

SISSA

Scuola
Internazionale
Superiore di
Studi Avanzati

Mathematics Area – PhD course in Mathematical Analysis, Modelling, and
Applications

Data-Driven Geometric Representation and Flow-Field Prediction in Industrial Fluid Dynamics

Candidate: Anouar Dahdah

Advisor: Prof. Gianluigi Rozza

Co-advisors:

Dr. Federico Pichi

Dr. Niccolò Tonicello

Dr. Giuseppe Alessio D’Inverno

Academic Year 2025–26



Abstract

High-fidelity computational fluid dynamics (CFD) simulations require significant resources, especially in the early stages of vehicle aerodynamic design when many shape variations are assessed. Data-driven surrogate models can lower these computational costs, but their performance often depends on the geometric representation used during training, such as fixed design parameters, a specific computational mesh, or a particular spatial discretization. As a result, it is difficult to explore a wide range of designs or use the same surrogate models across different vehicle types and design processes. A two-stage aerodynamic surrogate framework is developed based on an explicit signed distance field (SDF) representation of geometry. In the first stage, geometry is reconstructed using a compact latent representation. For AhmedML, a parameter-driven branch maps design variables to the latent space, while a volumetric encoder offers a geometry-driven alternative based directly on the SDF. Both branches utilize a shared geometric decoder. To accommodate more realistic vehicle configurations, DrivAerML and Siemens geometries are processed through the volumetric SDF approach. This methodology enables both public and industrial vehicle geometries to be represented through a unified spatial interface without the need for a shared low-dimensional design parameterization. Stage 2 maps the reconstructed or reference signed distance function (SDF) to three-dimensional aerodynamic fields using a volumetric U-Net. The direct configuration is trained with reference SDFs, while the end-to-end configuration is trained with geometries reconstructed by Stage 1. Stage 1 is trained initially and then kept fixed, ensuring that the aerodynamic model is exposed during training to the same type of geometric input as during end-to-end inference. Consequently, the two stages are optimized sequentially rather than through joint backpropagation. Drag is modeled separately using a scalar regression approach when appropriate supervision is available. The framework is evaluated using geometries of increasing complexity, including analytical shapes, AhmedML, DrivAerML, and industrial Siemens vehicle configurations. Results indicate that the geometric stage reconstructs vehicle representations with high volumetric and surface agreement prior to aerodynamic prediction. The subsequent field-prediction stage maintains the principal flow structures across all datasets. Comparison between direct and end-to-end configurations enables identification of the impact of geometric reconstruction on downstream aerodynamic accuracy. Transfer experiments further demonstrate that the shared SDF interface allows adaptation of the staged methodology from benchmark geometries to realistic and industrial vehicle families without dependence on a common design-variable space. The primary contribution of this thesis is methodological. The proposed framework distinguishes geometry reconstruction from aerodynamic prediction by introducing an explicit, independently testable geometric interface while maintaining a consistent volumetric representation across diverse vehicle families. This distinction allows for independent and joint analysis of geometric and aerodynamic errors throughout the entire design-to-geometry-to-flow

process. By integrating both parameter-driven and geometry-driven entry paths with a unified signed distance function (SDF) representation, the framework supports rapid aerodynamic evaluation and facilitates broader exploration of vehicle design spaces where a single common parameterization is not feasible.

Contents

Contents	i
List of Figures	v
List of Tables	xi
List of Acronyms	xiii
Nomenclature	xv
1 Introduction	1
1.1 Framework and State of the Art	1
1.2 Research Questions and Contributions	4
1.3 Structure of the Thesis	6
2 Physical and Mathematical Preliminaries	7
2.1 Aerodynamics of Road Vehicles	7
2.1.1 Characteristic Flow Structures in the Benchmark Geometries	11
2.2 Reduced-Order Models and Surrogates	13
2.3 Convolutional Autoencoders for Field Prediction	16
2.4 Geometry-Aware Learning Beyond Fixed Discretizations	18
2.5 The Signed Distance Function as Geometric Interface	19
2.6 Mathematical Formulation of the Surrogate Problem	20
2.7 Chapter Summary	23
3 Methodology	25
3.1 Deep learning for aerodynamic prediction: state of the art	25
3.2 From CFD simulations to training data	26
3.3 Metrics and evaluation protocol	30
3.4 The two-stage prediction architecture	33

4	Learning Geometry: A Latent Representation of Vehicle Shapes	43
4.1	Evaluation of the Reconstructed Surface	44
4.2	Pipeline Validation on Canonical Geometries	44
4.2.1	Three-dimensional reconstruction of spheres and ellipsoids	45
4.2.2	Cross-sectional reconstruction	45
4.2.3	Training behavior	47
4.2.4	Architecture comparison	47
4.2.5	Generalization across the ellipsoid family	49
4.2.6	Canonical geometry findings	49
4.3	Ahmed-Body Geometry	50
4.3.1	Design parameters of the family	50
4.3.2	Effect of the activation-loss parameter	51
4.3.3	Surface accuracy over the Ahmed-body model-selection/evaluation set	53
4.3.4	Longitudinal reconstruction	54
4.3.5	Three-dimensional reconstruction	55
4.4	Realistic Vehicle Geometry: DrivAer	56
4.4.1	Surface metrics	56
4.4.2	Longitudinal reconstruction	58
4.4.3	Three-dimensional reconstruction of a DrivAer example	59
4.4.4	Parameter input reconstruction of DrivAerML	59
4.4.5	DrivAer findings	61
4.5	Industrial Siemens Geometries	61
4.5.1	Surface metrics	61
4.5.2	Longitudinal reconstruction	61
4.5.3	Three-dimensional reconstruction	63
4.5.4	Siemens findings	63
4.6	Chapter Summary	63
5	Predicting Flow Fields and Drag	65
5.1	Volumetric field prediction on the Ahmed body	65
5.2	Volumetric field prediction on DrivAerML	68
5.3	Volumetric field prediction on Siemens geometries	71
5.4	Drag coefficient prediction	73
5.5	Discussion	75
6	The End-to-End Surrogate	77
6.1	From geometry to flow field: the complete pipeline	77
6.2	End-to-end accuracy on DrivAerML	81
6.2.1	Mixed-transfer volumetric Stage 1 route	81
6.2.2	DrivAerML-only parameter-driven Stage 1 route	81
6.3	End-to-end accuracy on Siemens	85
6.4	Discussion	87

6.5	Comparison with the published state of the art	88
6.6	Toward deployment: accuracy requirements and inference cost	89
7	Conclusion and Perspectives	91
7.1	Contribution	91
7.2	Limitations	92
7.3	Perspectives	93
	Bibliography	95

List of Figures

2.1	Structure of the Free-Form Deformation map $\mathcal{M}(\cdot; \mu)$ as the composition of the map Π onto the reference configuration, the Bernstein-polynomial deformation $\hat{T}(\cdot; \mu)$ driven by the displacement of the control points, and the back-map Π^{-1} to the physical domain. Only the control points of the top lattice row are displaced here, so that the effect of a single group of design parameters on the embedded geometry remains visible.	15
3.1	The two-stage surrogate. Stage 1 predicts the signed distance field. In the E2E configuration, reconstructed Stage 1 SDFs are used already during Stage 2 U-Net training/fine-tuning and again at inference, so the flow network is exposed to the same type of geometric input in both phases.	34
3.2	The geometric stage.	37
3.3	The predictive stage. The signed distance field enters as a single channel; multiscale geometric features are extracted by the contracting path and returned to full resolution by the expanding path, with skip connections preserving high-resolution spatial detail, yielding the four-channel volumetric output. The drag coefficient is regressed from pooled encoder features.	39
4.1	Three-dimensional ground-truth and predicted surfaces for representative canonical geometries.	46
4.2	Cross-sectional reconstruction for representative canonical geometries. The first and third rows correspond to spherical designs of different sizes, while the second row corresponds to an ellipsoid. The columns show the reference field, reconstructed field, and absolute difference. The black and white contours indicate the reference and predicted zero level sets.	47
4.3	Training and validation losses for the spherical geometries reconstruction experiment on a logarithmic scale.	48

4.4	Architecture comparison on spherical geometries shape prediction.	48
4.5	Generalization across the ellipsoid family. Training, validation, and test relative errors are reported as a function of the amount of training data.	49
4.6	The six AhmedML design parameters visualization.	51
4.7	Reconstruction of one Ahmed-body evaluation design under each activation-loss setting, together with the ground-truth (GT) geometry.	52
4.8	Sensitivity of the Ahmed-body reconstruction to the activation-loss parameter ε_{act} . Each value corresponds to a separately trained Stage 1 configuration. The metrics are averaged over the 57 Ahmed model-selection/evaluation designs.	52
4.9	Dice coefficient and normalized 95th-percentile Hausdorff distance, HD_{95} , over the 57 Ahmed model-selection/evaluation designs, sorted by Dice. The mean values are 0.989 and 0.0044. . . .	53
4.10	Longitudinal reconstruction for Ahmed evaluation designs 146, 153, and 337. Each row shows the reference SDF, reconstructed SDF, and absolute error with reference and predicted zero contours. The adopted sign convention assigns negative values to the solid interior, zero to the body surface, and positive values to the exterior fluid region.	54
4.11	Reference and reconstructed zero-isosurfaces for Ahmed geometry-evaluation design 146. The reconstruction retains the complete three-dimensional body, its global proportions, ground clearance, and rear treatment.	55
4.12	DrivAerML geometric design parameters. Adapted from the DrivAerML dataset website [16] and the dataset described in [7].	56
4.13	Dice coefficient and normalized 95th-percentile Hausdorff distance, HD_{95} , for the 9 DrivAerML Stage 1 evaluation cases, sorted by Dice. The mean Dice coefficient is 0.962, and the mean normalized HD_{95} is 0.0070. The latter corresponds to approximately 78 mm before normalization by the computational-box diagonal.	57
4.14	Longitudinal reconstruction for DrivAerML Stage 1 evaluation designs 2, 19, and 51. Each row shows the reference field, reconstructed field, and absolute error with zero-contour overlays. The global vehicle envelope is preserved.	58
4.15	Reference and reconstructed zero-isosurfaces for DrivAerML Stage 1 evaluation design 2.	59
4.16	Dice coefficient and normalized 95th-percentile Hausdorff distance, HD_{95} , for the 10 DrivAerML test designs evaluated with the parameter-driven Stage 1 branch. The mean Dice coefficient is 0.928, and the mean normalized HD_{95} is 0.0072.	60

