datasets that encompass a comprehensive range of features for data-driven aerodynamic design. This gap underscores the necessity for datasets that not only provide high-fidelity simulations but also ensure experimental validation to confirm the accuracy and reliability of the computational models. DrivAerNet++ addresses these needs by including multiple data modalities (3D meshes, point clouds, CFD data, parametric data, and part annotations), and considers the modeling of rotating wheels and underbody. While DrivAerNet [24] was based on a single car category, DrivAerNet++ incorporates a variety of car designs and categories.

In the following, we highlight the limitations of the existing datasets:

- **Lack of diversity:** The datasets from [39, 45, 24, 7, 22, 62] are based on the same parametric models, resulting in generated cars that stem from the same car designs. This lack of diversity limits the generalization capabilities and creativity in design exploration.

- **Small dataset size:** In the engineering design process, changes are not limited to simple geometric parameter adjustments but often involve adding or removing entire components. A significant limitation is the availability of high-quality, watertight meshes necessary for CFD

3

simulations. Most existing datasets [45, 58, 69, 22] are either based on ShapeNet [12], which contains very few car designs suitable for CFD and features meshes with low resolutions compared to what is typically used in academia or industry for car aerodynamic design, or are based on morphed geometries from single designs like the Ahmed body [2] or DrivAer[3] body. Consequently, most existing datasets are relatively small, typically on the order of hundreds, with the largest being [63] with 2,474 cars.

- **Lower simulation fidelity:** Due to the expensive computational cost of running high-fidelity CFD simulations, there is a trade-off between dataset size and simulation fidelity. As a result, existing datasets, such as [63, 7, 57, 58, 70, 31], used low simulation fidelity, which reduces practical utility.

Our dataset, DrivAerNet++, attempts to provide both design variations and diversity, and simulation fidelity, making it highly suitable for conceptual design stages. This balance ensures that designers can explore a wide range of aerodynamic concepts without sacrificing the quality of the simulations.

3 Dataset Presentation

Baseline geometry generation   In automotive aerodynamics, production cars are typically classified into three categories based on the airflow patterns at their rear end [35, 36]: estateback, fastback, and notchback cars. To ensure our dataset covers the entire design space of most conventional car designs, we created multiple parametric models with different designs based on the DrivAer model [49]. This includes various rear configurations—fastback, estateback, and notchback—each leading to different wake structures and flow field patterns. Additionally, we have varied the wheels, incorporating both open and closed designs, as well as smooth and detailed options. For the car underbody, we included both detailed underbodies typical for ICE cars and smooth underbodies suitable for electric cars (see Figure 2). By exploring various rear, wheels, and underbody configurations, we aim to provide a comprehensive understanding of their aerodynamic impacts, thus supporting the development of more robust and generalizable deep learning models. For the creation of the parametric models, we utilized the commercial software ANSA® to define 26 geometric parameters, allowing us to morph these parametric models, resulting in a large-scale dataset of 3D cars. Our goal was to develop a procedural

Figure 2: Baseline models from which the parametric models of DrivAerNet++ are derived, demonstrating a range of shape designs and configurations. Variations include estateback, fastback, and notchback car body types alongside different underbody configurations such as smooth and detailed. Wheel options are presented with closed, open, detailed, and smooth styles.

generator that creates topologically valid car designs, ensuring each design meets the necessary requirements for evaluation by CFD solvers and usability by automotive designers.

High-resolution 3D industry-standard designs   Our design curation involved selecting valid car configurations followed by detailed CFD simulations to evaluate aerodynamic performance. We aimed to create a balanced dataset that encompasses a wide variety of car designs, ensuring coverage of diverse aerodynamic performance metrics and aesthetic considerations. Figure 3 shows a subset of the design parameters used for the generation of the DrivAerNet++ dataset. By defining a lower and upper bound for each parameter and morphing the baseline parametric model, we ensure a comprehensive and well-defined representation suitable for engineering applications and simulations. Our design methodology is diversity-preserving, ensuring that optimization does not result in overly similar designs. To achieve this, we employed Optimal Latin Hypercube sampling for the Design of Experiments (DoE). Specifically, we used the Enhanced Stochastic Evolutionary Algorithm

---

[3]The DrivAer model [35, 49] is a well-established conventional car reference model developed by researchers at the Technical University of Munich (TUM). It combines features of BMW 3 Series and Audi A4 designs, bridging the gap between simplified models and complex proprietary designs.

4

(ESE) [18] to ensure efficient sampling of the design space. Using these steps, DrivAerNet++ is significantly more diverse than DrivAerNet [24].

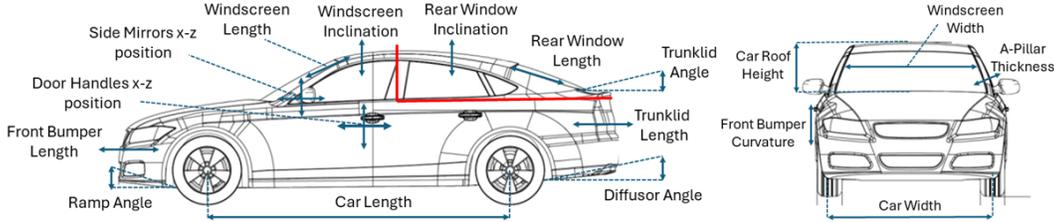



Figure 3: Design parameters for the generation of the DrivAerNet++ dataset. Several geometric parameters with significant impact on aerodynamics were selected and varied within a specific range. These parameter ranges were chosen to avoid values that are either difficult to manufacture or not aesthetically pleasing. The car sketch is adapted from [34].

Comparison of mesh resolutions across large-scale datasets   Here we compare the mesh resolution on car surfaces across different existing large-scale datasets. The results are summarized in Table 2. The mesh resolution comparison highlights the varying levels of detail used in different large-scale datasets for aerodynamic simulations. DrivAerNet++ (ours) provides the highest mesh resolution, ranging from 350,000 to 750,000 cells, ensuring detailed representation of car surfaces for accurate simulations. In contrast, other datasets like the Ahmed body by Li et al. (2023) [45] and ShapeNet (Song et al., 2023 [63]) use lower resolutions, with 100,000 cells and 30,000–50,000 cells, respectively. This finer mesh resolution in DrivAerNet++ enables better fidelity in aerodynamic analyses, particularly for capturing intricate details crucial for engineering design.

Table 2: Comparison of mesh resolutions on car surfaces in large-scale datasets.

Dataset	Mesh Resolution on Car Surface
DrivAerNet++ (Ours)	350k to 750k cells
Ahmed body (Li et al., 2023) [45]	100k cells
ShapeNet (Song et al., 2023) [63]	30–50k cells
ShapeNet (Umetani et al., 2018) [69]	10k cells

CFD mesh generation   For the mesh generation, we utilized the open-source SnappyHexMesh utility [50]. Following the best meshing practices [33, 56, 34], we ensured that our meshes accurately model boundary layer interactions. Each mesh consists of 24 million cells in total, with 500-750k cells specifically dedicated to the car surface, ensuring detailed meshing for the car body and wheels to accurately capture the necessary aerodynamic phenomena. For comparison, the largest dataset after DrivAerNet [24] and DrivAerNet++, introduced by [63], which has 2474 car designs, used about 2 million cells for each CFD simulation. All technical details regarding the meshing process and validation are provided in the supplementary.

Automated high-fidelity CFD simulations   We employed the open-source software OpenFOAM® v11 [30] to conduct steady-state incompressible simulations using the k-ω SST turbulence model, based on Menter's formulation [47]. We ran quality checks for the generated geometries to ensure they were simulation-ready and correctly aligned within the CFD domain, followed by quality checks for the CFD meshing, and finally, checks to ensure the convergence of each CFD simulation. In total, we generated an additional 4,000 simulations to our recently published DrivAerNet dataset [24], encompassing various designs, flow behaviors, turbulence, and separation phenomena.

Computational cost   Running the high-fidelity CFD simulations for DrivAerNet++ required substantial computational resources. The simulations were conducted on the MIT Supercloud, leveraging parallelization across 60 nodes, totaling 2880 CPU cores, with each CFD case using 256 cores and 1000 GBs of memory. The full dataset requires **39 TB** of storage space, and the total number of files for the CFD simulations amounts to 834,332. The job parallelization was managed using MPI [20], ensuring efficient distribution of computational tasks. The simulations took approximately 3×10^6 **CPU-hours** to complete.

5

Dataset structure   Our dataset represents car designs using multiple modalities to ensure comprehensive coverage and ease of use in various applications.

- 3D Car Designs: We provide 3D STL meshes ideal for engineering design, CFD analysis, design optimization, and generative AI applications.
- ANSA® 3D Parametric Models: Provided to enable researchers to generate their own datasets and incorporate custom parameters as needed for specific research objectives.
- Tabular Parametric Data: Each 3D car geometry is parametrized with 26 parameters that completely describe the design. This data can be used for parametric regression, classification, and design feature importance analysis.
- Aerodynamic Performance Data: Includes force coefficients such as drag C_d, lift (total C_l, rear $C_{l,r}$, front $C_{l,f}$), and moment C_m values, both mean and standard deviation.
- CFD Data: Includes raw and post-processed data, 3D full flow field information with velocity and pressure, surface fields with pressure and wall shear stress, streamlines around the car body, and 2D slices.
- Point Cloud: Representations uniformly sampled over the car surface in 5k, 10k, 100k, 250k, and 500k nodes.
- Annotated Labels: For both car designs and segmentation of different car components of each individual car, useful for tasks such as object detection, semantic segmentation, parametric studies, and automated CFD meshing.

Figure 4 shows a comparative analysis of aerodynamic performance across various design configurations and categories, highlighting the diversity and size of our dataset.

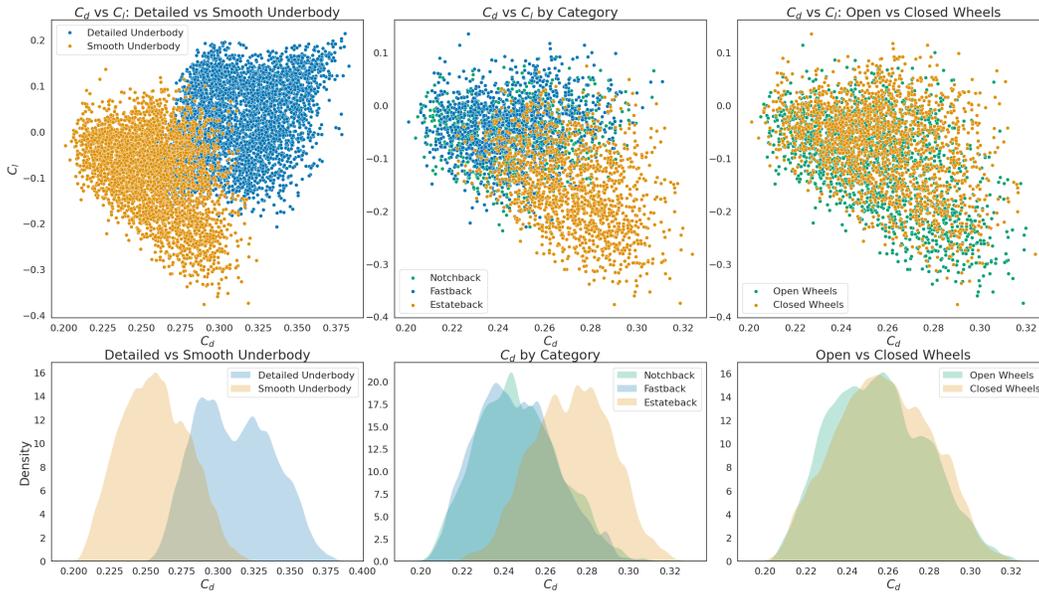



Figure 4: The scatter plots in the top row illustrate the relationship between C_d and C_l for different configurations: the first plot shows the influence of underbody configurations, comparing detailed versus smooth underbodies typically used in electric cars. The second plot highlights the impact of design aesthetics and style across car categories (notchback, fastback, and estateback). The third plot examines the effect of different wheel configurations, emphasizing the significance of small geometric modifications on aerodynamics. The density plots in the bottom row show the distribution of C_d for the same configurations, providing a detailed view of how these design elements and categories influence aerodynamic efficiency.

We provide detailed metadata for the DrivAerNet++ dataset using the Croissant format [3] to ensure comprehensive documentation and ease of use for the research community. We also include datasheets for datasets [29], and the dataset is provided under the Creative Commons Attribution-NonCommercial (CC BY-NC) license. DrivAerNet++ will be hosted on the Harvard Dataverse

6

Repository to ensure optimal accessibility and systematic data management. Since the 39 TB of data might pose challenges for data sharing and access, we also provide subsets of our dataset tailored for different tasks, with detailed metadata included to facilitate usability.

Scaling up the dataset size vs simulation fidelity   Scaling the dataset size in engineering design and CFD poses significant challenges due to the high computational and storage requirements. Despite these challenges, we argue that our simulation fidelity, mesh size, and the use of the k-ω SST turbulence model are sufficient for the initial design stages. The results shown in Section 5.1.2 and in our previous work [24] highlight the importance of scaling the training dataset size for better generalization to unseen new designs. While hybrid RANS-LES methods with mesh sizes of $\mathcal{O}(100M)$ [4, 5, 22] cells provide more accurate turbulence modeling, their computational cost and storage demands are prohibitive. The k-ω SST model offers a balanced approach that supports the scaling of training dataset sizes for deep learning models. Our dataset surpasses all open-source datasets and most literature in terms of simulation fidelity and size, enabling good coverage of the design space and providing insights into various design strategies.