4.17	Parameter-driven Stage 1 reconstruction for DrivAerML design 114. The panels show the reference SDF, the SDF reconstructed from the DrivAerML morphing parameters, and the corresponding absolute SDF error on a span-normal cut. The black contours indicate the zero level sets defining the reference and reconstructed vehicle surfaces.	60
4.18	Dice coefficient and normalized 95th-percentile Hausdorff distance, HD_{95} , for the five test set Siemens designs, sorted by Dice. The mean values are 0.969 and 0.0069, respectively.	62
4.19	Longitudinal reconstruction for Siemens configurations Design_5_2, Design_6_2, and Design_7_2. Each row shows the reference field, reconstructed field, and absolute error with zero-contour overlays. The principal industrial vehicle profile is preserved, while the largest localized residuals occur near front structures, the lower body, and small components.	62
4.20	Reference and reconstructed zero-isosurfaces for Siemens configuration Design_6_2. The reconstruction preserves the complete industrial vehicle envelope and its principal exterior features. . .	63
5.1	Per-design relative L^2 errors for the AhmedML direct Stage 2 evaluation. The reference SDF is used as input. The figure complements the 45-design mean values of Table 5.1 by showing the variation across individual test set designs.	66
5.2	Direct Stage 2 prediction for a selected test set Ahmed design longitudinal slice. Rows: spanwise velocity, streamwise velocity, vertical velocity, and pressure. The streamwise field is displayed as $u_y - U_\infty$ to make the wake deficit visible; the reported relative error is computed on the full field. Columns show the input distance field, reference, prediction, and absolute error.	68
5.3	Representative DrivAerML direct-flow visualization on a longitudinal slice near the symmetry plane. Rows show the streamwise, spanwise, and vertical velocity components followed by pressure; columns show reference, prediction from the reference SDF, and absolute error. The visualization complements the quantitative test set evaluation reported in the text.	69
5.4	Second representative DrivAerML direct-flow visualization, using the same layout as Figure 5.3. The largest residuals are concentrated around the wake onset, rear shear layer, and underbody. .	70
5.5	Per-design relative L^2 errors for the DrivAerML direct Stage 2 evaluation. The curve shows the variation across the 10 test set flow designs and complements the mean values reported in Table 5.1.	71

5.6	Representative Siemens direct-flow visualization for Design_5_2 on a longitudinal mid-span slice. Rows show the three velocity components and pressure; columns show the reference field, prediction from the reference SDF, and absolute error. This configuration is not among the ten test configurations of Table 5.1, and it differs from the configuration shown for the end-to-end pipeline in Figure 6.11; the two figures should therefore not be read as a paired comparison.	72
5.7	Per-design relative L^2 errors for the Siemens direct Stage 2 evaluation. The curve shows the distribution across the 10 test set industrial flow configurations and complements the mean values reported in Table 5.1.	73
5.8	AhmedML direct drag-coefficient parity. Predicted and reference C_D values are shown for the training, validation, and test set drag groups.	74
5.9	DrivAerML direct drag-coefficient parity under leave-one-out scalar regression over the 97-design scalar-regression working set. The Stage 2 feature extractor remains fixed across folds.	75
6.1	Per-design relative L^2 errors for the AhmedML end-to-end evaluation over 45 test set flow designs. Stage 2 receives the SDF reconstructed by Stage 1. The designs are displayed in the worst-to-best ordering used by the evaluation figure, and the legend reports the corresponding mean errors.	79
6.2	End-to-end flow prediction for Ahmed design 146. Each row contains the CFD reference, the end-to-end prediction, and the absolute-error map. The displayed relative L^2 errors are 2.70% for $ \mathbf{U} $ and 9.12% for P	80
6.3	AhmedML drag-coefficient parity for the E2E configuration on the test set scalar-regression group.	81
6.4	Per-design relative L^2 errors for the DrivAerML end-to-end evaluation over 10 test set flow designs. The DrivAerML geometry is represented as an SDF and enters the volumetric Stage 1 branch; the reconstructed SDF is then supplied to the E2E Stage 2 U-Net. The design-by-design errors are displayed in the sorted order used by the evaluation figure.	82
6.5	End-to-end flow prediction for DrivAerML design 2. Each row contains the CFD reference, the end-to-end prediction, and the absolute-error map. The displayed relative L^2 errors are 3.97% for $ \mathbf{U} $ and 16.59% for P	83

6.6	Per-design relative L^2 errors for the DrivAerML parameter-driven end-to-end experiment over 10 test designs. Stage 1 receives the DrivAerML morphing parameters and reconstructs the SDF before Stage 2 predicts the aerodynamic fields. For each reported quantity, the 10 errors are independently ordered from worst to best; consequently, the same rank on the horizontal axis does not necessarily correspond to the same design for different quantities. The mean errors are 33.96% for U_x , 7.92% for U_y , 36.56% for U_z , 28.07% for P , and 7.97% for $ \mathbf{U} $	84
6.7	Parameter-driven end-to-end velocity-magnitude prediction for DrivAerML design 114 on a span-normal cut. The panels show the CFD reference, the Stage 2 prediction obtained from the parameter-reconstructed SDF, and the corresponding absolute error. The displayed relative L^2 error for $ \mathbf{U} $ is 8.43%.	84
6.8	Parameter-driven end-to-end pressure prediction for DrivAerML design 114 on a span-normal cut. The panels show the CFD reference, the Stage 2 prediction obtained from the parameter-reconstructed SDF, and the corresponding absolute error. The displayed relative L^2 error for pressure is 28.91%.	85
6.9	DrivAerML E2E drag-coefficient parity under leave-one-out scalar regression over the 97-design scalar-regression working set. . . .	85
6.10	Per-design relative L^2 errors for the Siemens end-to-end evaluation over 10 test set industrial flow configurations. The geometry-derived SDF is reconstructed through the volumetric Stage 1 branch before being supplied to the E2E Stage 2 network. The configurations are displayed in the sorted order used by the evaluation figure. The legend reports mean errors of 34.1% for U_x , 11.1% for U_y , 37.0% for U_z , 26.7% for pressure, and 10.9% for velocity magnitude.	86
6.11	End-to-end flow prediction for Siemens configuration Design_6_2. Each row contains the CFD reference, the end-to-end prediction, and the absolute-error map. The displayed relative L^2 errors are 12.50% for $ \mathbf{U} $ and 26.27% for P	87

List of Tables

3.1	Task-specific training and evaluation protocol. Training and evaluation groups are defined separately for geometry reconstruction, direct flow prediction, end-to-end flow prediction, and drag regression. The AhmedML geometry task uses all 380 configurations; its flow and drag tasks use the 299 retained after excluding incomplete field data.	34
3.2	Layer structure of the reconstruction network. Output shapes are given as channels $\times$ spatial resolution for a single design.	37
3.3	Layer structure of the aerodynamic network. Output shapes are given as channels $\times$ spatial resolution for a single design.	39
3.4	Reference configurations of Stage 1, Stage 2, and the drag regressor.	42
4.1	Design parameters of the canonical families and the intervals from which they are sampled. All lengths are expressed in the units of the fixed computational box.	45
4.2	Geometric parameters of the AhmedML family and the intervals from which they are sampled [6].	50
4.3	Design parameters of the DrivAerML family and the intervals from which they are sampled, expressed as changes relative to the baseline geometry [7]. All entries are control-point displacements of the morphing boxes, in millimetres, except the vehicle pitch, which is a rotation in degrees.	57
5.1	Direct Stage 2 field-prediction results. The reference SDF is supplied to the U-Net, and the values are mean relative L^2 errors over the test set flow designs of each dataset.	66

6.1	Direct and end-to-end field prediction. Values are mean relative L^2 errors in percent. U_{vec} denotes the relative L^2 error of the complete three-component velocity vector. For DrivAerML, the end-to-end row corresponds to the mixed realistic-vehicle configuration using the volumetric Stage 1 route. The direct and end-to-end rows use different fine-tuning sizes and different 10-design evaluation groups and therefore do not form a paired comparison. The additional DrivAerML-only parameter-driven configuration is reported separately in Section 6.2.2.	78
6.2	Drag regressed from the design parameters and from the learned descriptor, on identical designs and with the same regressor ensemble.	79
6.3	DrivAerML drag-coefficient accuracy under leave-one-out scalar regression over the 97-design scalar-regression working set. The scalar regressor is refitted on 96 designs in each fold, while the learned neural representations remain fixed.	84
6.4	Contextual comparison with published AB-UPT results on AhmedML and DrivAerML. Values are relative L^2 errors in percent; lower is better. U_{vec} denotes the complete three-component velocity-vector error. The DrivAerML present-work row corresponds to the mixed-transfer volumetric end-to-end configuration of Table 6.1; the additional parameter-driven DrivAerML-only experiment is reported separately in Section 6.2.2. The numerical values are not treated as a strict head-to-head ranking because the methods use different spatial discretizations and evaluation domains.	88
6.5	Cost of one aerodynamic evaluation. Reference costs are those reported for the public datasets [6, 7]; the surrogate is timed on a single GPU. The hardware is different and the table must not be interpreted as a normalized speed-up.	90

List of Acronyms

Acronym	Meaning
AB-UPT	Anchored-Branched Universal Physics Transformers
CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
CGNS	CFD General Notation System
CNN	Convolutional Neural Network
CPU	Central Processing Unit
E2E	End-to-End
FFD	Free-Form Deformation
FNN	Fully Connected Neural Network
GPU	Graphics Processing Unit
HD95	95th-percentile Hausdorff distance, denoted HD_{95} in equations
LES	Large-Eddy Simulation
LOO	Leave-One-Out
MAE	Mean Absolute Error
MLP	Multilayer Perceptron
MSE	Mean Squared Error
POD	Proper Orthogonal Decomposition
RANS	Reynolds-Averaged Navier–Stokes
ROM	Reduced-Order Model
SciML	Scientific Machine Learning
SDF	Signed Distance Function and, where the distinction is immaterial, its discrete sampling, the signed distance field; see Section 2.5

Nomenclature

Symbol	Meaning
<i>Coordinates and aerodynamic fields</i>	
x, y, z	Cartesian coordinates: x spanwise/lateral, y streamwise, and z vertical
$\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$	Cartesian unit vectors; $\mathbf{e}_y$ is the streamwise direction
$\mathbf{u}$	Continuous velocity field with components u_x, u_y, u_z
U_x, U_y, U_z	Discrete velocity fields used by the surrogate; U_y is streamwise, U_x spanwise, and U_z vertical
$ \mathbf{U} $	Velocity-magnitude field derived from the three predicted velocity components
p	Pressure in the governing-equation notation
P	Discrete pressure field used in the surrogate and result notation
$\mathbf{q}$	Aerodynamic state containing the three velocity components and pressure
$\hat{\mathbf{q}}$	Predicted volumetric aerodynamic state
ρ	Fluid density
ν	Kinematic viscosity
ν_t	Turbulent kinematic viscosity
$\mathbf{f}$	Body force per unit mass in the momentum equation
U_∞	Freestream velocity
p_∞	Freestream pressure
Re_L	Reynolds number based on the characteristic length L
Ma	Mach number
$\mathbf{S}$	Mean strain-rate tensor
k	Turbulent kinetic energy
$\boldsymbol{\tau}_w$	Wall shear-stress vector
$\boldsymbol{\eta}$	Vector of aerodynamic operating conditions in the general operator formulation

LIST OF ACRONYMS

Symbol	Meaning
<i>Aerodynamic coefficients</i>	
F_D	Aerodynamic drag force along the streamwise direction $\mathbf{e}_y$
C_D	Drag coefficient
$\hat{C}_D$	Predicted drag coefficient
C_L	Lift coefficient; defined in the physical preliminaries but not used as a surrogate target
C_p	Pressure coefficient
C_f	Skin-friction coefficient
A_{ref}	Reference frontal area
ΔC_D	Drag-coefficient difference; one drag count corresponds to $\Delta C_D = 10^{-3}$
<i>Geometry and discretization</i>	
φ	Signed distance function and, by context, its sampled signed distance field
$\hat{\varphi}$	Reconstructed signed distance field
$\hat{\varphi}_\mu$	SDF reconstructed from the design-parameter branch
$\hat{\varphi}_\varphi$	SDF reconstructed from the volumetric geometry branch
Γ	Body surface represented by the zero level set of φ
Ω_b	Region occupied by the solid body
Ω	Generic physical/computational domain; also used in the ROM/FFD discussion with the meaning specified locally
$\mathcal{D}$	Cartesian computational box used for the surrogate representation
Ω_h	Discrete Cartesian voxel grid; its resolution depends on the stage and dataset
$\mathcal{F}$	Fluid subset of the discrete computational domain used for field evaluation
φ	Rear slant angle of the Ahmed body
$\mathbf{n}$	Outward unit normal to the vehicle surface
κ	Surface/level-set curvature in the SDF preliminaries
$\boldsymbol{\mu}$	Vector of design parameters
$\mathcal{P}$	Design-parameter space
$\mathcal{M}$	Context-dependent notation: solution manifold in the ROM discussion and deformation map in the FFD subsection
$\mathcal{X}_\varphi$	Space of signed distance fields
$\mathcal{X}_q$	Space of volumetric aerodynamic states
<i>Networks, latent representation, and regression</i>	
d	Latent-space dimension; $d = 32$ for AhmedML and $d = 128$ for the realistic-geometry configuration
N	Number of values in a discretized field; resolution dependent
M	Number of training samples in the generic learning formulation

Symbol	Meaning
θ	Parameters of the geometric representation network
ψ	Context-dependent notation: map to the reference cube in the FFD discussion, and parameters of the Stage 2 volumetric aerodynamic network in the surrogate formulation
γ	Parameters of the generic scalar drag-regression map
E_θ^φ	Volumetric SDF encoder
N_θ^μ	Design-parameter branch, implemented as a multilayer perceptron
D_θ^φ	Shared SDF decoder
$\mathcal{G}_\theta^\mu$	Parameter-to-SDF geometric operator, $D_\theta^\varphi \circ N_\theta^\mu$
$\mathcal{G}_\theta^\varphi$	Geometry-input reconstruction operator, $D_\theta^\varphi \circ E_\theta^\varphi$
$z^{(p)}, z^{(s)}$	Latent codes produced by the parameter and volumetric SDF branches, respectively
$\mathcal{F}_\psi$	Stage 2 volumetric aerodynamic predictor, $\mathcal{X}_\varphi \rightarrow \mathcal{X}_q$
$\mathcal{H}_\psi$	Mapping from an input SDF to the pooled Stage 2 encoder descriptor used for drag regression
$\mathbf{g}$	Pooled multi-scale Stage 2 encoder feature vector used by the final drag regressor
$\mathcal{R}_\gamma$	Generic scalar regression map from pooled encoder features to $\hat{C}_D$
f_{ET}	ExtraTrees component of the final drag-regression ensemble
f_{R}	Ridge-regression component of the final drag-regression ensemble
$\mathcal{A}_{\psi, \gamma}$	Combined aerodynamic map returning volumetric fields and, on tiers where drag is evaluated, the separately regressed drag coefficient
ε_{act}	Smoothing width of the optional hyperbolic-tangent indicator used in the Stage 1 auxiliary loss
$\chi_{\varepsilon_{\text{act}}}$	Smoothed indicator transformation used by the optional Stage 1 activation loss
$\mathcal{Q}$	Set of Stage 2 predicted fields, $\{U_x, U_y, U_z, P\}$
$M_i(\mathbf{x})$	Fluid-domain mask for design i in the Stage 2 loss
$N_{f,i}$	Number of fluid voxels contributing to the Stage 2 loss for design i
B	Mini-batch size in the Stage 2 loss
$a_c^{(k)}, m_c^{(k)}, s_c^{(k)}$	Mean, maximum, and standard deviation pooled from channel c of Stage 2 feature map k
$\tilde{\mathbf{g}}$	Standardized pooled Stage 2 descriptor used by the scalar drag regressors
$\lambda_{\text{SDF}}, \lambda_{\text{latent}}, \lambda_{\text{act}}$	Weights of the Stage 1 SDF reconstruction, latent-alignment, and auxiliary activation-loss terms
$\mathcal{L}$	Loss function
ℓ	Per-sample loss

LIST OF ACRONYMS

Symbol	Meaning
<i>Evaluation metrics</i>	
RelL2	Relative L^2 error used as the principal volumetric field metric
MSE	Mean squared error
MAE	Mean absolute error
Dice	Dice coefficient used to quantify overlap of the reconstructed and reference solid regions
HD_{95}	Symmetric 95th-percentile Hausdorff distance between reconstructed and reference surfaces
R^2	Coefficient of determination, used as a complementary statistical indicator for drag regression

Introduction

1.1 Framework and State of the Art

This thesis presents the outcomes of a collaborative research project between SISSA mathLab and Siemens. The study integrates computational fluid dynamics (CFD), reduced-order modeling, and scientific machine learning with industrial vehicle data and engineering workflows [15, 119, 44]. This research primarily investigates data-driven surrogate models that reduce the computational cost of aerodynamic prediction while remaining applicable to increasingly realistic vehicle geometries. Ultimately, the collaboration seeks to integrate the developed surrogate methodology into the Siemens Scientific Machine Learning (SciML) Sandbox.

This work is motivated by practical needs. High-fidelity aerodynamic simulations give detailed data on velocity, pressure, wall shear stress, and drag. These simulations are key for understanding vehicle aerodynamics and testing design changes. However, they become costly when many different shapes need to be tested. Depending on the setup, one vehicle simulation can use tens of millions of computational cells and take many hours or even days on a computing cluster [6, 7, 107]. Running this process for many designs creates a heavy computational load. This cost limits how much of the aerodynamic design space can be explored. In the early stages of vehicle development, there are often many design options, but only a few can be tested with high-fidelity CFD due to limited resources [52, 75]. As the number of design variables grows, the possible configurations increase quickly, making it impossible to test them all. As a result, some promising areas of the design space may not be explored because high-fidelity CFD is too expensive for every option.

Data-driven surrogate modeling offers an alternative approach to address this limitation. After being trained on completed computational fluid dynamics (CFD) simulations, a surrogate model can estimate aerodynamic quantities

for new geometries at substantially lower computational cost than conducting new high-fidelity simulations [15, 119, 53]. This approach enables the screening of a larger set of candidate designs, facilitates the identification of promising regions within the design space, and allows expensive CFD calculations to be reserved for configurations that require more detailed analysis. The objective is not to replace CFD as the reference aerodynamic tool, but rather to introduce a computationally efficient intermediary between geometric design and high-fidelity simulation. In this thesis, CFD solutions serve as reference data for training and evaluation. These solutions are not considered exact, as they retain the modeling, discretization, and convergence characteristics inherent to the numerical solver.

A significant challenge emerges in the representation of geometry. When a design family is predefined and described by a fixed set of variables, parameter-based approaches offer an effective framework for capturing geometric variations[57, 102].

Projection- and deformation-based techniques, such as Free-Form Deformation, can be employed to modify parametrized domains [104, 57, 102, 32]. These methods have also been successfully integrated with reduced-order modeling approaches [98, 90, 43, 10, 44].

However, their applicability becomes more challenging when geometries cannot be described using a common set of design parameters [49, 104].

This limitation is particularly evident when substantial geometric modifications are introduced, such as the addition or removal of spoiler wings or other aerodynamic components [49]. Chapter 2 provides a more detailed discussion of these approaches.

This consideration motivates the adoption of geometry-based representations. Instead of representing a shape solely through a fixed vector of design parameters, geometry-based methods supply the spatial geometry directly to the learning model. Convolutional grids, graphs, point clouds, and neural operators have each been explored for this purpose [39, 117, 86, 88, 89, 54, 62, 24].

Although these approaches vary in sampling, discretization, and geometry processing, they share the goal of learning aerodynamic quantities directly from spatial representations of shape. This characteristic is especially advantageous for design-space exploration, as the resulting model is not inherently limited to a single explicit set of design parameters.

This thesis adopts the signed distance field (SDF) as the standard geometric representation [35, 83, 37]. The vehicle surface is implicitly defined by the zero level set of the SDF, and the field is sampled on a regular three-dimensional Cartesian grid for direct input into the learned models. Geometries produced by various design, CAD, or meshing workflows can thus be converted into

a unified volumetric representation prior to integration into the surrogate framework [77, 83]. This standardized representation is especially valuable when transitioning between different vehicle families that lack a shared design-parameter space.

The proposed formulation decouples geometry reconstruction from aerodynamic prediction. The initial learned stage reconstructs geometry using the standard signed distance function (SDF) representation. Subsequently, a second stage maps this representation to the corresponding aerodynamic fields. This approach establishes a clear design-to-geometry-to-flow pathway, enabling independent evaluation of the geometric representation prior to assessing its impact on aerodynamic prediction. In addition to reducing the computational cost of individual predictions, this formulation may facilitate broader exploration of aerodynamic design spaces by offering a unified interface between diverse geometries and learned flow models.

The Siemens Scientific Machine Learning (SciML) Sandbox establishes the broader industrial context for this research. This internal environment supports the development and testing of scientific machine-learning methods for engineering simulations, enabling the integration of data from tools such as Simcenter STAR-CCM+ with reduced-order models, surrogate models, neural operators, and other data-driven components. The long-term objective of the collaboration is to adapt the surrogate methodology developed in this thesis for integration into the SciML Sandbox. However, full software integration of the proposed framework into the SciML Sandbox is not presented as a completed outcome of this thesis. Implementation details regarding SDF construction, computational domain standardization, grid resolution, orientation, sign convention, and surface preprocessing are discussed in Chapter 3.

Proposed Framework. This thesis introduces a two-stage surrogate in which the SDF forms the explicit link between geometric representation and aerodynamic prediction. The main methodological idea is to evaluate geometry reconstruction and aerodynamic prediction separately before evaluating their composition.

The surrogate contains two learned stages:

1. **Stage 1 (Geometric):** a compact latent geometric representation is learned using a parameter branch and a volumetric SDF branch that share a decoder for three-dimensional SDF reconstruction.
2. **Stage 2 (Aerodynamic):** a three-dimensional U-Net maps the SDF to the three velocity components and pressure. Drag is handled separately by a scalar regressor using pooled Stage 2 encoder features on datasets where drag labels are available.