Leveraging multi-fidelity data and transfer learning for efficient surrogate model development has shown promising results, as demonstrated by [25] for flow field prediction and by [65] for multi-fidelity shape optimization. Utilizing large-scale and diverse datasets like DrivAerNet++ facilitates a comprehensive examination of the design space. Subsequent fine-tuning on higher fidelity datasets from hybrid RANS-LES simulations can further improve predictive capabilities. This approach mirrors strategies in the broader deep learning community, demonstrating the synergy between data scale and model refinement for significant performance enhancements.

4   Benchmarking Setup

In this paper, we explore various machine learning tasks, specifically focusing on surrogate modeling (regression) of aerodynamic drag (C_d). This study is distinctive as it is the first to benchmark diverse models using a large-scale, diverse, and high-fidelity dataset. While previous research [24, 39, 45, 7, 69, 70] typically limits comparisons to a single car design or category, our approach, in contrast, enables fair and generalized comparisons across models by employing a comprehensive and publicly accessible dataset, showcasing real-world applicability and performance in automotive aerodynamics.

Metrics and visualization   We assess different models' performance using several metrics: Mean Squared Error (MSE) quantifies the average squared differences between predicted and actual values, making it sensitive to large errors. Mean Absolute Error (MAE) measures the average magnitude of errors and is less affected by outliers. Maximum Absolute Error (Max AE) identifies the largest prediction error, indicating worst-case accuracy. The Coefficient of Determination (R^2 Score) represents the proportion of variance explained by the model, with a value of 1 indicating a perfect fit. Lower MSE and MAE values, along with a higher R^2 score and a lower Max AE, indicate more accurate predictions. For C_d estimation, an MAE of less than 0.005 compared to wind tunnel measurements is considered acceptable [61, 39].

5   Benchmarking Results

5.1   Surrogate modeling of the aerodynamic drag

In the conceptual and initial phases of car design, the aerodynamic drag coefficient is a critical metric as it indicates design efficiency and impacts driving range. Therefore, having accurate and rapid drag estimates is crucial in the design process. In this section, we introduce two approaches for aerodynamic drag prediction: first, using 3D geometric deep learning with 3D meshes, and second, using automated machine learning based on the parametric dataset.

5.1.1   Aerodynamic drag prediction from 3D meshes using geometric deep learning

Here, we test different geometric deep learning models (PointNet [55], GCNN [44], and RegDGCNN [24]) implemented in PyTorch [51] and PyTorch Geometric [27] for the task of surrogate modeling of aerodynamic drag, highlighting the importance of dataset diversity and scaling. Specifically, we train models using different representations, including graph-based and point cloud-based

7

models. A different approach from previous studies [39, 22, 45, 7] is taken by training the models on

Table 3: Comparative analysis of deep learning models for aerodynamic drag prediction on the test set from the DrivAerNet [24] dataset (fastback design with open wheels, with mirrors, and with detailed underbody configuration) comprising 600 car designs.

Model	MSE ↓ ($\times 10^{-5}$)	MAE ↓ ($\times 10^{-3}$)	Max AE ↓ ($\times 10^{-3}$)	R^2 ↑	Training Time ↓	Inference Time ↓	Number of Parameters
PointNet [55]	12.0	8.85	10.18	0.826	0.5hrs	0.51s	2,348,097
GCNN [44]	10.7	7.17	10.97	0.874	10.4hrs	20.71s	100,481
RegDGCNN [24]	8.01	6.91	8.80	0.901	3.2hrs	0.52s	3,164,257

a single car design (fastback) and, in another experiment, on all designs (fastback, notchback, and estateback). First, we train the deep learning models on the DrivAerNet dataset [24], which includes variants of the fastback with detailed underbody, open wheels, and mirrors. This dataset comprises 4,000 car designs (2,800 for training, approximately 600 for validation, and 600 for testing), with results presented in Table 3. Then, we train and test the same models on the DrivAerNet++ dataset, which has 8,000 car designs with extensive variations (fastback, estateback, notchback, smooth and detailed underbodies, and different wheel configurations), divided into 5,600 for training, 1,200 for validation, and 1,200 for testing, with results shown in Table 4.

The shape variation in DrivAerNet++ (refer to Figure 4) introduces additional challenges. For example, replacing the detailed underbody with a smooth underbody can shift the drag distribution for the same car design. Changing the wheels from open to closed can slightly affect the drag. Additionally, different rear configurations can result in varying flow field separation behaviors, causing significant changes in drag values. These factors make DrivAerNet++ a very challenging task for generalization, as seemingly minor changes can significantly impact drag values and present difficulties for deep learning models in accurately learning the features of these variations.

Table 4: Comparative analysis of deep learning models for aerodynamic drag prediction on the test set from DrivAerNet++ (All cars) comprising 1,200 car designs.

Model	MSE ↓ ($\times 10^{-5}$)	MAE ↓ ($\times 10^{-3}$)	Max AE ↓ ($\times 10^{-3}$)	R^2 ↑	Training Time ↓	Inference Time ↓	Number of Parameters
PointNet [55]	14.9	9.60	12.45	0.643	2.06hrs	0.84s	2,348,097
GCNN [44]	17.1	10.43	15.03	0.596	49hrs	50.8s	100,481
RegDGCNN [24]	14.2	9.31	12.79	0.641	12.6hrs	0.85s	3,164,257

5.1.2 Aerodynamic drag prediction from tabular parametric data

We also approach the task of aerodynamic drag prediction using parametric data. For this purpose, we employ an AutoML (automated machine learning) framework [26] optimized with Bayesian hyperparameter tuning, along with models such as Gradient Boosting [28], XGBoost [13], LightGBM [42], and Random Forests [11]. These methods leverage design parameters to estimate aerodynamic drag, eliminating the need for detailed 3D geometry. Such an approach is invaluable for efficiently evaluating the impact of geometric modifications on drag and overall car performance. The use of parametric data offers significant advantages due to its accessibility and ease of manipulation compared to 3D mesh modifications. Engineers can swiftly adjust design parameters and immediately observe the effects on aerodynamic performance, thereby streamlining the design process.

We train the models on a single parametric car design (fastback) and, in another experiment, on all parametric models (fastback, notchback, and estateback), to explore how well the models generalize across different designs. For both experiments, we split the dataset into 80% for training and 20% for testing. We then further divided the training set into subsets at 20%, 40%, 60%, 80%, and 100% of the training portion.

To standardize the parametric studies, we focused on 26 parameters instead of 50, as the 50 geometric parameters model from DrivAerNet [24] was based only on one car category, specifically the fastback with a detailed underbody and open wheels. The results, illustrated in Figure 5, reveal that AutoGluon performs better in the single fastback category, whereas LightGBM excels on the combined dataset. Nonetheless, performance declines for all models on the combined dataset. A significant finding is that for all models, both in single and combined categories, enlarging the dataset size leads to enhanced performance. For example, the R^2 value for XGBoost increases from approximately 0.35 to 0.55 by increasing the training set size from 640 to 3,200.

8

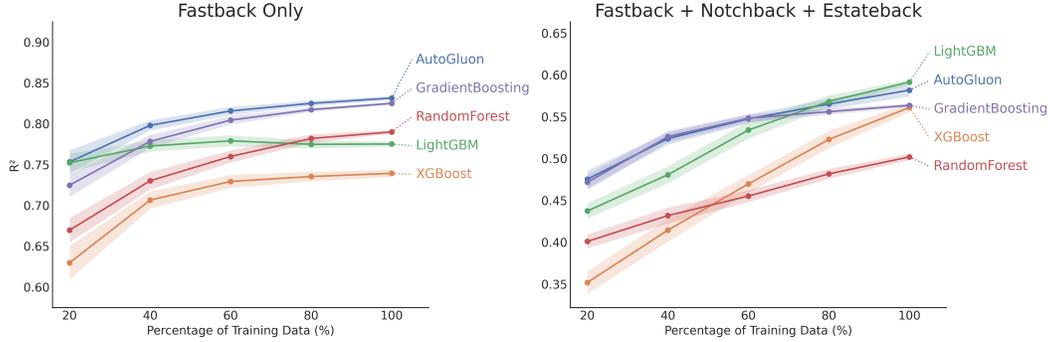



Figure 5: Drag coefficient prediction based on the parametric data for different car categories. The plots show the median and 95% confidence interval of the R^2 score as a function of the percentage of the training data.

6 Conclusion

In this paper, we introduced DrivAerNet++, the largest, multimodal 3D dataset for data-driven aerodynamic design, incorporating high-fidelity CFD simulations and a diverse array of car designs. Our dataset includes 8,000 cars based on industry-standard shapes, offering extensive coverage of various aerodynamic performance metrics. The dataset, requiring 39 TB of storage, is significantly larger than comparable datasets in engineering and is made publicly available. The computational cost of generating DrivAerNet++ was an order of magnitude larger than that of the recently published CFD dataset from [45], which utilized 185,744 CPU-hours, whereas ours required 3,000,000 CPU-hours.

This comprehensive dataset supports a wide range of machine learning tasks, including surrogate modeling of aerodynamic performance, acceleration of CFD simulations, data-driven design optimization, generative AI, shape and part classification, and 3D shape reconstruction. We also presented the first benchmarking results, demonstrating the effectiveness of geometric deep learning models and AutoML frameworks for drag coefficient prediction. Furthermore, we explored the challenges of creating a generalized model for surrogate modeling of drag across different car categories. While models trained on a single car category performed well, their performance in terms of R^2 decreased significantly from 0.82-0.9 to 0.6 when applied to the full diverse dataset. This underscores the complexity of achieving robust performance across varied designs.

Our dataset can be utilized for data-driven design of both internal combustion engine (ICE) cars and electric vehicles, covering main design aspects such as aesthetics/style and aerodynamic efficiency and performance. We believe this dataset will serve as a cornerstone for advancing research in engineering design and CFD, offering rich resources for developing more accurate and efficient predictive models.

7 Limitations and Future Work

The surrogate models we trained on DrivAerNet++ are currently not complex enough to fully capture the intricate geometric and aerodynamic features present in diverse car designs. More advanced models, such as Geometry-Informed Neural Operators [45], Convolutional Occupancy Networks [53], and sophisticated graph models, should be explored to improve predictive performance. We primarily focused on the surrogate modeling of drag, as it is a critical factor in the initial design phase. However, leveraging DrivAerNet++ for additional applications, such as accelerating CFD simulations, would provide a more comprehensive tool for aerodynamic design.

To enhance the dataset, future work will focus on integrating transient CFD simulations and incorporating additional modalities, such as 2D image renderings and multimodal learning approaches. These enhancements aim to improve model accuracy and robustness, fostering further innovation in automotive design and optimization.

9

Acknowledgments and Disclosure of Funding

The authors thank Ravi Nimbalkar and Yatin Kumbhar from BETA CAE Systems USA Inc for their support in creating the parametric models in ANSA®. The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC and consultation resources that have contributed to the research results reported within this paper. Special thanks to Chris Hill for providing access to MIT SuperCloud and to Lauren Milechin for her support in setting up the simulation environment. Additionally, the authors wish to thank Amy Nurnberger from MIT and Sonia Barbosa from Harvard University for their support with uploading the dataset to Harvard Dataverse. Finally, the authors extend their gratitude to Leonid Andreev for his technical support with uploading the dataset to Harvard Dataverse.

10

Checklist

1. For all authors...

    (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

    (b) Did you describe the limitations of your work? [Yes] This is done in Section 7

    (c) Did you discuss any potential negative societal impacts of your work? [N/A]

    (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] Our work does not involve human participants or crowd-sourcing. The data used is licensed, consented, and adheres to the FAIR (Findable, Accessible, Interoperable, and Reusable) principles [73].

2. If you are including theoretical results...

    (a) Did you state the full set of assumptions of all theoretical results? [N/A]

    (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments (e.g. for benchmarks)...

    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The source code utilized for model training and assessment in this research can be accessed at the given repository `https://github.com/Mohamedelrefaie/DrivAerNet`, where the README file offers detailed guidance for usage.

    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] DrivAerNet++ is designed as a benchmark dataset for 3D problems, complete with a split into training, validation, and test sets to ensure fair comparisons among various machine and deep learning models. Details on the dataset splits can be found here: `https://github.com/Mohamedelrefaie/DrivAerNet`

    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]

    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The high-fidelity CFD simulations for DrivAerNet++ were conducted on the MIT SuperCloud, utilizing parallelization across 60 nodes, totaling 2880 CPU cores. The simulations took approximately 3×10^6 CPU-hours to complete. For more details, refer to Section 3: Computational cost.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

    (a) If your work uses existing assets, did you cite the creators? [Yes] The baseline CAD geometry of the DrivAer model is available online for free download [49], but registration is required, and we have cited the creators.

    (b) Did you mention the license of the assets? [N/A] The CAD geometry is not provided under a license, however it requires registration for access.

    (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] All new car designs created by the authors of this paper are included in the dataset and will be hosted on Harvard Dataverse Repository.

    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Consent is not applicable as all car designs were created by the authors of this paper.

    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data does not contain any personally identifiable information or offensive content.

5. If you used crowdsourcing or conducted research with human subjects...

    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

11

Table of Contents for Appendices

A Reproducibility Statement   13

B URL and Links   13

C Author Statement and Data License Confirmation   13

D Broader Impact   14

E DrivAerNet++ Dataset Generation   15

   E.1 Generating a wide range of car designs   15

   E.2 CFD Meshing   17

   E.3 Numerical Simulation   18

        E.3.1 Domain and Boundary Conditions   18

        E.3.2 Meshing Sensitivity   20

        E.3.3 Validation of the Numerical Results   20

   E.4 Convergence Detection   21

F In-Depth Insights from DrivAerNet++   22

   F.1 Dataset Annotations   26

G Deep Surrogate Models   26

   G.1 RegDGCNN   26

   G.2 PointNet   26

   G.3 Graph Neural Network   27

H Evaluation   27

I Additional Experiments   28

   I.1 3D Surface Field Prediction   28

   I.2 t-SNE Visualization of the Design Space   29

12

A   Reproducibility Statement

We provide a GitHub repository to facilitate the reproduction of all experiments in the main paper, along with links to download the dataset. The repository includes the following: code to replicate the machine learning experiments and code to generate the figures presented in the paper. The full dataset will be made available after the peer-review process is complete. The training and experiments were performed on a machine equipped with AMD EPYC 7763 64-Core Processors, totaling 256 CPU cores, and four Nvidia A100 80GB GPUs.

B   URL and Links

DrivAerNet++ Dataset: The DrivAerNet [24] and the DrivAerNet++ datasets are available on our GitHub repository at `https://github.com/Mohamedelrefaie/DrivAerNet`.

Croissant Metadata: The URL to Croissant metadata record documenting the dataset/benchmark is available on our GitHub repository at `https://github.com/Mohamedelrefaie/DrivAerNet/tree/main/mlcroissant`.

DL Code: The source code utilized for model training and assessment in this research can be accessed at the given repository `https://github.com/Mohamedelrefaie/DrivAerNet`, where the README file offers detailed guidance for usage.

Tutorials and Browsing Tools: Additional resources, including data handling tutorials and contribution guidelines, are available at `https://github.com/Mohamedelrefaie/DrivAerNet/tree/main/tutorials`

C   Author Statement and Data License Confirmation

We, the authors, hereby declare that we bear full responsibility for any violations of rights or other issues arising from the use of the data. We confirm that the dataset is available under the Creative Commons Attribution-NonCommercial (CC BY-NC) license. This license allows others to remix, tweak, and build upon our work non-commercially, and although their new works must also acknowledge our original work and be non-commercial, they don't have to license their derivative works on the same terms.

13

D   Broader Impact

The DrivAerNet++ dataset can be used for several tasks, such as data-driven design optimization, training surrogate models, data-driven turbulence modeling, accelerating CFD, 3D shape and part classification, 3D shape reconstruction, and generative design. We are confident that DrivAerNet++ will serve as a highly valuable resource for multiple research communities in various ways:

1. **Advancing Automotive Aerodynamic Design and Accelerating Design Cycles:** Enabling detailed studies in aerodynamics to foster innovation in car design for enhanced performance and efficiency, while assisting CFD engineers in creating more efficient and accurate surrogate models, speeding up design iteration cycles, and democratizing simulations to provide designers with greater freedom.

2. **Accelerating CFD Simulations:** DrivAerNet++ provides high-fidelity simulation data that can be used to train surrogate models, reducing the computational cost and time required for CFD simulations. This allows for faster iterations and more efficient exploration of the design space.

3. **Machine Learning Integration:** Offering a rich resource for training and testing advanced machine learning models, including those in geometric deep learning, to understand complex aerodynamic phenomena.

4. **Benchmarking and Validation:** Serving as a benchmark dataset for ML algorithms, aiding in the validation and comparison of different modeling approaches.

5. **Environmental Impact:** Contributing to the development of more aerodynamically efficient cars, which can lead to reduced fuel consumption and lower emissions, and extending the range of electric cars.

6. **Safety and Prototyping:** Reducing the need for physical prototyping and wind tunnel testing, thereby lowering costs and enhancing safety by enabling virtual testing of numerous design variants.

7. **Cross-Disciplinary Insights:** Offering insights into the intersection of CFD, ML, and automotive design, encouraging cross-disciplinary research and collaboration.

This multimodal, large-scale, and high-fidelity CFD dataset of 8000 car shapes is expected to catalyze significant advancements in both the academic and industrial realms. The methodology and approach implemented here are not only limited to car design but can also be extended to other applications involving numerical simulations and deep learning, such as aerospace design.

14

E DrivAerNet++ Dataset Generation

E.1 Generating a wide range of car designs

The goal is to create a large-scale dataset of 3D shapes of cars with high-quality mesh resolutions, watertight geometries, and diverse configurations that resemble real-world car designs. The ShapeNet [12] dataset offers a wide range of car shapes; however, as discussed, the mesh resolution and non-watertightness make it unsuitable for large-scale engineering designs or CFD datasets. Other generic car models, like the Ahmed body [2] or the Windsor body [52] are very simple and do not represent actual cars with realistic shapes. Therefore, we chose the DrivAer body [35, 49] and parametrized the model with different configurations. In addition, the choice of the DrivAer model is supported by the availability of both computational and experimental references, allowing us to validate our results against established data [35, 34, 33].

The parametric models[4], along with the constraints and the bounds applied during the Design of Experiments (DoE) were generated using the commercial software ANSA® (see Figure 6). We

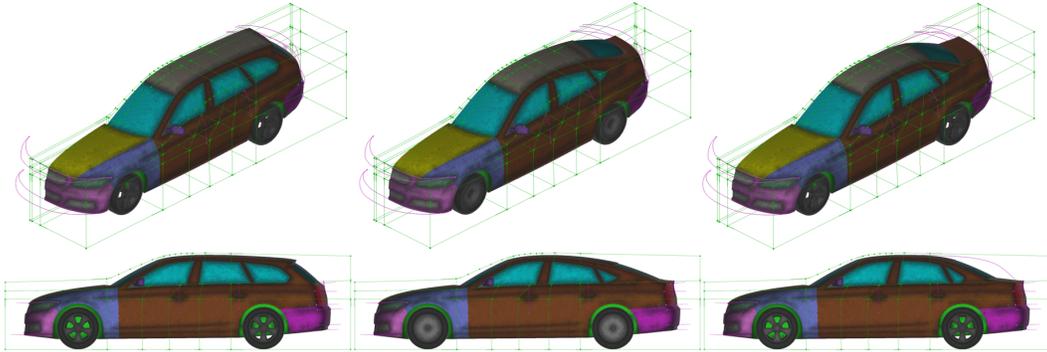



Figure 6: Three parametric models from DrivAerNet++. The green boxes represent the morphing boxes, and the purple curves illustrate the trajectories followed during the morphing of the car geometry. A total of 26 parameters were utilized to adjust and refine the car designs.

employed two of the ANSA® advanced tools to create the parametric models and generate the DoE: the morphing tool and the optimization tool. The morphing tool allows for versatile processing of alternative variations in each model, with functions that can reshape geometry entities. To create diverse car designs, we used two main morphing methods: morphing boxes and direct morphing. The morphing boxes can load any type of entities; in our case, we used this method to create four parameters that control the overall length, width, roof height, and greenhouse angle of the car.

To create detailed parameters that control specific features of the car model, such as windshield inclination, ramp angle, diffusor angle, etc., the direct morphing method was employed using different movement types provided by the DFM (Direct Fitting Movements) function.

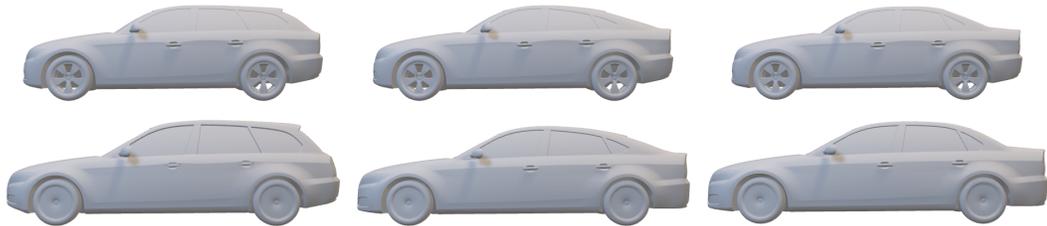



Figure 7: Variations of estateback, fastback, and notchback car designs from left to right, showcasing models with the smallest (top) to largest (bottom) volume.

Figure 7 illustrates the variations in car designs, highlighting the range from smallest to largest volumes across estateback, fastback, and notchback models. This illustration highlights the extensive

---

[4]Available at: `https://github.com/Mohamedelrefaie/DrivAerNet/ParametricModels`

15

Table 5: Design parameter names, step sizes, ranges, and units, categorized by group. These parameters are used for morphing the baseline geometries to create the DrivAerNet++ dataset. For Group I: Underbody (0 for smooth underbody, 1 for detailed underbody), Car Rear (0 for fastback, 1 for estateback, 2 for notchback), and Wheels (0 for open detailed, 1 for open smooth, 2 for closed smooth).

#	Group	Parameter Name	Minimum Value	Maximum Value	Step Value	Range	Units
1		Car Width	-60	150		Bounds	mm
2	A	Car Length	-60	130		Bounds	mm
3		Car Roof Height	-60	90		Bounds	mm
4		Car Green House Angle	-200	150		Bounds	°
5		Diffusor Angle	-8	15		Bounds	°
6	B	Ramp Angle	-8	15		Bounds	°
7		Trunklid Angle	-8	20		Bounds	°
8		Side Mirrors Rotation	-20	20		Bounds	°
9	C	Side Mirrors Translate X	-10	20		Bounds	mm
10		Side Mirrors Translate Z	-5	10		Bounds	mm
11		Rear Window Inclination	-1.8	2.5		Bounds	°
12	D	Rear Window Length	-95	150		Bounds	mm
13		Windscreen Inclination	-1.8	2.5		Bounds	°
14		Windscreen Length	-50	50		Bounds	mm
15	E	A B C Pillar Thickness	-15	5	0.1	Step	%
16		Fenders Arch Offset	-25	55		Bounds	mm
17		Door Handles Thickness	-20	30		Bounds	mm
18	F	Door Handles X Position	-50	50		Bounds	mm
19		Door Handles Z Position	-30	30		Bounds	mm
20	G	Trunklid Curvature	-0.4	0.8	0.1	Step	%
21		Trunklid Length	-40	40		Bounds	mm
22	H	Front Bumper Curvature	-0.4	1		Bounds	%
23		Front Bumper Length	-25	60		Bounds	mm
24		Underbody	0	1		Binary	0/1
25	I	Car Rear	0	2		Options	0/1/2
26		Wheels	0	2		Options	0/1/2

range covered by the DrivAerNet++, which includes all conventional production car configurations to ensure comprehensive aerodynamic analysis. Larger volumes might be relevant for passenger comfort or electric battery installation, reflecting the practical considerations in car design.

The parameters listed in Table 5 are used to morph the baseline geometries. Each parameter is categorized by group (A, B, C, etc.) and includes its minimum and maximum values, step sizes, range type, and units. Some design parameters are more relevant than others depending on the task; for example, A/B/C pillar thickness might be relevant for structural integrity and crashworthiness, while parameters like the diffusor angle are more critical for aerodynamics. Other parameters, such as windscreen inclination, are more relevant from an aesthetic perspective. This comprehensive parameterization ensures that the dataset captures diverse design attributes, enabling deep learning models to generalize well to out-of-distribution designs and flow field features.

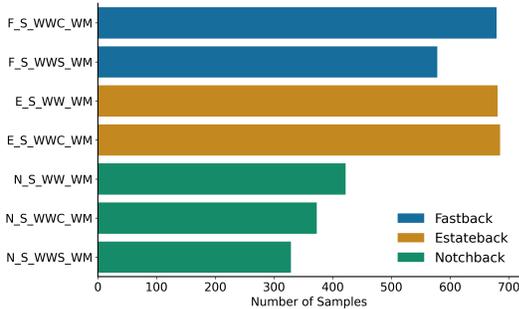



Figure 8: Sample counts for different car configurations in the DrivAerNet++ dataset. The figure shows the number of samples for each car category.

Figure 8 presents the distribution of samples across different car categories within the DrivAerNet++ dataset. The first letter in the configuration name refers to the car body type: **F** for Fastback, **E** for Estateback, and **N** for Notchback. The second letter indicates the underbody type: **S** for Smooth underbody. The third letters describe the wheels configuration: **WWC** for with wheels closed, **WWS** for with wheels open smooth, and **WW** for with wheels open detailed. The fourth letters denote the presence of mirrors: **WM** for with mirrors. This detailed classification ensures precise categorization and analysis of the different car designs within the dataset. For comparison, the DrivAerNet [24] was based only on the fastback category with detailed underbody, with open detailed wheels, and with mirrors F_D_WW_WM.

16

In Figure 9, we show the statistics of the design parameters in the DrivAerNet++ dataset. Each subplot represents a different design parameter.