The Stage 1 framework utilizes two entry paths based on the available geometric information. When a consistent low-dimensional parameterization exists, as demonstrated in AhmedML and DrivAerML, the parameter-driven branch reconstructs the SDF directly from the relevant design variables. In cases where a shared parameterization is absent, the volumetric SDF branch serves as an alternative geometric input path. For example, in the mixed DrivAerML–Siemens transfer, the Siemens dataset comprises only 20 designs and lacks a sufficiently rich basis for establishing and learning a dedicated low-dimensional parameterization. Although DrivAerML possesses its own morphing parameters and can be evaluated through the parameter branch, these parameters are unique to DrivAerML and cannot be directly applied to Siemens geometries. Consequently, the dual-branch structure enhances the framework’s flexibility: the parameter-based path is employed when appropriate design variables are available, whereas the volumetric SDF path accommodates geometries without a common parameter description within the Stage 1 framework.

For aerodynamic evaluation, the direct Stage 2 model uses the reference SDF, whereas the end-to-end Stage 2 model is trained separately using SDFs reconstructed by the retained Stage 1 model.

The framework is evaluated using shapes with progressively increasing geometric complexity, including analytical geometries, the parametric AhmedML dataset [6], the full-vehicle DrivAerML dataset [7], and industrial Siemens vehicle configurations. This study examines supervised prediction of steady or time-averaged aerodynamic fields on fixed Cartesian grids. Consequently, the conclusions are limited to the geometry families, physical conditions, and discretizations represented in the available data.

1.2 Research Questions and Contributions

This thesis explores whether a compact learned representation of vehicle geometry using signed distance fields can support fast three-dimensional aerodynamic prediction while retaining an independently testable geometric stage. Three research questions are considered.

Research Question 1. *Can a neural network learn a compact latent representation that accurately reconstructs vehicle signed distance fields? How does reconstruction accuracy change as the geometry becomes more complex, moving from simple shapes to full production vehicles?*

This question is studied using both volumetric and surface-sensitive geometric metrics.

Research Question 2. *Is it possible for a three-dimensional convolutional model to predict volumetric velocity and pressure fields using only the signed distance field and also provide useful drag estimates?*

This question is evaluated by testing Stage 2 directly from reference SDFs and measuring the resulting aerodynamic field and drag accuracy.

Research Question 3. *Is it possible to combine the geometric and aerodynamic stages into a single surrogate that maintains useful predictive accuracy while reducing the computational cost of aerodynamic evaluation?*

This question is addressed through the complete Stage 1-Stage 2 composition, with the detailed training partitions, field errors, and per-design distributions reported in Chapter 6.

Main Contributions.

- This approach uses a two-stage SDF-based surrogate, where geometry reconstruction is validated on its own. The framework keeps geometry reconstruction and aerodynamic prediction separate by using a signed-distance-field interface. The geometric stage is trained and tested independently with both volumetric and surface-sensitive metrics. Once the geometry is reconstructed, it is passed to the aerodynamic model. This setup helps identify whether errors come from the geometric representation or from the flow-field prediction.
- A shared geometric representation can be used for different types of vehicles. The signed-distance method offers a common interface for AhmedML, DrivAerML, and Siemens geometries, even though each uses a different geometric description. For example, both DrivAerML and Siemens models can be handled using the same volumetric geometry approach. This means there is no need for a shared design-parameter space, so the staged methodology can be applied to a wider range of realistic vehicle families.
- This end-to-end design-to-flow framework has been tested on both benchmark and industrial geometries. It combines two learned stages and evaluates them together, predicting aerodynamic fields directly from the design description using the reconstructed geometry instead of a reference surface. The framework was applied to AhmedML, DrivAerML, and Siemens production geometries, which helped measure how geometric reconstruction affects aerodynamic prediction as the method moves from benchmark to real-world and industrial cases.

Part of this work has been submitted for publication:

- A. Dahdah, F. Pichi, N. Tonicello, G. A. D’Inverno, and G. Rozza, “Geometry-Centric Aerodynamic Surrogate Modeling via Signed Distance Field Reconstruction,” submitted for publication, 2026.

1.3 Structure of the Thesis

The rest of this thesis is structured as follows.

Chapter 2 covers the physical and mathematical background. It introduces the main equations, non-dimensional quantities, projection-based reduced-order modeling, Free-Form Deformation, autoencoders, and signed distance functions.

Chapter 3 specifies the experimental procedure and implementation. It covers dataset preparation, task-specific partitions, geometry and field pre-processing, evaluation metrics, network architectures, loss functions, training protocols, and transfer between datasets.

Chapter 4 evaluates Stage 1, focusing on SDF reconstruction, zero-level-set and surface accuracy, and the effect of increasing geometric complexity.

Chapter 5 evaluates Stage 2 independently using reference SDFs. It reports three-dimensional flow prediction, per-design error distributions, qualitative field comparisons, and drag regression where scalar targets are available.

Chapter 6 evaluates the complete surrogate using the geometry reconstructed by Stage 1 as input to the end-to-end Stage 2 model. It examines the accuracy of the composed prediction path and its inference cost relative to the CFD reference calculations.

Chapter 7 summarizes the results in relation to the three research questions, discusses the main limitations of the framework, and identifies directions for future development.

Chapter 2

Physical and Mathematical Preliminaries

This chapter sets out the physical and mathematical background needed for the methods developed later in the thesis. It introduces the governing equations and aerodynamic quantities, reviews projection-based reduced-order modeling and geometric deformation, presents convolutional field approximation and recent geometry-aware learning approaches, and gives the mathematical definitions associated with signed distance functions. The experimental implementation is described in Chapter 3.

2.1 Aerodynamics of Road Vehicles

A single coordinate system is employed throughout this thesis. The streamwise direction is denoted as y , the lateral or spanwise direction as x , and the vertical direction as z . Accordingly, U_y represents the streamwise velocity, while U_x and U_z correspond to the lateral and vertical velocity components, respectively.

Road-vehicle flows are characterized by high Reynolds numbers, turbulent boundary layers, flow separation, and complex three-dimensional wakes [46, 87]. For the datasets considered in this study, the operating conditions remain within the low-Mach-number regime, for which compressibility effects can be neglected and an incompressible-flow formulation is appropriate [6, 7]. The governing equations are presented as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}, \quad \nabla \cdot \mathbf{u} = 0. \quad (2.1)$$

Here, $\mathbf{u}(t, \mathbf{x}) : [0, T] \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the velocity, p is the static pressure, ρ is the fluid density, and ν is the kinematic viscosity. Moving surfaces like

wheels and road planes are included through boundary conditions instead of a volumetric forcing term.

Reference CFD solutions are obtained by discretizing the fluid domain into finite-volume meshes based on underlying computer-aided design (CAD) or surface geometries [118, 25]. The meshing process is determined by the chosen numerical model, specific operating conditions, and the required level of surface-conforming accuracy [92]. Since the surrogate model does not utilize these solver-native meshes directly, the solution fields are interpolated onto a standardized Cartesian grid for training purposes, as described in Chapter 3.

Dimensionless Numbers and Field Normalization. The Reynolds and Mach numbers provide the principal non-dimensional measures of the operating regime,

$$\text{Re}_L = \frac{U_\infty L}{\nu}, \quad \text{Ma} = \frac{U_\infty}{a_\infty}. \quad (2.2)$$

Here, U_∞ is the freestream velocity, L is a characteristic vehicle length, and a_∞ is the speed of sound. For the road-vehicle conditions discussed, Re_L is usually between 10^6 and 10^7 [6, 7]. Inertia is much more significant than molecular viscous effects in most of the flow [1, 46, 50]. The low Mach number also means the incompressible formulation above is valid [7, 50]. Velocity and pressure can be expressed relative to the freestream as follows:

$$\mathbf{u}^* = \frac{\mathbf{u}}{U_\infty}, \quad p^* = \frac{p - p_\infty}{\frac{1}{2}\rho U_\infty^2}. \quad (2.3)$$

Here, p_∞ stands for the freestream pressure. This scaling sets the standard non-dimensional convention when the dataset is stored that way. Not all source datasets follow the same approach: some use non-dimensional fields, while others keep dimensional solver outputs. Chapter 3 explains which convention is used for each campaign.

Neural-network standardization is a separate operation from physical non-dimensionalization. The training-time transformation and its inversion before evaluation are defined in Chapter 3; reversing this transformation restores the original dataset convention, not necessarily SI units.

At Reynolds numbers relevant to automotive applications, turbulent flow exhibits a wide range of spatial and temporal scales. Energy transfers from large-scale vortices to progressively smaller structures until viscous dissipation becomes dominant at the Kolmogorov scale [87]. Resolving this entire spectrum using Direct Numerical Simulation (DNS) is computationally infeasible for the vehicle configurations examined in this study [17]. Therefore, practical engineering workflows typically employ turbulence modeling or hybrid scale-resolving approaches [71, 112, 38, 56].

Turbulence modeling. In a Reynolds-averaged description, velocity and pressure are decomposed into mean and fluctuating parts [87, 71],

$$\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}', \quad p = \bar{p} + p', \quad (2.4)$$

which yields the Reynolds-Averaged Navier–Stokes equations

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + (\bar{\mathbf{u}} \cdot \nabla) \bar{\mathbf{u}} = -\frac{1}{\rho} \nabla \bar{p} + \nu \nabla^2 \bar{\mathbf{u}} - \nabla \cdot \overline{\mathbf{u}' \otimes \mathbf{u}'}, \quad \nabla \cdot \bar{\mathbf{u}} = 0. \quad (2.5)$$

After averaging the incompressible Navier–Stokes system, the extra tensor $\overline{\mathbf{u}' \otimes \mathbf{u}'}$ shows the effect of unresolved turbulent fluctuations and leads to the closure problem [87, 23].

According to the Boussinesq hypothesis, the anisotropic part of the Reynolds stress is described using a modeled eddy viscosity [87, 71],

$$-\overline{\mathbf{u}' \otimes \mathbf{u}'} = 2\nu_t \mathbf{S} - \frac{2}{3}k \mathbf{I}, \quad \mathbf{S} = \frac{1}{2} (\nabla \bar{\mathbf{u}} + \nabla \bar{\mathbf{u}}^\top), \quad (2.6)$$

where $\mathbf{S}$ is the mean strain-rate tensor, ν_t the turbulent kinematic viscosity, k the turbulent kinetic energy, and $\mathbf{I}$ the identity tensor. The equations required to determine ν_t depend on the chosen closure model.

Steady RANS is still widely used in industrial aerodynamics because it gives mean-flow estimates at a reasonable cost for repeated engineering analysis [71, 23]. However, its weaknesses show up in separated, unsteady wakes, where steady models struggle to capture large coherent structures [38]. Large-Eddy Simulation can resolve the main energy-carrying scales and models the smaller ones, but it becomes very expensive near walls at vehicle Reynolds numbers [56, 72, 109, 33, 17]. Hybrid RANS–LES methods, such as Detached-Eddy Simulation, use RANS for attached boundary layers and switch to scale-resolving models in separated regions [112, 110, 106, 28, 38].