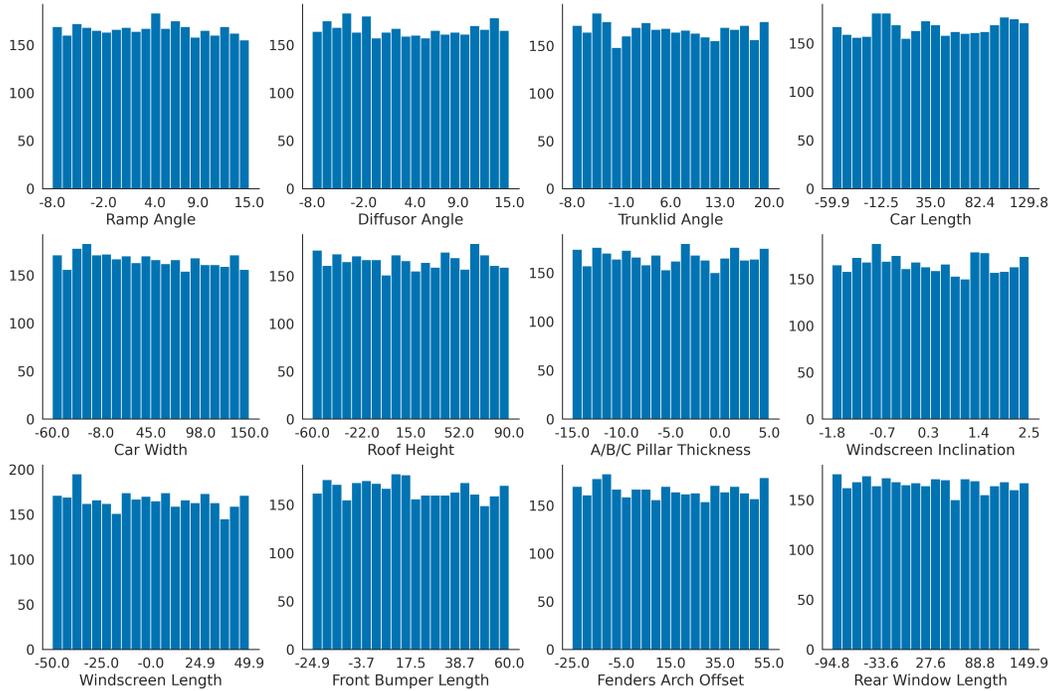



Figure 9: Distribution of various geometric parameters in the DrivAerNet++ dataset. The histograms demonstrate the range and frequency of these parameters, showcasing the diversity in our dataset.

E.2 CFD Meshing

The meshing procedure was conducted using the `snappyHexMesh` utility [50]. This process involved three main stages: castellated mesh generation, snapping, and layer addition, ensuring a high-quality, boundary-fitted mesh suitable for accurate aerodynamic simulations. Firstly, the geometry components, including the car body, front wheels, and rear wheels, were defined as triangular surface meshes.

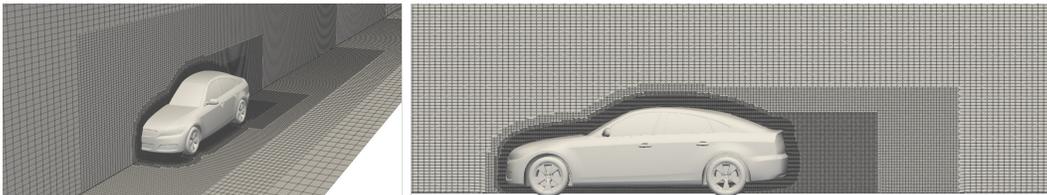



Figure 10: Mesh generation for the fastback model using SnappyHexMesh (SHM) [50]. The left view shows a perspective of the car model within the computational domain, while the right view highlights the mesh refinement around the car body from a side perspective. The computational grid in the y-normal symmetry plane reveals four distinct regions of refinement, with additional layers surrounding the car body. These refinements are designed to capture both wake dynamics and boundary layer development accurately.

In the castellated mesh generation stage, the mesh was initially refined around the surfaces, with the car body and wheels receiving high refinement levels (up to level 5). The snapping stage involved aligning the mesh with the geometry surfaces. Patch smoothing iterations ensured the mesh points conformed accurately to the surface features. Five mesh relaxation iterations were performed to achieve the correct mesh alignment. The final stage, layer addition, involved growing prism layers on the car body, wheels, and ground surfaces. Five layers of different resolutions were applied

17

adjacent to the car body, with the rear region receiving additional refinement to accurately capture the detaching and reattaching flow.

Volumes of refinement (VoR) were specified using searchable boxes with precise coordinates to target critical areas for mesh refinement, as shown in Figure 10. Instead of using one mesh template for all simulations, we developed a mesh refinement strategy based on the car shape, category, and dimensions. The size, dimensions, and position of the refinement boxes were adjusted according to the car geometry and category to ensure accurate modeling of the wake flow. Figure 11 illustrates the variation in mesh sizes within the DrivAerNet++ dataset. The meshes in DrivAerNet++ consist mainly of hexahedra cells, along with prisms, tetrahedral wedges, tetrahedra, and polyhedra. The distribution of points, cells, faces, and internal faces is also illustrated, providing an overview of the mesh quality and complexity.

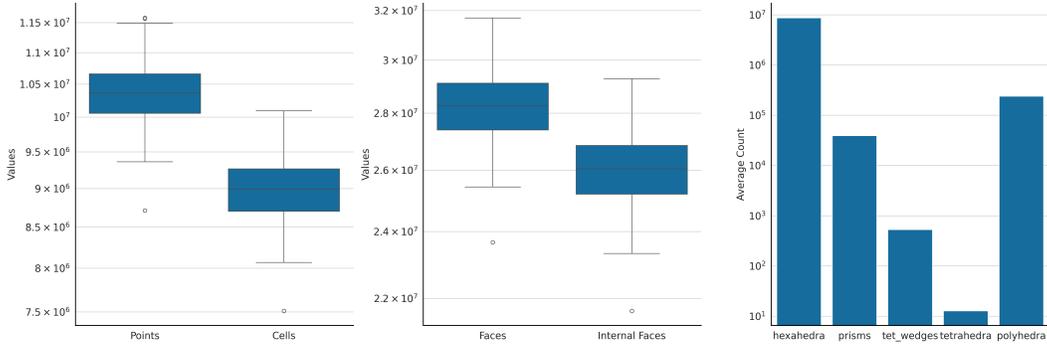



Figure 11: Mesh statistics for the DrivAerNet++ dataset. Left: distribution of points and cells; middle: distribution of faces and internal faces; right: average cell counts by type.

E.3 Numerical Simulation

E.3.1 Domain and Boundary Conditions

For the CFD simulations, we used the open-source software OpenFOAM® v11,[5] a comprehensive suite of C++ modules that supports solver customization, utility development, and parallelization in CFD studies. In the field of aerodynamics, only two open-source tools—OpenFOAM [30] and SU2 [21]—are commonly used, with OpenFOAM being the state-of-the-art tool most widely adopted in both industry and academia for car aerodynamics. A review of the literature indicates no significant technical advantages of using SU2 over OpenFOAM for this application. We utilized 1:1 scaled models of all car designs and the SIMPLE algorithm (Semi-Implicit Method for Pressure Linked Equations) to couple pressure and velocity in the incompressible fluid simulations. These were carried out using the *simpleFoam* solver, which is ideal for steady-state, turbulent, and incompressible flow simulations (In OpenFOAM® v11, the *incompressibleFluid* modular solver replicates *simpleFoam*).

We selected the state-of-the-art k-ω-SST turbulence model based on Menter's formulation [47] for the Reynolds-Averaged Navier-Stokes (RANS) simulations. This model is widely used in both academic research and industry due to its ability to overcome the limitations of the standard k-ω model, particularly its sensitivity to the freestream values of k and ω, and its effectiveness in predicting flow separation.

Following a similar approach to [34], we set the solver tolerances to 10^{-6} for pressure (p), 10^{-8} for velocity (U) and turbulence quantities (k and ω), and 10^{-7} for the potential solver. The relative solver tolerance, defined as the ratio between the current and initial residuals, was set to 10^{-2} for pressure, 10^{-3} for velocity and turbulence quantities, and 10^{-2} for the potential solver. The simulations were conducted at a flow velocity (u_∞) of 30 m/s (108 km/h), corresponding to a Reynolds number range of approximately 8.366×10^6 to 1.006×10^7, using the car's length as the characteristic length scale. These Reynolds numbers are based on the car with the smallest length (4344 mm) and the largest length (5210 mm). All simulations were performed using 12 million cells (refer to the meshing section for more details), and due to the symmetry, this amounts to a total of 24 million cells for

---

[5]https://openfoam.org/version/11/

18

the full simulation. Additional layers were added around the car body to accurately capture wake dynamics and boundary layer evolution (see Figure 10). Different meshing and refinement strategies were employed based on the car category, as the rear end varies, necessitating distinct meshing at the back of the car to account for flow separation and accurately model the wake flow. For the car underbody, we differentiated between smooth and detailed underbody configurations.

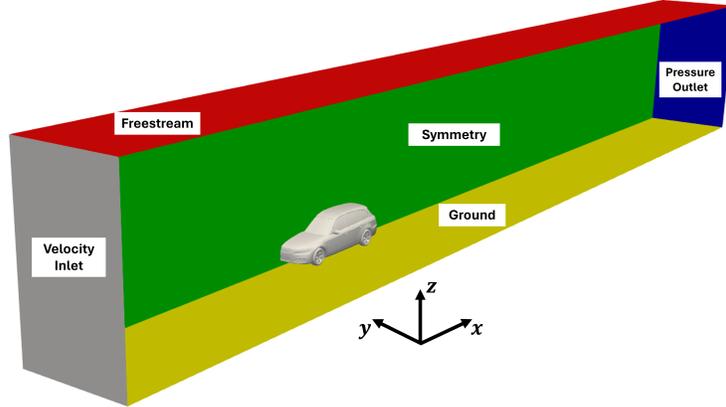



Figure 12: CFD simulation domain and boundary conditions used for our DrivAerNet++ dataset. The domain includes the velocity inlet (grey), pressure outlet (blue), freestream (red), symmetry (green), ground (yellow), and side wall (not shown) boundaries. The car surface was assigned no-slip conditions, the ground was assigned a moving wall condition, and the wheels were modeled with a rotating wall velocity boundary condition.

Boundary conditions were defined with a uniform velocity at the inlet and pressure-based conditions at the outlet. To prevent backflow into the simulation domain, the velocity boundary condition at the outlet was set as an *inletOutlet* condition. The car surface was assigned no-slip conditions, and the ground was assigned a moving wall condition, while the wheels were modeled to rotate with the *rotatingWallVelocity* condition, defined as:

$$u_{\text{wheels}} = \omega \times r \tag{1}$$

where ω is the angular velocity and r is the wheel radius. Slip conditions were applied to the lateral and top boundaries of the domain.

As illustrated in Figure 12, the domain includes the velocity inlet, pressure outlet, freestream, symmetry, ground, and side wall boundaries. Compared to DrivAerNet [24], we utilized the symmetry boundary condition, which leverages the symmetry of the car and domain. This allowed us to double the mesh resolution while maintaining the same computational time for the symmetric car designs.

Near-wall viscosity effects were addressed using the *nutUSpaldingWallFunction* wall function approach. This wall function for the viscosity term applies a continuous turbulent viscosity profile near the wall, based on velocity, as proposed by [64]. For divergence terms, the default Gauss linear scheme was utilized, with the velocity convective term discretized using a bounded Gauss *linearUpwindV* scheme to ensure second-order accuracy. Gradient calculations were performed using the Gauss linear method, complemented by a multi-dimensional limiter to enhance solution stability. The primary quantities of interest were the 3D flow field, surface pressure, wall-shear stresses, 3D streamlines, and aerodynamic coefficients.

Table 6: Definition of the quantities and their values for air at sea level and at a temperature of 298.15 K (25 °C).

Quantity	Definition	Value
ρ	Density of air	1.184 kg m^{-3}
ν	Kinematic viscosity of the fluid	1.56×10^{-5} m^2 s^{-1}
u_∞	Inlet velocity	30 m/s (108 km/h)
Re_L	Reynolds number	$8.366 - 10.06 \times 10^6$

19

We used the *potentialFoam* solver to initialize the OpenFOAM® simulations. This solver calculates a potential flow solution to quickly establish an initial velocity field, thereby accelerating the convergence of the solution in complex aerodynamic simulations [30]. The potential flow assumption neglects viscous effects and assumes irrotational flow, providing a good starting point for the *simpleFoam* solver. The simulations were conducted using air with the properties listed in Table 6.

E.3.2 Meshing Sensitivity

Mesh sensitivity analysis is conducted to ensure that simulation results are independent of the mesh resolution, which can significantly impact the accuracy of the CFD simulations. By varying the mesh density and observing its effect on outputs like the drag coefficient, analysts can determine the optimal mesh size that balances computational cost with solution accuracy. This process helps to confirm that the results are not artifacts of the chosen mesh and improves confidence in the reliability of the simulation predictions. We conducted a mesh sensitivity analysis to examine the effect of mesh resolution on the drag coefficient specifically for the fastback configuration. The findings are summarized in Table 7, which shows that as the mesh quality improves from coarse (6M cells) to fine (24M cells), the relative error in drag coefficient decreases from 8.21% to 2.17%, indicating a convergence towards the reference CFD results from TUM [34]. However, the computational time increases significantly with finer meshes, demonstrating the trade-off between accuracy and computational cost. This analysis highlights the importance of selecting an appropriate mesh resolution to balance precision with efficiency in CFD simulations.

Table 7: Influence of mesh resolution on drag coefficient for the fastback case. The relative error is computed with respect to the CFD results from TUM [34].

Mesh Quality	Number of Cells	Relative Time	Relative Error in Drag (%)
Coarse	6M	1.00	8.21%
Medium	12M	1.25	6.55%
Fine	24M	1.88	2.17%

E.3.3 Validation of the Numerical Results

In this section, we present validation results for the baseline simulations. The assumption is that given a baseline geometry, conducting the CFD simulation and benchmarking the results against both reference simulations and wind tunnel experimental data should establish a foundation for using the same CFD meshing approach, boundary conditions, and solvers for the morphed geometries of this baseline geometry. However, DrivAerNet++ encompasses various car designs and categories, making a single mesh unsuitable due to variations in car length, rear end, and underbody features. Therefore, as detailed in the meshing section E.2, we employ different meshing strategies based on car categories. Similarly, we benchmark the CFD simulation results across different geometries, focusing on the following car categories:

1. Fastback with open detailed wheels, detailed underbody, and mirrors
2. Fastback with open detailed wheels, smooth underbody, and mirrors
3. Notchback with open detailed wheels, smooth underbody, and mirrors
4. Estateback with open detailed wheels, smooth underbody, and mirrors

Based on this, the only factors that need adjustment for other car categories should be the geometry and meshing strategy, not the CFD solver or discretization scheme. To ensure convergence, 7000 iterations were performed, with forces averaged over the last 1000 iterations. We provide both the mean and standard deviation for uncertainty quantification in both the CFD and deep learning model predictions.

We compare our results with the original paper by the creators of the DrivAer body [34, 35]. The primary goal is to determine whether the CFD modeling accurately predicts the correct trend of drag values within acceptable tolerance. The drag coefficient is defined as:

$$C_d = \frac{F_d}{\frac{1}{2}\rho u_\infty^2 A_{\text{ref}}} \tag{2}$$

20

where F_d represents the drag force experienced by the body, A_{ref} is the effective frontal area, u_∞ is the freestream velocity, and ρ is the air density.

Table 8 presents the comparison of drag coefficients (C_D) from our simulations, the reference CFD simulation, and wind tunnel experimental data provided by TUM. The simulation results show good agreement with the reference simulations and the experimental results, demonstrating a strong correlation between the drag values and ensuring a balance between simulation accuracy (fidelity) and computational cost (simulation time and mesh size). For the fastback car with a smooth underbody, the difference is less than 3%, and with the detailed body, we have a difference of 2.18%. For both the notchback and estateback configurations, the differences are still below 5%. The estateback configuration exhibits the highest drag coefficient, whereas the difference in drag coefficients between the fastback and notchback configurations is minimal.

Table 8: Comparison of drag coefficients (C_D) from experiments and CFD simulations.

	Experiment TUM	Reference CFD TUM	Ours	Difference Exp.	Difference CFD
Fastback smooth	0.243	0.241	0.236	2.88%	2.07%
Fastback detailed	0.275	0.278	0.269	2.18%	3.24%
Notchback	0.246	n/a	0.234	4.88%	n/a
Estateback	0.292	n/a	0.280	4.11%	n/a

The results in Table 9 demonstrate our model's ability to accurately predict differences in drag coefficients (ΔC_D) between various car configurations. For instance, the predicted drag difference between the fastback detailed (FD) and fastback smooth (FS) configurations is 0.033, closely aligning with the experimental difference of 0.032. Similarly, the difference between the estateback (E) and notchback (N) configurations is exactly predicted as 0.046, matching the experimental result. However, some discrepancies, such as the difference between the estateback and fastback smooth configurations (0.039 vs. 0.049 experimentally), indicate areas for further refinement. Overall, the model shows strong alignment with experimental data for most configurations, with deviations generally within an acceptable range, highlighting the balance between simulation accuracy and the computational cost required.

Table 9: Comparison of differences in drag coefficients (ΔC_D) from experiments and CFD simulations.

	Experiment TUM	Reference CFD TUM	Ours
FD - FS	0.032	0.037	0.033
FD - N	0.029	n/a	0.035
E - FD	0.017	n/a	0.011
N - FS	0.003	n/a	-0.002
E - FS	0.049	n/a	0.039
E - N	0.046	n/a	0.046

E.4 Convergence Detection

For detecting the convergence of each CFD case, we implemented the following approaches:

- **Geometry Quality**: Ensuring car designs are watertight and free from intersecting faces or holes is critical. We use the functionalities provided by ANSA® and Blender [16] software for geometry cleaning and processing. In addition, we ensure that the 3D meshes are correctly aligned within the simulation domain and that the wheelbase distance between the wheels is accurately defined, which is necessary for the *rotatingWallVelocity* boundary condition. Additionally, we ensure the correct estimation of the car's frontal projected area, which is essential for accurate drag coefficient calculations.

- **CFD Mesh Quality**: For mesh quality, we employ the *checkMesh* utility from OpenFOAM®. This utility verifies various aspects such as the overall domain bounding box, geometric and solution directions, boundary and cell openness, aspect ratio, face and cell volumes, mesh non-orthogonality, face pyramids, skewness, and coupled point location match. These checks ensure that the mesh meets predefined criteria, guaranteeing high quality and suitability for accurate CFD simulations, as detailed in the OpenFOAM® documentation.

- **Drag Convergence Detection**: Using a runtime control algorithm, we average drag values and provide statistical errors (mean and standard deviation) to assist in uncertainty quantification and improve deep learning model training.

21

- **Residual Monitoring**: Monitoring residuals for main physical quantities, ensuring convergence when all RMS residual values are below 10^{-5}, which indicates a well-converged numerical solution.
- **Outlier Detection**: Identifying and removing outlier designs that passed quality checks but adversely affected model predictions during training and validation.

F In-Depth Insights from DrivAerNet++

In this section, we provide an overview of the contents and results from the CFD simulations included in our DrivAerNet++ dataset, highlighting the diversity and richness of the data. This includes pair plots of aerodynamic coefficients (Figure 13), visualizations of streamlines (Figure 14), pressure distribution on the car surface (Figure 15), and the 3D velocity field around the car (Figure 16), illustrating the comprehensive range of aerodynamic behaviors captured in the dataset.

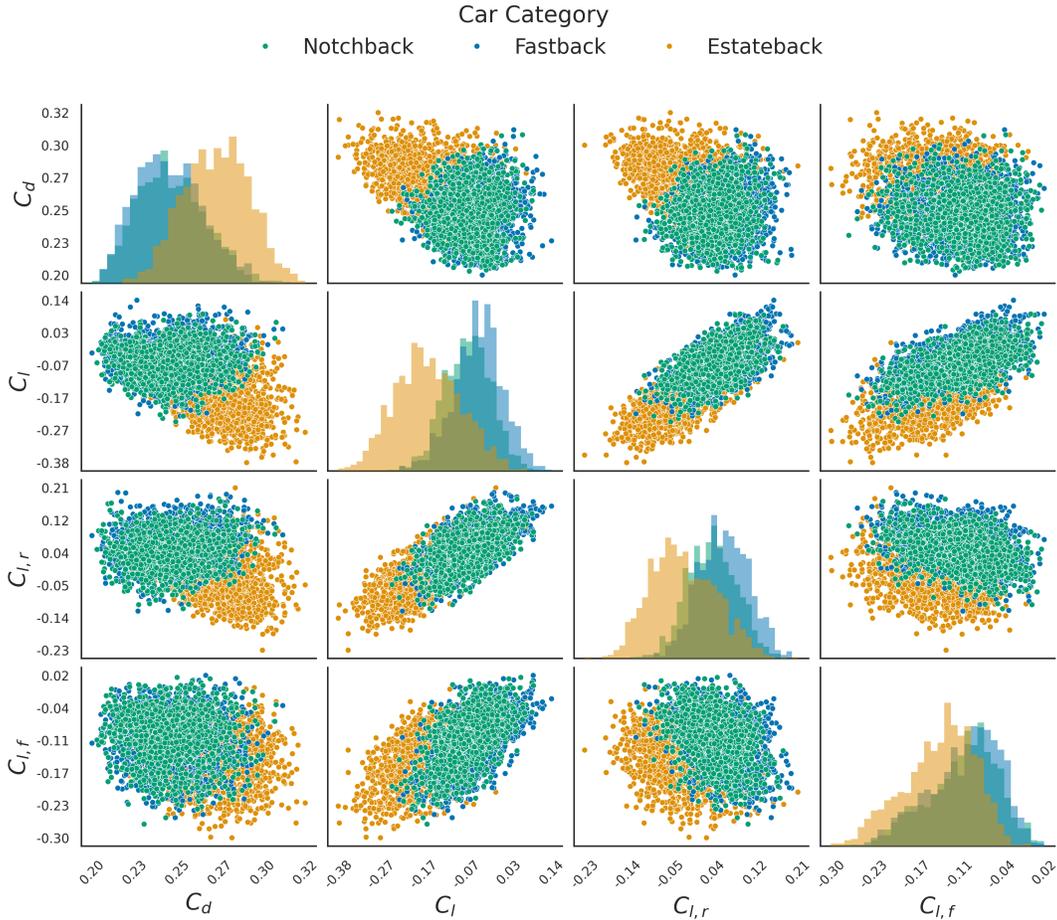



Figure 13: Pair plot showing the relationships between different aerodynamic coefficients (C_d, C_l, $C_{l,r}$, and $C_{l,f}$) in the DrivAerNet++ dataset. The diagonal histograms represent the distribution of each coefficient, while the scatter plots show the pairwise correlations between the coefficients. The colors represent different configurations within the dataset, illustrating the diversity in aerodynamic performance.

22

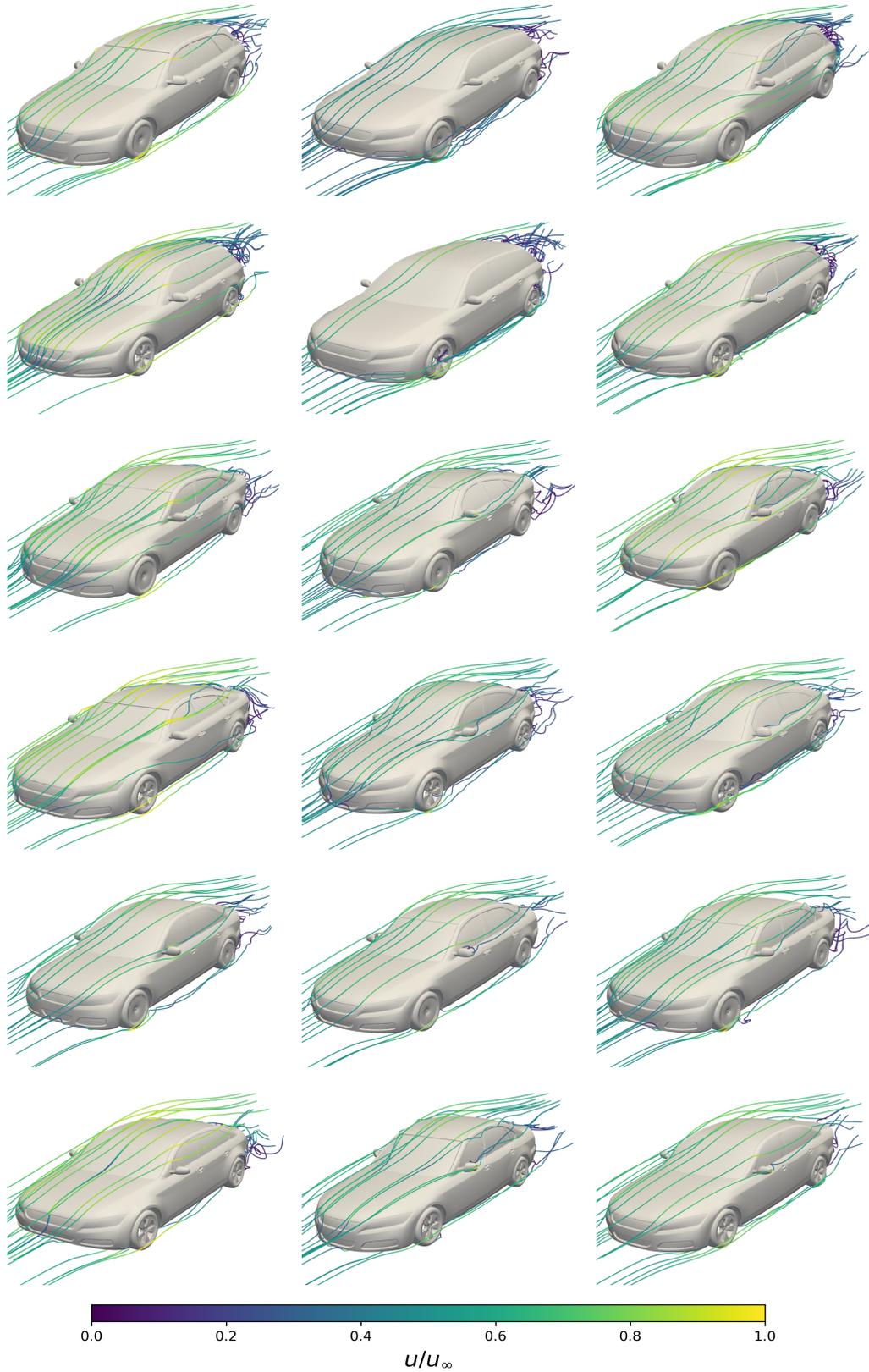



Figure 14: Streamlines visualizations for different car designs, showcasing the large variations in shape within the DrivAerNet++ dataset. The color indicates the normalized velocity u/u_∞, with values ranging from 0 to 1.