The two open-source datasets used later in this thesis were created using hybrid scale-resolving CFD and then time-averaged [6, 7]. The Siemens industrial cases, by contrast, use steady RANS solutions on meshes containing several tens of millions of cells. These CFD fields are treated as numerical reference data for learning and evaluation and retain the modeling, discretization, and convergence characteristics of the simulations from which they were obtained.

Computational domain and boundary conditions. Each vehicle sits within a limited fluid domain. A set freestream enters from upstream, and the downstream boundary lets the wake exit the domain. The vehicle surface uses no-slip conditions. Road and wheel setups vary between reference campaigns: the ground can be stationary or moving, and wheels can be either stationary or rotating. Far-field boundaries are placed far enough from the body to reduce confinement effects [46, 42].

Aerodynamic forces and drag. The total aerodynamic force follows from the integral of pressure and viscous tractions over the body surface Γ [46],

$$\mathbf{F} = \oint_{\Gamma} (-(p - p_{\infty}) \mathbf{n} + \boldsymbol{\tau}_w) dS. \quad (2.7)$$

Here, $\mathbf{n}$ represents the outward unit normal, and $\boldsymbol{\tau}_w$ is the wall-shear-stress vector. Using gauge pressure matches the normalization in Equation (2.3). For a closed surface, uniform freestream pressure does not contribute overall since $\oint_{\Gamma} \mathbf{n} dS = \mathbf{0}$.

The drag force is the streamwise component,

$$F_D = \mathbf{F} \cdot \mathbf{e}_y, \quad (2.8)$$

and its non-dimensional coefficient is

$$C_D = \frac{F_D}{\frac{1}{2} \rho U_{\infty}^2 A_{\text{ref}}}, \quad (2.9)$$

where A_{ref} denotes the selected frontal reference area. Similarly, the lift coefficient C_L is defined along $\mathbf{e}_z$. Consistent use of the reference-area convention is essential when comparing different geometries. Although C_L contributes to the aerodynamic characterization, it is not employed as a surrogate target in this thesis.

Equation (2.9) defines the physical drag quantity used as the regression target later in the thesis. The computational procedure used to predict C_D is described in Chapter 3.

For road vehicles, pressure drag typically dominates due to bluff body separation, which generates a broad low-pressure wake [46, 38, 36, 7]. Local pressure loading is quantified by

$$C_p = \frac{p - p_{\infty}}{\frac{1}{2} \rho U_{\infty}^2}, \quad (2.10)$$

while the local viscous contribution may be represented by

$$C_f = \frac{\|\boldsymbol{\tau}_w\|}{\frac{1}{2} \rho U_{\infty}^2}. \quad (2.11)$$

These surface quantities help explain the global drag coefficient by showing where aerodynamic loading happens.

Differences in C_D are also expressed in drag counts as follows:

$$1 \text{ drag count} \equiv \Delta C_D = 10^{-3}. \quad (2.12)$$

2.1.1 Characteristic Flow Structures in the Benchmark Geometries

Ahmed [6, 1] and DrivAer [7, 24, 42] offer different but complementary aerodynamic test cases. The Ahmed body focuses on a few geometric features and shows clear changes in wake patterns as the rear shape changes. DrivAer adds the complex details and flow interactions found in real passenger cars. Moving from Ahmed to DrivAer helps show if the same geometric approach still works as shapes get more complex and aerodynamic effects become stronger.

Flow around the Ahmed body

The Ahmed body is a standard bluff-body shape with a rounded front, parallel sides, a flat bottom, a sloped rear, and a vertical back [1]. Even with its simple design, it shows key features of road-vehicle aerodynamics, such as boundary-layer growth, pressure-driven separation, shear-layer reattachment, streamwise vortices, and a large turbulent wake [1, 63, 55, 36].

As air approaches the front of the Ahmed body, it slows down and forms a stagnation region, then speeds up around the rounded edges. Boundary layers form along the roof, sides, and bottom. The angle of the rear slant, φ , is important because it affects how the flow separates at the back and interacts with the wake [1, 55].

When the slant angle is small, the airflow mostly stays attached to the sloped surface and only separates at the vertical back [1]. This creates a large recirculation area, a loss of streamwise momentum, and low pressure at the base [1, 63]. At medium angles, the flow separates near the top of the slant, forming a shear layer above the rear. The pressure difference between the top and sides also creates a pair of counter-rotating streamwise vortices [1, 63, 55].

In the classic 25° case, these vortices can move higher-momentum fluid toward the slant and promote reattachment roughly partway along the rear surface before a second separation at the base. The near wake is then determined by the interaction of the slant shear layer, the vortex pair, and the recirculating base flow [63, 55, 6]. At 35° , the longitudinal vortices are weaker, and the flow remains separated over most or all of the slant, producing a different pressure distribution and wake state [1, 93, 6].

These flow patterns do not change in a simple way as the rear angle changes. Even small changes in shape can shift the flow between attached, reattached, and fully separated states [1, 38, 55]. The drag depends on how the vortices and base wake interact, and the predicted separation points are sensitive to how the simulation is set up [56, 38, 93, 41, 6]. To distinguish different flow states, a learned model must accurately capture the geometry near the slant

and base. Otherwise, small errors in shape can cause much bigger differences further downstream.

AhmedML contains 500 different shapes created by changing the main body sizes and the rear design. Each case was simulated using a time-accurate hybrid RANS–LES method and provides time-averaged 3D fields, surface data, and force coefficients [6]. This dataset gives a controlled set of examples where changes in shape can be connected to changes in flow separation and wake structure.

Flow around the DrivAer model

The DrivAer model was developed as a generic passenger-vehicle shape to bridge the gap between simple bluff bodies and actual production cars [42]. Its design incorporates realistic features such as greenhouse, windscreen, and roof curves, A-pillars, mirrors, wheels with arches, underbody details, and multiple rear-end configurations. The DrivAerML subset utilized in this study comprises 500 notchback variants, each generated using a consistent, high-fidelity computational fluid dynamics (CFD) process [7].

Airflow slows down at the front of the vehicle, then speeds up over the hood and around the front corners. The hood, windscreen, and roof shape the boundary layer on top and affect how pressure recovers further back. Vortices from the A-pillars and wakes from the mirrors add three-dimensional flow features that move along the car’s sides [42, 7].

The areas around the wheels and wheel arches cause more flow separation and swirling motion, which interact with the side flow, underbody, and the wake behind the rear wheels. Under the car, the narrow gap between the underbody and the road speeds up the airflow and makes it sensitive to ride height, underbody shape, and how the wheels meet the ground. These boundary conditions depend on the dataset and are explained in Chapter 3 [42, 7].

At the back of the notchback body, the wake is shaped by the roof, C-pillars, rear window, trunk deck, side flow, underbody, and wheel wakes. Flow separation can happen at several of these spots, so the final wake is a complex three-dimensional mix of shear layers, streamwise vortices, wheel wakes, and a large low-speed recirculation area [42, 7].

This means that many parts of the car’s shape affect its aerodynamics, rather than just one feature. Adjusting the hood, windscreen angle, roof curve, ride height, wheel position, rear window, or trunk shape can change both local airflow and how it interacts with the wake behind the car. This has several implications for surrogate modeling:

- **Multiscale geometry.** When modeling a vehicle, it is important to show both the overall shape and smaller details like mirrors, wheel openings,

sharp edges, and underbody parts [42, 101, 8]. On a fixed Cartesian grid, some of these features occupy only a few voxels.

- **Long-range aerodynamic interaction.** Changing the shape of a surface in one area can affect a shear layer or vortex, and these changes often show up clearly several body lengths away. Because of this, a good model should provide detailed information about small areas while also covering a wide region [30, 68].
- **Coupled wake structures.** A-pillar vortices, mirror wakes, wheel wakes, underbody flow, and rear separation all interact in complex ways. Because of this, it is usually not possible to link the overall effect to just one feature [2, 122, 121].
- **Localized sensitivity.** Most of the computational domain is nearly uniform freestream, but the strongest gradients and design changes happen near the vehicle and in its wake. As a result, global training losses may not focus enough on the areas most important for aerodynamic behavior.
- **Limited sampling of a high-dimensional design family.** Although there are several hundred CFD cases, they only represent a limited set of the possible combinations found in real vehicle designs. As a result, performance should be understood within the design ranges shown in the data.
- **Dependence on operating conditions.** The geometric datasets set the inflow, Reynolds number, road treatment, and wheel conditions. To make predictions for other situations, such as different yaw angles or speeds, the model training would need to include those changes.

Complementary role of the two benchmarks

The two benchmarks are complementary. Ahmed isolates strong changes in separation and wake behavior within a comparatively compact geometric family, while DrivAer introduces the coupled surface details and multiscale flow interactions of a full vehicle. Together, they provide the physical progression from a controlled bluff-body benchmark to a realistic passenger-car geometry.

2.2 Reduced-Order Models and Surrogates

Due to the significant computational expense of high-fidelity simulations, surrogate models are utilized to approximate the input-output behavior of the full-order problem at a substantially reduced cost [10, 43, 100]. In the classical parametrized context, the objective is to determine a solution $u(\mu)$

for $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^P$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in V, \quad (2.13)$$

where V denotes the space of admissible test functions. The forms a and f include the governing equations, boundary conditions, and any parameter dependence. The parameter vector may represent physical quantities, operating conditions, or a finite-dimensional geometric description.

Projection-based reduced-order models take advantage of the fact that the solution manifold

$$\mathcal{M} = \{u(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\} \quad (2.14)$$

can be approximated by a low-dimensional linear space. Given M snapshots $u(\boldsymbol{\mu}_m)$, POD selects an orthonormal basis $\{\zeta_n\}_{n=1}^r$ that minimizes the mean projection error [108, 12],

$$\min_{\substack{\{\zeta_n\}_{n=1}^r \\ (\zeta_i, \zeta_j) = \delta_{ij}}} \frac{1}{M} \sum_{m=1}^M \left\| u(\boldsymbol{\mu}_m) - \sum_{n=1}^r (u(\boldsymbol{\mu}_m), \zeta_n) \zeta_n \right\|^2. \quad (2.15)$$

The basis is usually found by performing a singular-value decomposition on the snapshot matrix or an eigendecomposition on its correlation matrix. The rate at which the singular values or eigenvalues decrease shows how well the sampled solution family can be represented [108, 12, 113, 15].

The reduced approximation is

$$u_r(\boldsymbol{\mu}) = \sum_{n=1}^r \theta_n(\boldsymbol{\mu}) \zeta_n. \quad (2.16)$$

The coefficients $\theta_n(\boldsymbol{\mu})$ are determined within the reduced space. Reduced-basis methods extend this approach by employing parameter-selection strategies and a posteriori error estimation [98, 43, 90, 10]. Projection-based reduction has also worked well for parametrized incompressible flow problems [113, 44].