23

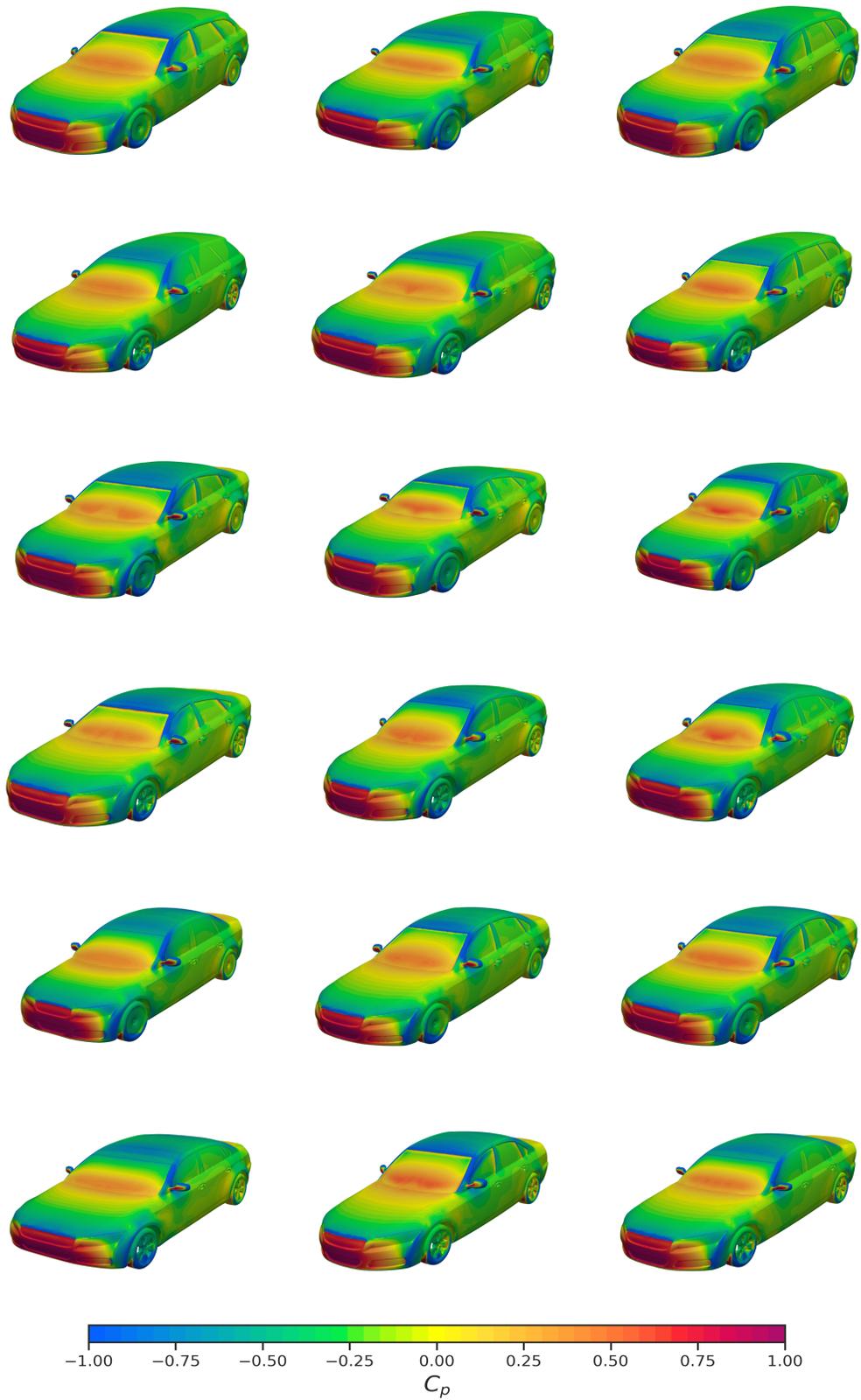



Figure 15: Pressure coefficient (C_p) visualizations for different car designs, illustrating the large variations in shape within the DrivAerNet++ dataset. The color indicates the pressure coefficient C_p, with values ranging from -1.00 to 1.00.

24

Figure 16: Visualizations of the 3D flow field around different car designs, showcasing the large variations in shape within the DrivAerNet++ dataset. The color indicates the normalized velocity u/u_∞, with values ranging from 0 to 1.2.

25

F.1 Dataset Annotations

In addition to the CFD simulation data, our dataset includes detailed annotations for various car components, such as wheels, side mirrors, and doors. These annotations are instrumental for a range of machine learning tasks, including classification, semantic segmentation, and object detection. The comprehensive labeling can also facilitate automated CFD meshing processes by providing precise information about different car components. By incorporating these labels, our dataset enhances the utility for developing and testing advanced algorithms in automotive design and analysis.

As illustrated in Figure 17, the dataset comprises annotations for 29 distinct car components, with each color representing a different part. This extensive annotation enables detailed analysis and optimization of aerodynamic performance, structural integrity, and aesthetic design, thus supporting a wide range of engineering and research applications.

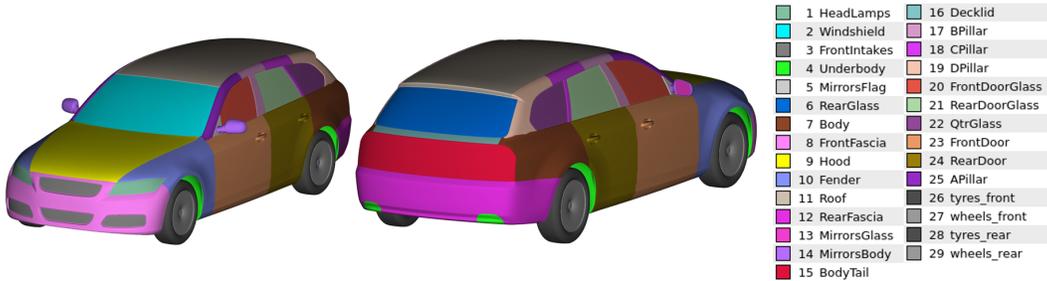



Figure 17: Annotated car components in the dataset, featuring 29 labels. Each color represents a different part of the car, including headlamps, windshield, front intakes, underbody, mirrors, rear glass, body panels, and wheels, providing a comprehensive labeling of the full car geometry.

G Deep Surrogate Models

G.1 RegDGCNN

In our previous work [24], we introduced the RegDGCNN model for an end-to-end aerodynamic drag prediction directly from 3D meshes. Our RegDGCNN model combines PointNet's spatial encoding [55] with graph CNNs' relational analysis to understand fluid dynamics. It uses edge convolution (EdgeConv) on dynamically updating graphs to capture complex fluid interactions, providing a novel approach for precise aerodynamic parameter estimation.

The RegDGCNN model is designed for regression tasks involving 3D point cloud data, leveraging graph-based convolutions to extract hierarchical features effectively [72]. The network architecture comprises several key components: EdgeConv layers utilize local neighborhood information to perform convolution operations on graph-structured data, with four EdgeConv layers featuring increasing channel sizes of 256, 512, 512, and 1024. Each layer is followed by batch normalization [37] and Leaky ReLU activation to stabilize and accelerate training. Dropout is applied in the fully connected layers to prevent overfitting, with a dropout rate of 0.4. After extracting features using EdgeConv layers, the network includes fully connected layers that progressively reduce the dimensionality of the feature vectors, with sizes of 128, 64, 32, and 16 neurons, respectively. The final output layer predicts the drag coefficient. Additionally, the network employs both global max pooling and global average pooling to aggregate features from all points in the point cloud, ensuring that the model captures the most critical information.

G.2 PointNet

We modified the original PointNet [55] network for the task of aerodynamic drag prediction. The modified model consists of a structured sequence of layers, beginning with an input layer for 3D coordinates, followed by six convolutional layers with increasing channel sizes: 64, 128, 256, 256, 512, and up to the embedding dimension, which we set to 1024. Each convolutional stage includes batch normalization [37] for enhanced training stability. A residual connection links the input directly

26

to the deeper layers, aiding gradient flow. The architecture concludes with three linear layers, reducing dimensions progressively from 1024 to 512, and finally to a single output value for the drag.

Compared to RegDGCNN [24], we scaled the number of point clouds up to 100k and 250k, which showed an increase in performance compared to the baseline implementation, typically using 1-5k points. The increased number of points provides a good representation of the actual geometry and can better capture small geometric modifications. The network is fully differentiable and can be integrated with downstream optimization tasks.

We also considered another variant of PointNet for the task of pressure prediction on the car surface (see Section I.1). The PointNet model for pressure prediction consists of an input layer for 3D coordinates, followed by three convolutional layers with increasing channel sizes: 64, 128, and 1024. A Spatial Transformer Network (STN) [40] is used to align the input points, enhancing the network's ability to learn spatial features.

G.3 Graph Neural Network

Here, we train a graph neural network (GNN) for predicting the aerodynamic drag. This model is implemented using the PyTorch Geometric library [27] and consists of several layers and components tailored for processing graph-structured data. The architecture of this model comprises several key components. The model begins with a series of four Graph Convolutional Network (GCN) layers, which use the *GCNConv* operator [44] for message passing to capture local structural information from the graph. These layers have increasing channel sizes of 64, 128, 128, and 256, respectively. Following the GCN layers, the model includes a fully connected network with three linear layers, which progressively reduce the feature dimensions to 128, 64, and finally to a single output value for drag prediction. ReLU activations and batch normalization [37] are used between the layers. To aggregate the node features into a graph-level representation, global mean pooling is applied, computing the mean of the node features for each graph in the batch.

The training was distributed across four NVIDIA A100 80GB GPUs, leveraging data parallelism for increased computational efficiency. The initial learning rate was set to 0.001, with a learning rate scheduler employed to reduce the rate when validation loss plateaued. Specifically, the *ReduceLROn-Plateau* scheduler was used, with a patience of 10 epochs and a reduction factor of 0.1. This strategy helped fine-tune the models by adjusting the learning rate based on validation performance. The training process spanned 100 epochs. The Adam optimizer [43] was chosen for its adaptive learning rate capabilities.

H Evaluation

We use various loss functions to assess model performance in predicting both the aerodynamic drag and the pressure values on the car's surface. Below are the key metrics:

Mean Squared Error (MSE): This metric quantifies the average of the squares of the errors, which are the differences between the ground truth drag coefficients from CFD simulations and the predictions made by the surrogate models. It is especially sensitive to large errors in prediction.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (C_{d_i} - \hat{C}_{d_i})^2 \qquad (3)$$

Mean Absolute Error (MAE): The MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is less sensitive to outliers than MSE.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |C_{d_i} - \hat{C}_{d_i}| \qquad (4)$$

Maximum Absolute Error (Max AE): The Max AE measures the largest absolute error among all prediction errors, highlighting the worst-case prediction accuracy.

$$Max\,AE = \max_{i=1}^{n} |C_{d_i} - \hat{C}_{d_i}| \qquad (5)$$

Coefficient of Determination (R^2 Score): The R^2 Score indicates the proportion of variance in the actual drag coefficients that is predictable from the model's predictions, with a value of 1 representing

27

a perfect fit.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(C_{d_i} - \hat{C}_{d_i})^2}{\sum_{i=1}^{n}(C_{d_i} - \bar{C}_d)^2} \tag{6}$$

Lower MSE and MAE values, along with a higher R^2 score and a lower Max MAE, indicate more accurate predictions of the aerodynamic drag coefficient by the surrogate models.

I Additional Experiments

In this section, we utilize our dataset for various tasks to demonstrate its broad applicability. We first perform a full 3D surface field prediction of the pressure field. Then, we explore and visualize the design space of DrivAerNet++ using dimensionality reduction techniques such as t-SNE [71].

I.1 3D Surface Field Prediction

Predicting the pressure distribution on the 3D mesh surface of a car is an essential task for aerodynamic analysis. Detailed pressure distribution data around a car's surface allows aerodynamics experts to design modifications that reduce drag, optimize lift and downforce, prevent flow separation, enhance cooling efficiency, and minimize noise [36]. Here, we employ geometric deep learning models for surface field prediction based on our dataset. As illustrated in Figure 18, our dataset, DrivAerNet++, provides a higher mesh resolution and more realistic car shapes compared to other datasets, with higher simulation fidelity.

In Figure 19, we compare the input 3D mesh used by a modified PointNet [55] and RegDGCNN [72, 24] for predicting surface fields. PointNet is trained on 100k points, while RegDGCNN is limited to 10k points due to memory constraints. The figure shows the initial input mesh, the predicted results from both models, and the differences between these predictions and the ground truth from high-fidelity CFD simulations. This comparison illustrates the effectiveness of both models in capturing surface field variations and highlights the discrepancies between the predictions and actual CFD results, providing insights into each model's strengths and limitations.

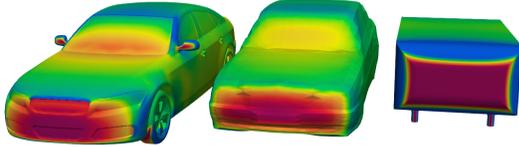



Figure 18: Pressure distribution on the car surface from three different datasets. The left car is from our dataset, DrivAerNet++, featuring realistic car shapes, higher mesh resolution, and greater simulation fidelity. The middle car is based on ShapeNet cars [69], and the right car uses the Ahmed body as presented in the study by Li et al. (2023) [45].

For interpolation, we used Radial Basis Function (RBF) interpolation. The results summarized in Table 10 present the L2 errors for the PointNet and RegDGCNN models in predicting surface fields. The L2 Error (Prediction vs Ground Truth) indicates how well the models' predictions align with the CFD data, while the L2 Error (Interpolated vs Full Mesh) evaluates the accuracy of the Radial Basis Function (RBF) interpolation against the full mesh data. For the PointNet model, the L2 error for predictions is 18.97% with 100k nodes, and the error increases to 24.84% with 350k nodes for the interpolated results. In comparison, the RegDGCNN model exhibits an L2 error of 24.66% with 10k nodes for predictions and 26.44% with 350k nodes for the interpolated mesh. These results suggest that the PointNet model demonstrates better prediction accuracy than the RegDGCNN model, particularly with a larger number of nodes, highlighting its effectiveness in capturing the geometric details necessary for accurate surface field predictions.

Table 10: L2 Errors for PointNet and RegDGCNN models in predicting surface fields.

Model	L2 Error (Prediction vs Ground Truth)	L2 Error (Interpolated vs Full Mesh)
PointNet	18.97% (100k nodes)	24.84% (350k nodes)
RegDGCNN	24.66% (10k nodes)	26.44% (350k nodes)

28

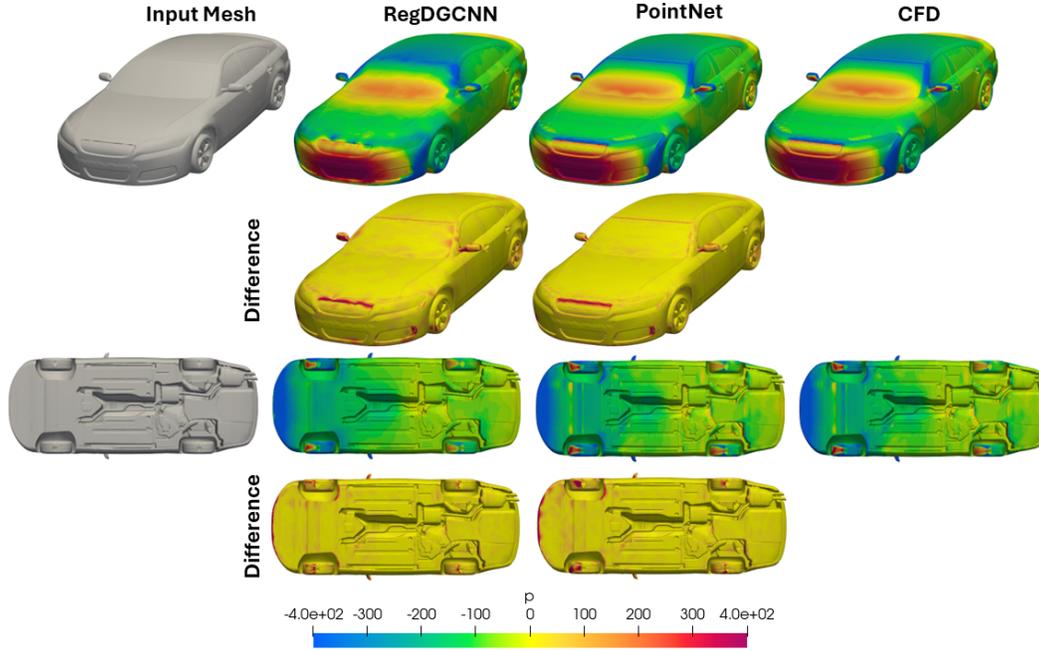



Figure 19: Comparison of the input 3D mesh to RegDGCNN and PointNet for surface field predictions. The figure shows the input mesh (left), the prediction results from RegDGCNN (middle left), PointNet (middle right), CFD results as ground truth (right), and the difference between the model predictions and the ground truth from CFD simulations (bottom row).

I.2 t-SNE Visualization of the Design Space

In this section, we train a PointNet [55] model for 3D shape classification and utilize the learned features to visualize the design space. By reducing the dimensionality of these features to 2D using t-SNE (t-distributed stochastic neighbor embedding) [71], we can create a plot that illustrates the similarity and variation between different car designs. This visualization aids designers in linking new designs to existing ones, potentially identifying the most similar designs and their performance using methods like K-Nearest Neighbors (KNN) [54]. Using t-SNE to visualize the high-dimensional design space provides an efficient means to explore the vast design space. This method eliminates the need to compute complex metrics, such as the Chamfer distance, between a new design and all existing designs. By performing inference on a new design with a pretrained model, extracting the learned features, reducing their dimensionality, and visualizing their position in 2D space, we can easily assess how similar or different a new design is compared to existing ones. Figure 20 demonstrates the t-SNE visualization, showing distinguished clusters for different car categories: Notchback, Estateback, and Fastback. Each point in the plot represents a unique design, illustrating the ability of t-SNE to capture and separate the underlying differences in car shapes effectively. In general, t-SNE is a powerful tool for reducing high-dimensional data to two or three dimensions while preserving the relative distances between data points, which helps in identifying patterns and similarities.

Furthermore, incorporating information about the aerodynamic performance of specific designs is crucial for analyzing new designs. Although Figure 20 focuses on shape similarities, the methodology can be extended to include performance values, providing a more comprehensive understanding of different designs. This combined approach enables a more thorough analysis of design efficiency and performance, facilitating better-informed decisions in the design process.

29

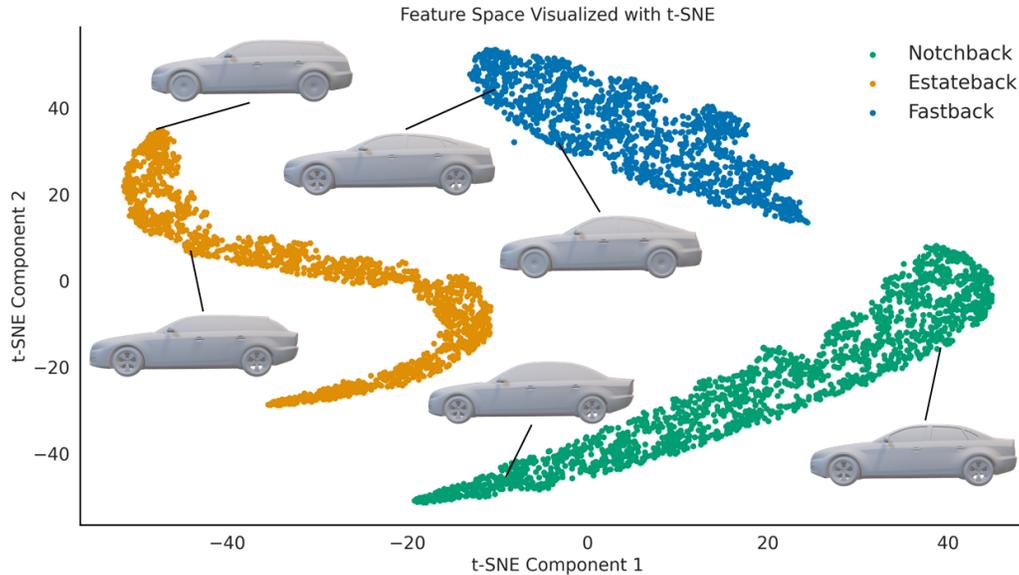



Figure 20: Feature space visualized with t-SNE for different car designs. The clusters represent three distinct car design categories: Notchback (green), Estateback (orange), and Fastback (blue). The t-SNE algorithm effectively separates the car designs into their respective categories, illustrating the ability of the feature representation to capture the underlying differences between the designs.

References

[1] Asad Abbas, Ashkan Rafiee, Max Haase, and Andrew Malcolm. Geometrical deep learning for performance prediction of high-speed craft. *Ocean Engineering*, 258:111716, 2022.

[2] S. R. Ahmed, G. Ramm, and G. Faltin. Some salient features of the time -averaged ground vehicle wake. *SAE Transactions*, 93:473–503, 1984. ISSN 0096736X. URL http://www.jstor.org/stable/44434262.

[3] Mubashara Akhtar, Omar Benjelloun, Costanza Conforti, Pieter Gijsbers, Joan Giner-Miguelez, Nitisha Jain, Michael Kuchnik, Quentin Lhoest, Pierre Marcenac, Manil Maskey, Peter Mattson, Luis Oala, Pierre Ruyssen, Rajat Shinde, Elena Simperl, Goeffry Thomas, Slava Tykhonov, Joaquin Vanschoren, Jos van der Velde, Steffen Vogler, and Carole-Jean Wu. Croissant: A metadata format for ml-ready datasets. DEEM '24, page 1–6, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706110. doi: 10.1145/3650203.3663326. URL https://doi.org/10.1145/3650203.3663326.

[4] Neil Ashton, A West, S Lardeau, and Alistair Revell. Assessment of rans and des methods for realistic automotive models. *Computers & fluids*, 128:1–15, 2016.

[5] Neil Ashton, Paul Batten, Andrew Cary, and Kevin Holst. Summary of the 4th high-lift prediction workshop hybrid rans/les technology focus group. *Journal of Aircraft*, pages 1–30, 2023.

[6] Tarek Ayman, Mayar A Elrefaie, Eman Sayed, Mohammed Elrefaie, Mahmoud Ayyad, Ahmed A Hamada, and Mohamed M Abdelrahman. Deep learning-based prediction of aerodynamic performance for airfoils in transonic regime. In *2023 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 157–160. IEEE, 2023.

[7] Pierre Baque, Edoardo Remelli, Francois Fleuret, and Pascal Fua. Geodesic convolutional shape optimization. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 472–481. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/baque18a.html.

[8] Florent Bonnet, Jocelyn Mazari, Paola Cinnella, and Patrick Gallinari. Airfrans: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier–stokes solutions. *Advances in Neural Information Processing Systems*, 35:23463–23478, 2022.

[9] Christian Brand, Jillian Anable, Ioanna Ketsopoulou, and Jim Watson. Road to zero or road to nowhere? disrupting transport and energy in a zero carbon world. *Energy Policy*, 139:111334, 2020.

30

[10] Adam Brandt, Henrik Berg, Michael Bolzon, and Linda Josefsson. The effects of wheel design on the aerodynamic drag of passenger vehicles. *SAE International Journal of Advances and Current Practices in Mobility*, 1(2019-01-0662):1279–1299, 2019.

[11] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[12] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[13] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[14] Wai Tong Chung, Bassem Akoush, Pushan Sharma, Alex Tamkin, Ki Sung Jung, Jacqueline Chen, Jack Guo, Davy Brouzet, Mohsen Talei, Bruno Savard, et al. Turbulence in focus: Benchmarking scaling behavior of 3d volumetric super-resolution with blastnet 2.0 data. *Advances in Neural Information Processing Systems*, 36, 2024.

[15] Adam Cobb, Anirban Roy, Daniel Elenius, Frederick Heim, Brian Swenson, Sydney Whittington, James Walker, Theodore Bapty, Joseph Hite, Karthik Ramani, et al. Aircraftverse: A large-scale multimodal dataset of aerial vehicle designs. *Advances in Neural Information Processing Systems*, 36:44524–44543, 2023.

[16] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL http://www.blender.org.

[17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[18] Guillaume Damblin, Mathieu Couplet, and Bertrand Iooss. Numerical studies of space-filling designs: optimization of latin hypercube samples and subprojection properties. *Journal of Simulation*, 7(4):276–289, 2013.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] Open MPI Documentation. *mpirun / mpiexec*, 2024. URL https://docs.open-mpi.org/en/v5.0.x/man-openmpi/man1/mpirun.1.html. Accessed: 2024-05-26.

[21] Thomas D Economon, Francisco Palacios, Sean R Copeland, Trent W Lukaczyk, and Juan J Alonso. Su2: An open-source suite for multiphysics simulation and design. *Aiaa Journal*, 54(3):828–846, 2016.

[22] Benet Eiximeno, Arnau Miró, Ivette Rodríguez, and Oriol Lehmkuhl. Toward the usage of deep learning surrogate models in ground vehicle aerodynamics. *Mathematics*, 12(7):998, 2024.

[23] Mohamed Elrefaie, Tarek Ayman, Mayar A Elrefaie, Eman Sayed, Mahmoud Ayyad, and Mohamed M AbdelRahman. Surrogate modeling of the aerodynamic performance for airfoils in transonic regime. In *AIAA SCITECH 2024 Forum*, page 2220, 2024.

[24] Mohamed Elrefaie, Angela Dai, and Faez Ahmed. Drivaernet: A parametric car dataset for data-driven aerodynamic design and graph-based drag prediction. *arXiv preprint arXiv:2403.08055*, 2024.

[25] Mohamed Elrefaie, Steffen Hüttig, Mariia Gladkova, Timo Gericke, Daniel Cremers, and Christian Breitsamter. Real-time and on-site aerodynamics using stereoscopic piv and deep optical flow learning. *arXiv preprint arXiv:2401.09932*, 2024.

[26] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[27] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[28] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

31

[29] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92, 2021.

[30] Christopher Greenshields. *OpenFOAM v11 User Guide*. The OpenFOAM Foundation, London, UK, 2023. URL https://doc.cfd.direct/openfoam/user-guide-v11.

[31] Erkan Gunpinar, Umut Can Coskun, Mustafa Ozsipahi, and Serkan Gunpinar. A generative design and drag coefficient prediction system for sedan car side silhouettes based on computational fluid dynamics. *CAD Computer Aided Design*, 111:65–79, 6 2019. ISSN 00104485. doi: 10.1016/j.cad.2019.02.003.

[32] Sheikh Md Shakeel Hassan, Arthur Feeney, Akash Dhruv, Jihoon Kim, Youngjoon Suh, Jaiyoung Ryu, Yoonjin Won, and Aparna Chandramowlishwaran. Bubbleml: A multiphase multiphysics dataset and benchmarks for machine learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Angelina Heft, Thomas Indinger, and Nikolaus Adams. Investigation of unsteady flow structures in the wake of a realistic generic car model. In *29th AIAA applied aerodynamics conference*, page 3669, 2011.

[34] Angelina I Heft, Thomas Indinger, and Nikolaus A Adams. Experimental and numerical investigation of the drivaer model. In *Fluids Engineering Division Summer Meeting*, volume 44755, pages 41–51. American Society of Mechanical Engineers, 2012.

[35] Angelina I Heft, Thomas Indinger, and Nikolaus A Adams. Introduction of a new realistic generic car model for aerodynamic investigations. Technical report, SAE Technical Paper, 2012.

[36] Wolf-Heinrich Hucho. *Aerodynamik des Automobils: eine Brücke von der Strömungsmechanik zur Fahrzeugtechnik*. Springer-Verlag, 2013.

[37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[38] M Islam, F Decker, E De Villiers, Aea Jackson, J Gines, T Grahs, A Gitt-Gehrke, and J Comas i Font. Application of detached-eddy simulation for automotive aerodynamics development. Technical report, SAE Technical Paper, 2009.