Geometric variability in projection-based reduction. If the physical domain changes, snapshots still need to be put into a shared function space so that the inner products used in projection make sense. One common approach is to use a fixed reference domain along with a deformation map [115, 5],

$$T(\cdot; \boldsymbol{\mu}) : \hat{\Omega} \longrightarrow \Omega(\boldsymbol{\mu}), \quad (2.17)$$

that links the reference configuration to the geometry associated with each parameter vector.

Free-Form Deformation. FFD constructs such a geometric map by embedding the body in a regular control lattice and deforming the surrounding space through smooth tensor-product interpolation [104, 102]. The transformation is written as

$$\mathcal{M}(\cdot; \mu) = \left(\Pi^{-1} \circ \hat{T} \circ \Pi \right) (\cdot; \mu), \quad (2.18)$$

where Π maps to the reference cube, $\hat{T}$ applies the Bernstein-polynomial deformation, and Π^{-1} returns the result to the physical domain [104, 32, 57]. The deformed domain is

$$\Omega(\mu) = \mathcal{M}(\Omega; \mu). \quad (2.19)$$

The design variables are the displacements of the control points. With this approach, a small set of parameters can create smooth changes to the overall geometry. This method also works well with reduced-order and shape-optimization techniques [99, 57, 102, 32].

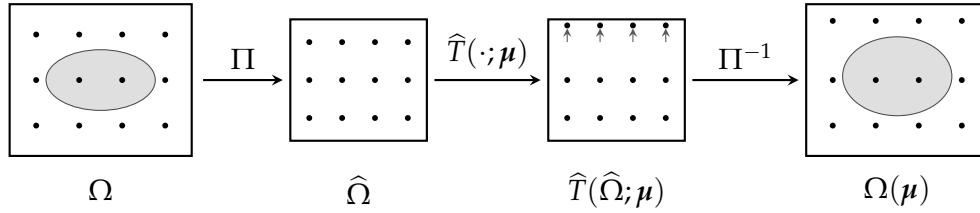


Figure 2.1: Structure of the Free-Form Deformation map $\mathcal{M}(\cdot; \mu)$ as the composition of the map Π onto the reference configuration, the Bernstein-polynomial deformation $\hat{T}(\cdot; \mu)$ driven by the displacement of the control points, and the back-map Π^{-1} to the physical domain. Only the control points of the top lattice row are displaced here, so that the effect of a single group of design parameters on the embedded geometry remains visible.

FFD works well because it uses a controlled, admissible family. If a geometry is created separately from the reference shape and lattice, it does not automatically fit into that parameter space [81]. This situation may require extra steps like registration, inverse parameterization, or creating a new deformation map [115]. The control variables are lattice displacements, not always physical design dimensions, and a regular lattice can limit how local the allowed changes are [32]. These limits are fine within a set design family, but they can be restrictive when geometries come from different CAD or remeshing processes [80].

Linear and nonlinear approximation. The Kolmogorov r -width measures how well a compact solution manifold can be approximated by an r -dimensional linear space [78, 10]. Transport-dominated flows are challenging because wakes, shear layers, and recirculation regions move and change shape as

parameters vary. As a result, a fixed global linear basis might need many modes [94, 115].

Nonlinear approximators do not require a linear-manifold assumption and can sometimes represent these families more efficiently, depending on the architecture, data, and training methods. Non-intrusive reduced models learn the input-output relationship from snapshot data instead of projecting the governing equations [43, 85]. Convolutional models have shown that, once training is finished, flow fields around different geometries can be evaluated much more efficiently than with full CFD simulations [39, 59, 96].

2.3 Convolutional Autoencoders for Field Prediction

Convolutional encoder–decoder networks provide nonlinear approximations for high-dimensional spatial fields. They are therefore useful both for dimensionality reduction and for mappings between different volumetric quantities [59, 76, 96].

Given the training dataset

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^M, \quad (2.20)$$

model training minimizes the average training loss

$$\min_{\theta} \frac{1}{M} \sum_{i=1}^M \ell(G_{\theta}(x_i), y_i), \quad (2.21)$$

where G_{θ} denotes the network and θ its trainable parameters. We use a task-dependent loss ℓ . To measure predictive accuracy, we evaluate it separately on held-out designs. The input may be a design vector or a volumetric geometric field, while the output could be an SDF, an aerodynamic field, or a scalar coefficient.

A deep network is a composition of layers,

$$G_{\theta} = g_L \circ \cdots \circ g_1. \quad (2.22)$$

Universal approximation theory shows that these models have wide expressive capacity [21]. However, it does not provide a practical way to design large 3D arrays. Using convolutions helps lower the number of parameters and takes advantage of spatial locality [58, 19, 39].

Three-dimensional convolutional layers. A 3D convolution is written as

$$(\mathcal{C}_w v)_c(\mathbf{x}) = b_c + \sum_{c'=1}^{C_{\text{in}}} \sum_{\delta \in \mathcal{K}} w_{c,c'}(\delta) v_{c'}(\mathbf{x} + \delta). \quad (2.23)$$

Here, v is the input field with C_{in} channels, $w_{c,c'}$ are the kernel weights, $\mathcal{K}$ stands for the kernel support and b_c is a bias term. With weight sharing, the number of parameters depends on the kernel and channel dimensions instead of the number of voxels. This approach also brings in translation equivariance, except near the boundaries. As the network gets deeper and uses more downsampling, the receptive field becomes larger [126, 19]. This lets the network capture both fine geometric details and the broader spatial context needed to represent a wake [39].

Autoencoders and nonlinear dimensionality reduction. An autoencoder combines

$$E_\theta : \mathbb{R}^N \longrightarrow \mathbb{R}^d, \quad (2.24)$$

with

$$D_\theta : \mathbb{R}^d \longrightarrow \mathbb{R}^N, \quad (2.25)$$

and is trained through a reconstruction objective such as

$$\min_{\theta} \frac{1}{M} \sum_{i=1}^M \|D_\theta(E_\theta(x_i)) - x_i\|^2. \quad (2.26)$$

When the data is mean-centered and the reconstruction error is measured by the squared difference, a linear autoencoder covers the same principal subspace as the main POD modes [9]. If the encoder and decoder are nonlinear, the decoded states instead lie on

$$\mathcal{M}_\theta = \left\{ D_\theta(\mathbf{z}) : \mathbf{z} \in \mathbb{R}^d \right\}, \quad (2.27)$$

which can be interpreted as a learned nonlinear manifold. Such representations have proved useful in model-reduction settings involving transport-dominated dynamics [59, 27, 96].

For a grid of dimensions $N_x \times N_y \times N_z$,

$$N = N_x N_y N_z. \quad (2.28)$$

For three-dimensional fields, N is typically very large compared with the latent dimension d . This dimensional gap motivates the compact nonlinear representation used later in the thesis; the dataset-specific grid sizes and latent dimensions are given in Chapter 3.

Encoder–decoder maps between different fields. When the input and output represent different physical variables on the same grid, the architecture works as a field-to-field mapping instead of an autoencoder. The U-Net architecture is often used for this, and it includes skip connections between matching encoder and decoder levels [97, 19]. These skip connections help

coarse, low-resolution features capture the overall spatial context, while also keeping local, high-resolution details [97, 19, 48]. This approach is especially useful for aerodynamic fields, where areas near the body need fine detail, and the wake behind the object covers larger spatial regions [13, 116, 31].

2.4 Geometry-Aware Learning Beyond Fixed Discretizations

A significant advancement in scientific machine learning is the shift from models constrained by fixed discretizations to architectures that operate on more general spatial representations. Convolutional models are highly effective on Cartesian grids due to the efficient evaluation of local operators on regularly sampled data [39, 13, 116, 31]. However, this regularity inherently links the representation to a specific domain and resolution. For problems characterized by complex or varying geometries, alternative representations such as point sets, meshes, coordinates, or continuous operators offer distinct approaches to reducing reliance on a fixed grid [54, 60, 64]. Point-cloud methods illustrate this transition. PointNet [88] introduced a neural architecture capable of processing unordered point sets directly, thereby removing the requirement for conversion into regular voxel grids. Permutation-invariant operations facilitate the extraction of geometric information independent of the order of sampled points. Although PointNet was originally developed for geometric classification and segmentation [88], its core principle extends to scientific learning: geometries can be analyzed using sampled spatial coordinates without necessitating regular volumetric discretization [89, 123].

A related advancement is operator learning, which aims to approximate mappings between functions instead of fixed-dimensional vectors [54]. DeepONet [67] employs a branch-trunk architecture, where one network encodes the input function and another network represents the coordinates for output evaluation. In contrast, Fourier neural operators [61] learn mappings between function spaces using integral operators parameterized in Fourier space. These approaches expand the applicability of data-driven partial differential equation (PDE) surrogates, although addressing varying and irregular geometries remains a significant challenge [62, 64].

Recent methodologies address geometric dependence with greater specificity. The Geometry-Informed Neural Operator (GINO) [62] integrates graph and Fourier neural operators and utilizes a signed distance representation of geometry. Irregular physical domains are transformed to and from a regular latent grid, enabling the application of Fourier operators while preserving geometric information. Alternatively, some methods eliminate the need for regular latent discretization. For example, CORAL [105] uses coordinate-based neural fields for operator learning on general geometries

and is designed to accommodate varying spatial samplings and resolutions [14, 40].

Transformer-based approaches provide an alternative methodology. Transolver [124] introduces a physics-attention mechanism that groups discretization points into learned physical states, which enables the modeling of correlations on complex geometries without requiring independent attention for each mesh point. Recent query-based architectures have also been developed for large-scale aerodynamic prediction. UPT [4] separates the encoding of geometric surfaces from the coordinates at which aerodynamic quantities are evaluated. This separation allows a single encoded geometry to support predictions at arbitrary spatial query points; [3] further advances the query-based approach for high-fidelity automotive aerodynamics.

This study retains a Cartesian volumetric representation and explicitly defines geometry using a signed distance field. This approach maintains the regular structure necessary for three-dimensional convolutional networks and provides a unified geometric quantity applicable across various vehicle families. The resolution limitation inherent to a fixed Cartesian grid remains present and is explicitly addressed in the methodology and results sections. The subsequent section introduces the signed distance field and the geometric properties used in this thesis.

2.5 The Signed Distance Function as Geometric Interface

Let $\Omega_b \subset \mathbb{R}^3$ be the solid body and

$$\Gamma = \partial\Omega_b \quad (2.29)$$

its boundary. The signed distance function [22, 79, 35] is defined as

$$\varphi(\mathbf{x}) = \begin{cases} d(\mathbf{x}, \Gamma), & \mathbf{x} \notin \Omega_b, \\ -d(\mathbf{x}, \Gamma), & \mathbf{x} \in \Omega_b, \end{cases} \quad d(\mathbf{x}, \Gamma) = \min_{\mathbf{y} \in \Gamma} \|\mathbf{x} - \mathbf{y}\|, \quad (2.30)$$

with the surface recovered from

$$\Gamma = \{\mathbf{x} : \varphi(\mathbf{x}) = 0\}. \quad (2.31)$$

Positive values represent the fluid, while negative values indicate the inside of the body. When needed, we distinguish between the continuous signed distance function and its Cartesian sampling. The networks use the sampled signed distance field.