[39] Sam Jacob Jacob, Markus Mrosek, Carsten Othmer, and Harald Köstler. Deep learning for real-time aerodynamic evaluations of arbitrary vehicle shapes. *SAE International Journal of Passenger Vehicle Systems*, 15(2):77–90, mar 2022. ISSN 2770-3460. doi: https://doi.org/10.4271/15-15-02-0006. URL https://doi.org/10.4271/15-15-02-0006.

[40] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[41] Ali Kashefi and Tapan Mukerji. Physics-informed pointnet: A deep learning solver for steady-state incompressible flows and thermal fields on multiple sets of irregular geometries. *Journal of Computational Physics*, 468:111510, 2022.

[42] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[44] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[45] Zongyi Li, Nikola Borislavov Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3d pdes, 2023.

[46] H Martins, CO Henriques, JR Figueira, CS Silva, and AS Costa. Assessing policy interventions to stimulate the transition of electric vehicle technology in the european union. *Socio-Economic Planning Sciences*, 87: 101505, 2023.

[47] Florian R Menter, Martin Kuntz, Robin Langtry, et al. Ten years of industrial experience with the sst turbulence model. *Turbulence, heat and mass transfer*, 4(1):625–632, 2003.

32

[48] Peter Mock and Sonsoles Díaz. Pathways to decarbonization: the european passenger car market in the years 2021–2035. *communications*, 49:847129–848102, 2021.

[49] Chair of Aerodynamics and Technical University of Munich Fluid Mechanics. Drivaer model geometry. `https://www.epc.ed.tum.de/en/aer/research-groups/automotive/drivaer/geometry/`, 2024. Accessed: 2024-05-21.

[50] OpenFOAM Foundation. *Meshing with snappyHexMesh*, 2023. URL `https://www.openfoam.com/documentation/guides/latest/doc/guide-meshing-snappyhexmesh.html`. Accessed: 2024-06-05.

[51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[52] Giancarlo Pavia and Martin Passmore. Characterisation of wake bi-stability for a square-back geometry with rotating wheels. In *Progress in Vehicle Aerodynamics and Thermal Management: 11th FKFS Conference, Stuttgart, September 26-27, 2017 11*, pages 93–109. Springer, 2018.

[53] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020.

[54] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[55] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[56] Peng Qin, Alessio Ricci, and Bert Blocken. Cfd simulation of aerodynamic forces on the drivaer car model: Impact of computational parameters. *Journal of Wind Engineering and Industrial Aerodynamics*, 248:105711, 2024. ISSN 0167-6105. doi: https://doi.org/10.1016/j.jweia.2024.105711. URL `https://www.sciencedirect.com/science/article/pii/S0167610524000746`.

[57] Edoardo Remelli, Artem Lukoianov, Stephan Richter, Benoit Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. Meshsdf: Differentiable iso-surface extraction. *Advances in Neural Information Processing Systems*, 33:22468–22478, 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/fe40fb944ee700392ed51bfe84dd4e3d-Paper.pdf`.

[58] Thiago Rios, Patricia Wollstadt, Bas Van Stein, Thomas Back, Zhao Xu, Bernhard Sendhoff, and Stefan Menzel. Scalability of learning tasks on 3d cae models using point cloud autoencoders. pages 1367–1374. Institute of Electrical and Electronics Engineers Inc., 12 2019. ISBN 9781728124858. doi: 10.1109/SSCI44817.2019.9002982.

[59] Thiago Rios, Bas Van Stein, Thomas Back, Bernhard Sendhoff, and Stefan Menzel. Point2ffd: Learning shape representations of simulation-ready 3d models for engineering design optimization. pages 1024–1033. Institute of Electrical and Electronics Engineers Inc., 2021. ISBN 9781665426886. doi: 10.1109/3DV53792.2021.00110.

[60] Thiago Rios, Bas van Stein, Patricia Wollstadt, Thomas Bäck, Bernhard Sendhoff, and Stefan Menzel. Exploiting local geometric features in vehicle design optimization with 3d point cloud autoencoders. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 514–521, 2021. doi: 10.1109/CEC45853.2021.9504746.

[61] Thomas Schütz. *Hucho-Aerodynamik des Automobils: Strömungsmechanik, Wärmetechnik, Fahrdynamik, Komfort*. Springer-Verlag, 2013.

[62] Shengrong Shen, Tian Han, and Jiachen Pang. Car drag coefficient prediction using long–short term memory neural network and lasso. *Measurement*, 225:113982, 2024.

[63] Binyang Song, Chenyang Yuan, Frank Permenter, Nikos Arechiga, and Faez Ahmed. Surrogate modeling of car drag coefficient with depth and normal renderings. *arXiv preprint arXiv:2306.06110*, 2023.

[64] D Brian Spalding. The numerical computation of turbulent flow. *Comp. Methods Appl. Mech. Eng.*, 3:269, 1974.

[65] Guocheng Tao, Chengwei Fan, Wen Wang, Wenjun Guo, and Jiahuan Cui. Multi-fidelity deep learning for aerodynamic shape optimization using convolutional neural network. *Physics of Fluids*, 36(5), 2024.

33

[66] Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36, 2020.

[67] Artur Toshev, Gianluca Galletti, Fabian Fritz, Stefan Adami, and Nikolaus Adams. Lagrangebench: A lagrangian fluid mechanics benchmarking suite. *Advances in Neural Information Processing Systems*, 36, 2024.

[68] Thanh Luan Trinh, Fangge Chen, Takuya Nanri, and Kei Akasaka. 3d super-resolution model for vehicle flow field enrichment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5826–5835, 2024.

[69] Nobuyuki Umetani and Bernd Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics*, 37, 2018. ISSN 15577368. doi: 10.1145/3197517.3201325.

[70] Muhammad Usama, Aqib Arif, Farhan Haris, Shahroz Khan, S. Kamran Afaq, and Shahrukh Rashid. A data-driven interactive system for aerodynamic and user-centred generative vehicle design. In *2021 International Conference on Artificial Intelligence (ICAI)*, pages 119–127, 2021. doi: 10.1109/ICAI52203.2021.9445243.

[71] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[72] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[73] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

[74] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[75] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 160–176. Springer, 2016.

34

Datasheets for Datasets

In line with contemporary practices in the deep learning field, we employ datasheets to document our dataset's intent, structure, collection, preprocessing, labeling, usage, distribution, and maintenance, adhering to the established methodology referenced in [29].

Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

The DrivAerNet++ dataset was developed to serve as a benchmark for training deep learning models. This dataset, characterized by its extensive collection of 3D shapes, leverages k-ω-SST RANS model to ensure high-fidelity turbulence modeling. Its significance lies in addressing the scarcity of publicly available, industry-grade datasets for training deep learning models in engineering design and CFD.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The dataset was created collaboratively by the Design Computation and Digital Engineering (De-CoDE) Lab at the Massachusetts Institute of Technology (MIT), the 3D AI Lab at the Technical University of Munich (TUM), and BETA CAE SYSTEMS USA, Inc.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

The creation of this dataset was significantly supported by computational resources provided by MIT.

Composition

What do the instances that comprise the dataset represent? Are there multiple types of instances? Please provide a description. Each entry in the DrivAerNet++ dataset encompasses comprehensive 3D fields of velocity, pressure, wall-shear stresses, and aerodynamic coefficients, along with detailed 3D meshes of car bodies, front, and rear wheels. The fluid dynamics data is stored in `.vtk` files, while the geometric meshes are kept in `.stl` format, all optimized

for high-speed data access. The dataset also includes parametric data saved in tabular format, point clouds, and part annotations as well as class labels for different car categories. To assist in efficient data utilization, a Python script is provided, enabling the conversion of these datasets into a format readily usable for machine learning applications, ensuring a smooth integration into various analysis pipelines.

How many instances are there in total (of each type, if appropriate)?

DrivAerNet++ contains a total of 8,000 3D full-domain CFD simulations.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

Yes, the DrivAerNet++ dataset encompasses the complete 3D spatial domain as defined during the simulation process.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

Each instance primarily consists of velocity and pressure fields, turbulence metrics, boundary conditions, meshes, and log files. Additionally, it features derived data from functions such as `forceCoeffsIncompressible` for aerodynamic coefficients, `residuals` for equation residuals, and `wallShearStress`, along with visualization outputs like `streamlinesSphere` and `cutPlaneSurface` for flow and pressure visualization, offering a comprehensive overview of the fluid dynamics simulation.

Is there a label or target associated with each instance? If so, please provide a description.

Yes, all the aerodynamic coefficients (C_d, C_l, $C_{l,r}$, $C_{l,f}$, and C_m) are clearly labeled and ready for use in regression tasks. Furthermore, the flow-field data can be leveraged to generate additional labels for specific applications, such as the total pressure coefficient C_{pt} and other quantities of interest. In addition, cars are labeled by design

35

category and for each car we provide the labels of the components (part annotations).

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

Yes, DrivAerNet++ is designed as a benchmark dataset for ML applications, complete with a split into training, validation, and test sets to ensure fair comparisons among various machine and deep learning models. Details on the data splits can be found here: `https://github.com/Mohamedelrefaie/DrivAerNet`.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

Yes. Reynolds-Averaged Navier-Stokes (RANS) turbulence models have been utilized. Despite the employment of these models, inherent limitations and approximations introduce potential inaccuracies. RANS models, particularly effective for a broad spectrum of flow conditions, may not capture the full range of turbulent scales and complex flow phenomena with absolute precision. Drawbacks stem from the models' reliance on empirical data and simplifying assumptions, which can lead to discrepancies in predicting highly unsteady or anisotropic turbulent flows. These limitations necessitate cautious interpretation of the data, especially in scenarios where turbulence plays a critical role. See [38] and [4] for more details.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

Our dataset is self-contained.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)? If so, please provide a description.

No.

Collection Process

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

For the DrivAerNet++ dataset, we predominantly utilized open-source tools to enable researchers to replicate the methodology and approach. The simulations were conducted using OpenFOAM® [30], while SnappyHexMesh was employed for mesh generation. Geometry processing was handled by Blender [16] and the commercial software ANSA®. Python scripts facilitated the automation of the processes, and ParaView was used for visualization and data analysis. Although we used the commercial software ANSA® for creating the parametric model and conducting the design of experiments, we are releasing the scripts used in ANSA® along with the parametric models. This ensures that results can be replicated using any CAD modeling software, facilitating reproducibility.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

The DrivAerNet++ dataset was developed and refined solely by the authors of this work without the involvement of crowdworkers or contractors, and no extra compensation was provided beyond the usual salary and stipend.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.

The data was collected in Q3, and Q4 of 2023 as well as Q1, and Q2 of 2024.

Preprocessing/cleaning/labeling

36

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

Yes, preprocessing was essential due to the wide range of morphing and geometry modifications applied. Some of the resulting geometries were unsuitable for CFD simulations due to issues such as intersecting faces and non-manifold edges. Additionally, some modifications led to unrealistic car shapes that deviated significantly from practical automotive designs. To address these issues, we implemented a validation process to identify and exclude such problematic geometries, ensuring that only feasible designs were included for simulations.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.

Yes, the raw data from the OpenFOAM® simulations were saved. However, due to the considerable size of the dataset, it will be shared upon request and subject to approval from MIT. This ensures accessibility while maintaining proper management and oversight.

Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.

Yes, we employed open-source tools including Python, Blender, and ParaView for preprocessing, cleaning, and labeling the dataset instances.

Uses

Has the dataset been used for any tasks already? If so, please provide a description.

Yes, the dataset has been utilized in our study focusing on the surrogate modeling of the aerodynamic drag and surface fields prediction, with findings documented in the paper.

What (other) tasks could the dataset be used for?

The DrivAerNet++ dataset could be utilized for a variety of tasks including data-driven aerodynamic design optimization, turbulence modeling, reduced-order modeling, generative AI, dimensionality reduction, and aerodynamic design analysis.

Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

Yes, the dataset is publicly available at `https://github.com/Mohamedelrefaie/DrivAerNet`

How will the dataset will be distributed (e.g., tarball on website, API, GitHub) Does the dataset have a digital object identifier (DOI)?

A Digital Object Identifier (DOI) will be assigned and shared alongside the dataset release on GitHub.

When will the dataset be distributed?

The complete dataset will be released after our study has been peer-reviewed and accepted for publication. Meanwhile, a smaller portion of the dataset will be released for initial public use and for replicating the results in the main paper.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

The dataset is provided under the Creative Commons Attribution-NonCommercial (CC BY-NC) license, allowing use and modification for non-commercial purposes provided appropriate credit is given. No fees are associated with its use. Full terms can be found at `https://creativecommons.org/licenses/by-nc/4.0/deed.en`, ensuring accessibility while acknowledging creators' contributions.

Maintenance

Who will be supporting/hosting/maintaining the dataset?

37

The dataset is maintained by the DeCoDE Lab at MIT. The DrivAerNet team is committed to maintaining the dataset by addressing issues reported on our GitHub issues page. The dataset will be hosted on the Harvard Dataverse Repository for long-term availability, and our data management approach aligns with the FAIR principles [73] for responsible scientific data handling.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

The DrivAerNet++ team can be reached at `https://decode.mit.edu/projects/DrivAerNet/`. Users are encouraged to submit inquiries to the GitHub page. For further inquiries and collaborations, Mohamed Elrefaie can be contacted directly.

Is there an erratum? If so, please provide a link or other access point.

Yes. A comprehensive issue-tracking system is available at `https://github.com/Mohamedelrefaie/DrivAerNet/issues`.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

Yes, the team plans to issue minor updates for the dataset to address any discovered errors. These updates will be announced on our GitHub page.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

Yes, we welcome contributions through creating pull requests at `https://github.com/Mohamedelrefaie/DrivAerNet/pulls`. All submissions will be reviewed for accuracy, and reliability before being added to the dataset. Changes from new contributions will be recorded in our version history.

38