When there is only one closest surface point,

$$\|\nabla\varphi\| = 1, \quad (2.32)$$

and near a smooth boundary the gradient provides the normal,

$$\mathbf{n} = \nabla \varphi \quad \text{on } \Gamma. \quad (2.33)$$

Such properties connect the representation to level-set geometry [22, 79, 35].

Analytical properties. The field is globally 1-Lipschitz,

$$|\varphi(\mathbf{x}) - \varphi(\mathbf{y})| \leq \|\mathbf{x} - \mathbf{y}\|. \quad (2.34)$$

However, it is not differentiable on the medial axis, which is where there is more than one closest surface point [22]. Outside of this set, the level-set curvature can be written as

$$\kappa = \nabla \cdot \left(\frac{\nabla \varphi}{\|\nabla \varphi\|} \right), \quad (2.35)$$

and, for an exact distance function, reduces to

$$\kappa = \nabla^2 \varphi, \quad (2.36)$$

because $\|\nabla \varphi\| = 1$. This means that errors close to the zero level set can change the surface position, normals, and curvature. Sharp features are especially sensitive since their normals shift quickly.

Useful geometric properties. The sign of the distance function gives a fluid–solid mask, its magnitude measures the distance to the wall, and its gradient estimates the surface normal where the function is differentiable. Wall distance also appears directly in turbulence models such as Spalart–Allmaras [111]. Signed distance functions are well-known implicit representations in learning-based shape modeling [83, 73, 37, 69].

Representation assumptions and limitations. Once the continuous SDF is sampled on a Cartesian grid, its accuracy is limited by that discretization. Features smaller than a few voxels can be smoothed or lost, and sharp edges can be softened [83, 73, 95]. The specific preprocessing choices and surface-sensitive evaluation metrics used in this thesis are given in Chapter 3.

2.6 Mathematical Formulation of the Surrogate Problem

Let

$$\mu \in \mathcal{P} \subset \mathbb{R}^P \quad (2.37)$$

denote the available design parameters and

$$\varphi \in \mathcal{X}_\varphi \quad (2.38)$$

the geometric field. The aerodynamic state is

$$\mathbf{q} = (u_x, u_y, u_z, p) \in \mathcal{X}_q, \quad (2.39)$$

with $C_D \in \mathbb{R}$ treated separately.

Stage 1: geometric representation. The volumetric branch maps an SDF to the shared latent space,

$$E_\theta^\varphi : \mathcal{X}_\varphi \longrightarrow \mathbb{R}^d, \quad (2.40)$$

whereas the parameter branch maps the design vector to that same space,

$$N_\theta^\mu : \mathcal{P} \longrightarrow \mathbb{R}^d. \quad (2.41)$$

The parameter map is not called an encoder because it does not reduce the dimensions of the design vectors. A shared decoder

$$D_\theta^\varphi : \mathbb{R}^d \longrightarrow \mathcal{X}_\varphi \quad (2.42)$$

produces

$$\hat{\varphi}_\mu = D_\theta^\varphi (N_\theta^\mu(\mu)), \quad \hat{\varphi}_\varphi = D_\theta^\varphi (E_\theta^\varphi(\varphi)), \quad (2.43)$$

so that the parameter-driven operator is

$$\mathcal{G}_\theta^\mu = D_\theta^\varphi \circ N_\theta^\mu : \mathcal{P} \longrightarrow \mathcal{X}_\varphi, \quad \mu \longmapsto \hat{\varphi}_\mu, \quad (2.44)$$

and the geometry-input reconstruction operator is

$$\mathcal{G}_\theta^\varphi = D_\theta^\varphi \circ E_\theta^\varphi : \mathcal{X}_\varphi \longrightarrow \mathcal{X}_\varphi, \quad \varphi \longmapsto \hat{\varphi}_\varphi. \quad (2.45)$$

Both routes therefore meet in the same latent representation and share the same decoder.

Stage 2: aerodynamic prediction. The volumetric aerodynamic operator is

$$\mathcal{F}_\psi : \mathcal{X}_\varphi \longrightarrow \mathcal{X}_q, \quad \varphi \longmapsto \hat{\mathbf{q}}, \quad (2.46)$$

and outputs the three velocity components together with the pressure. Here, ψ collects the trainable parameters of the aerodynamic network. The operator is realized by the three-dimensional U-Net described in Section 3.4, and it does contain an encoding and a decoding path: a contracting path extracts multiscale features from the distance field, and an expanding path restores them to full resolution. It is nonetheless written as a single operator, rather than as the composition of an encoder and a decoder as in Stage 1, because skip connections pass features directly from each contracting level to the matching expanding level. The output therefore does not depend on the

input only through a single low-dimensional code, and $\mathcal{F}_\psi$ does not factor through a latent space in the way that $\mathcal{G}_\theta^\varphi$ does.

For AhmedML and DrivAerML, drag is represented by a separate scalar regressor acting on pooled Stage 2 encoder features.

Let

$$\mathcal{H}_\psi : \mathcal{X}_\varphi \longrightarrow \mathbb{R}^m \quad (2.47)$$

and

$$\mathcal{R}_\gamma : \mathbb{R}^m \longrightarrow \mathbb{R} \quad (2.48)$$

represent the feature and scalar maps. The feature map shares the parameters ψ of $\mathcal{F}_\psi$ because it is the contracting path of the same network, whose activations are pooled into a descriptor of dimension m , while γ denotes the parameters of the scalar regressor. Then

$$\hat{C}_D = \mathcal{R}_\gamma(\mathcal{H}_\psi(\varphi)), \quad (2.49)$$

and, where both field and drag outputs are used, the aerodynamic map can be written as

$$\mathcal{A}_{\psi,\gamma} : \mathcal{X}_\varphi \longrightarrow \mathcal{X}_q \times \mathbb{R}, \quad \varphi \longmapsto (\mathcal{F}_\psi(\varphi), \mathcal{R}_\gamma(\mathcal{H}_\psi(\varphi))). \quad (2.50)$$

More generally, aerodynamic response also depends on operating conditions,

$$\mathcal{A} : (\varphi, \eta) \longmapsto (\mathbf{q}, C_D). \quad (2.51)$$

Here, η can include freestream speed, Reynolds number, yaw, ground treatment, and wheel state. Each CFD campaign sets these conditions, so the learned mapping is considered geometry-dependent for that specific operating regime.

Integrated surrogate. The framework offers two geometric entry paths, chosen based on the information available for each dataset. If there is a consistently low-dimensional parameterization, the full surrogate is expressed as

$$\mathcal{A}_{\psi,\gamma}(\mathcal{G}_\theta^\mu(\boldsymbol{\mu})), \quad (2.52)$$

where the geometry map is defined through the parameter-driven Stage 1 path

$$\mathcal{G}_\theta^\mu = D_\theta^\varphi \circ N_\theta^\mu. \quad (2.53)$$

Here, the design parameters $\boldsymbol{\mu}$ are first mapped to a latent representation, which is then decoded into the SDF used for aerodynamic prediction.

If a suitable parameterization is not available, or if geometries from different datasets do not share the same design variables, the framework can instead use the volumetric SDF path:

$$\mathcal{G}_\theta^\varphi = D_\theta^\varphi \circ E_\theta^\varphi. \quad (2.54)$$

This architecture encodes and reconstructs an SDF based on geometry, then sends it to the aerodynamic model. This approach creates a shared geometric interface, even if there is no clear low-dimensional parameter description.

The mapping from SDF to flow can also be viewed directly as

$$\mathcal{F}_\psi(\varphi). \quad (2.55)$$

Equation (2.55) separates aerodynamic prediction from a given SDF, while the full surrogate also includes the earlier Stage 1 geometric reconstruction. As a result, datasets like DrivAerML can follow the parameter-driven approach when design parameters are available. The SDF-based approach is used when parameters are missing or when a shared parameterization does not fit all geometries. Chapter 3 explains the training and evaluation protocols for each dataset.

2.7 Chapter Summary

This chapter explained the main physical and mathematical ideas used in the thesis. It discussed vehicle aerodynamics, turbulence modeling, aerodynamic forces, and the main flow features of the Ahmed and DrivAer shapes. The chapter also covered reduced-order models and Free-Form Deformation. It also introduced convolutional neural networks and recent methods for learning on complex geometries. The chapter described the main properties of signed distance functions and defined the geometric and aerodynamic mappings used in this work. Chapter 3 explains how these models are implemented, trained, and evaluated.

Methodology

This chapter presents the experimental setup and implementation procedures employed in this thesis. While Chapters 1 and 2 introduced the mathematical formulation of the two-stage framework, the current chapter addresses the computational execution, with emphasis on data preparation, network architectures, loss functions, and evaluation methodologies.

The remainder of this chapter is organized as follows. Section 3.1 positions the proposed framework in relation to state-of-the-art learning-based aerodynamic surrogates. Section 3.2 describes the datasets, task-specific train/test splits, and preprocessing workflows. Section 3.3 outlines the quantitative metrics and evaluation protocols. Section 3.4 details the neural network architectures, loss functions, integrated drag regressor, and the overall training pipeline.

3.1 Deep learning for aerodynamic prediction: state of the art

Chapter 1 provided a comparative overview of parameter-based and geometry-based aerodynamic surrogate models. In the context of geometry-based methods, Bhatnagar et al. [13] demonstrated in two dimensions the mapping of Cartesian signed distance fields to flow fields using convolutional architectures. Similarly, Thuerey et al. [116] predicted two-dimensional pressure and velocity fields on Cartesian grids with a U-Net architecture [97], although their approach represented geometry with a binary airfoil mask instead of a signed distance field.

Garcia-Fernandez et al. [31] employ a signed-distance representation for heavy ground vehicles. Their neural network utilizes two distance-field channels, corresponding to the vehicle and the road, as well as a boundary-condition channel. The model predicts velocity and pressure, normalized

by the freestream velocity and by ρU_∞^2 , respectively, in accordance with the convention established in Equation (2.3). Two of their findings are particularly relevant to the present study. The authors identify the boundary layer as the primary source of error and demonstrate that employing a mesh graded towards the body reduces both maximum and mean errors compared to a uniform grid. This approach serves as the two-dimensional analog of the resolution limitation quantified in Section 4.4. Additionally, they report a computational speed-up of four orders of magnitude relative to the reference solver, which aligns with the cost analysis presented in Section 6.4. However, their study is restricted to two dimensions and predicts fields for a fixed family of geometries, rather than reconstructing the geometry itself. Jacob et al. [48] extended the SDF-based convolutional approach to three-dimensional vehicle geometries by employing a modified U-Net to predict velocity fields and drag coefficients from voxelized signed distance fields. A primary limitation of these methods is the reliance on a fixed Cartesian grid, which constrains the level of geometric detail that can be represented. Recent transformer-based operators overcome this constraint by eliminating the fixed prediction grid [124, 4]. Alkin et al. [4] separate geometry encoding from field decoding in their GP-UPT model, enabling a compact point-cloud encoding of the surface to be queried at arbitrary locations. This approach allows for volumetric predictions on meshes with approximately twenty million cells, without requiring a CFD simulation mesh as model input. On the AhmedML dataset [6], GP-UPT achieves a relative L^2 error of 0.0272 for the total pressure coefficient, compared to 0.0418 for GINO [62]. For DrivAerML [7], predictions of surface pressure and wall-shear stress yield drag and lift correlations of approximately $R^2 = 0.97$ after integrating the predicted surface quantities. AB-UPT [3] adopts the same query-based neural-field paradigm and provides the volumetric field errors reported in Table 6.4.

3.2 From CFD simulations to training data

The proposed approach retains the voxel-based distance-field representation and its associated resolution bound. However, it differs from previous methods in that the geometry is reconstructed by a learned stage, which is trained and evaluated before predicting any aerodynamic quantity. Consequently, the composed pipeline operates on a design description instead of a reference surface. Table 6.4 compares the field errors of these benchmark models with the present results.

Experimental protocol at a glance

Three vehicle datasets are used in the main experiments: AhmedML, DrivAerML, and the industrial Siemens collection. Stage 1 and Stage 2 solve

different learning problems and are therefore trained with task-specific splits. The evaluation logic used in the thesis is as follows:

1. Stage 1 is trained or adapted on its geometry-training data and is evaluated on a held-aside geometry group.
2. The direct Stage 2 model is trained using reference SDFs and CFD fields.
3. The end-to-end Stage 2 model is trained separately using Stage 1 reconstructed SDFs and the same type of CFD targets.
4. The reported direct and end-to-end field results are evaluated on the test flow designs defined for the corresponding dataset.

The direct and end-to-end aerodynamic models are trained separately. The direct model uses the reference SDF, while the end-to-end model uses the Stage 1 reconstructed SDF during training and inference.

Datasets

AhmedML. AhmedML [6] comprises 500 parametric Ahmed-body variants. The dataset was not fully processed at the outset. During model development, the number of geometries in the working collection was progressively increased while monitoring the reconstruction performance of Stage 1. When reconstruction quality reached a level deemed sufficient for subsequent aerodynamic experiments, the working collection was fixed at 380 processed designs. Thus, the selection of 380 designs reflects the iterative development procedure for the geometric surrogate, rather than a predetermined fraction of the AhmedML dataset.

Stage 1 is trained on 285 designs, validated on 38 designs, and evaluated on 57 designs. For the volumetric-flow experiments, the final Stage 2 working collection consists of 299 out of the 380 processed AhmedML geometries. This collection was established through iterative development of the aerodynamic surrogate. The dataset for the aerodynamic learning task is partitioned into 224 training designs, 30 validation designs, and 45 test designs for field evaluation.

The same 45 test flow designs are used to report both the direct and end-to-end field results. The distinction lies in the input provided to Stage 2: the direct model receives the reference signed distance function (SDF), whereas the end-to-end model receives an SDF reconstructed by Stage 1 from the design parameters.

The drag experiment utilizes the same 299-design base collection but employs a separate scalar-regression split: 224 training designs, 29 validation designs, and 46 test drag designs. This partitioning is independent of the geometry and volumetric-flow experiments.

DrivAerML and Siemens transfer. DrivAerML [7] comprises 500 morphed notchback vehicle variants resolved with a hybrid RANS-LES formulation. In this thesis DrivAerML serves as a source for transfer learning and fine-tuning rather than for training models from scratch. The quantity of DrivAerML data in the working collection was incrementally increased during development until the geometric model achieved a reconstruction quality deemed sufficient for subsequent experiments, giving a final working collection of 97 configurations. The Siemens industrial collection comprises 20 production vehicle configurations, whose CFD solutions are steady Reynolds-averaged Navier-Stokes.

To enable realistic geometry reconstruction, DrivAerML and Siemens data are combined in Stage 1 through the volumetric signed distance function branch, giving a pooled collection of 117 geometries. Each geometry is represented as an SDF, processed by the volumetric encoder, and reconstructed through the shared decoder, so that no common parameter space need be defined between two design families that share no parameterization. The pooled model is partitioned into a training and a held-aside group, the latter comprising a tenth of the combined collection.

The industrial reconstruction is additionally specialized by a separate fine-tune on the twenty industrial configurations alone, with fifteen used for adaptation and five held aside, and it is this specialized model that produces the industrial geometric metrics. The two models therefore have distinct partitions, both reported in Table 3.1. Geometry metrics for DrivAerML and Siemens are reported separately in Chapter 4. In addition to the mixed DrivAerML-Siemens configuration, a separate DrivAerML-only Stage 1 fine-tuning experiment is conducted using the available DrivAerML morphing parameters. In this experiment, the parameter-driven branch is employed, mapping the DrivAerML design variables to the latent representation and decoding them into the corresponding SDF. This approach evaluates the parameter-based formulation independently on DrivAerML, whereas the mixed DrivAerML-Siemens transfer utilizes the volumetric SDF branch.

Stage 2 is adapted to utilize mixed DrivAerML and Siemens data, with the direct and end-to-end configurations employing distinct task-specific training and evaluation groups. The direct model utilizes a reduced DrivAerML transfer set, while the end-to-end model employs the larger working collection. In both configurations, Siemens maintains a consistent partition of 10 designs for training and 10 for evaluation. The precise dataset counts are provided in Table 3.1, and aerodynamic results are reported separately for each design family.

For the mixed DrivAerML and Siemens end-to-end experiments, both families follow the volumetric Stage 1 path defined above. The additional DrivAerML-only experiment instead follows the parameter-driven Stage 1 path using its

morphing parameters.

Drag prediction is evaluated independently using leave-one-out regression on the 97-design DrivAerML working set, as detailed later in this chapter. Geometry, direct-flow, and end-to-end results therefore refer to different learning tasks and are interpreted using the corresponding training and evaluation groups in Table 3.1.

From solver output to tensors

Each aerodynamic dataset is converted into two learning objects: a signed distance field and a set of volumetric aerodynamic fields. All cached reference data are stored on a regular 128^3 Cartesian grid inside a fixed computational box for each dataset. For AhmedML, Stage 1 reconstructs directly at 128^3 . For DrivAerML and Siemens, Stage 1 reconstructs at 64^3 because these geometries are substantially larger and more complex than the AhmedML configurations, and the reduced resolution helps shorten the Stage 1 training time and computational cost. The reconstructed SDF is then resampled to 128^3 before being used in Stage 2. The preprocessing pipeline has four main steps.

1. Surface preparation. For each configuration, a watertight surface triangulation is either extracted or checked. The surface must be high quality because the SDF’s inside-outside classification relies on having a consistent, closed boundary.

2. Signed distance construction. The SDF is calculated from the surface using the narrow-band tools in OpenVDB [77]. Exact distances are measured near the surface, and then the field is extended through the surrounding volume before being sampled on the target Cartesian grid.

3. Flow-field interpolation and alignment. The CFD velocity components and pressure are resampled from the solver mesh onto the same Cartesian grid as the SDF. Geometry and flow must match voxel by voxel in world coordinates. This alignment is checked directly because solver meshes, surface files, and sampling boxes sometimes use different axis orders or coordinate systems.

4. Field standardization. Each aerodynamic channel is standardized separately using statistics calculated only from the training data. For a field q ,

$$q_{\text{std}} = \frac{q - \mu_q^{\text{train}}}{\sigma_q^{\text{train}}}, \quad (3.1)$$

where μ_q^{train} and σ_q^{train} denote the mean and standard deviation computed from the training set. These parameters are stored and subsequently applied without modification to the validation and evaluation data. Standardization is inverted prior to calculating the relative-error metrics presented in this thesis.

All preprocessing and evaluation procedures therefore use a dataset-specific coordinate transformation so that the convention defined in Section 2.1 is respected. Geometry and field axes are transformed or relabeled as required.

The computational workflow was executed on the Galileo100 system at CINECA. Downloaded files are validated prior to cache construction, and cached tensors are checked for shape, statistical properties, and coordinate consistency before model training. Neural network training is performed on GPU nodes.

3.3 Metrics and evaluation protocol

Field accuracy

Let f denote a reference volumetric field, $\hat{f}$ its prediction, and Ω_h the Cartesian voxel grid. The primary aerodynamic metric is the relative L^2 error, evaluated separately for each predicted quantity and each test design.

Definition 3.1 (Relative L^2 error) *Let f and $\hat{f}$ denote a reference field and its prediction, respectively, and let $\mathcal{F} \subset \Omega_h$ denote the fluid region, defined as the set of voxels outside the body according to the sign of the reference signed distance field. The relative L^2 error is defined as*

$$\text{RelL2}(\hat{f}, f) = \frac{\|\hat{f} - f\|_{2,\mathcal{F}}}{\|f\|_{2,\mathcal{F}}}. \quad (3.2)$$

Here, $\|\cdot\|_{2,\mathcal{F}}$ denotes the Euclidean norm restricted to the fluid region $\mathcal{F}$. When reported as a percentage, the value of $\text{RelL2}(\hat{f}, f)$ is multiplied by 100.

The restriction to the fluid domain is essential because solid voxels are masked and should not artificially reduce the reported error. The metric is computed after inverting the training-time z-score standardization. The resulting fields are therefore evaluated in the dataset convention that existed before neural-network standardization.

Error evaluation in the end-to-end setting. The reconstructed geometry never defines the grid on which errors are computed. All reference quantities are stored on one fixed 128^3 Cartesian grid per dataset: the CFD velocity and pressure are interpolated from the solver mesh onto this grid once,

during preprocessing, and the reference SDF is sampled on the same grid. Stage 2 always predicts on this grid, whatever geometry it receives. For AhmedML, Stage 1 reconstructs directly at 128^3 ; for DrivAerML and Siemens it reconstructs at 64^3 , and the reconstructed SDF is resampled to 128^3 before entering Stage 2. This resampling changes only the geometric input, not the location of the predicted values, so the prediction and the reference are compared voxel by voxel without any interpolation of the flow fields.

The comparison is restricted to the fluid region $\mathcal{F}$ of the reference geometry, as in Definition 3.1, and not to that of the reconstruction. Where the reconstructed surface extends beyond the reference one, the corresponding voxels belong to the reference fluid, and the predicted values there are compared with the true flow, so this geometric error is counted in the reported flow error. Where the reconstruction falls short of the reference surface, the affected voxels lie inside the reference body and are excluded, but the resulting change in the predicted flow is still measured in the surrounding fluid. The end-to-end figures therefore measure the cost of the composed pipeline, including its geometric error, on a fixed set of evaluation voxels that is the same for the direct and end-to-end configurations.

Because the reference fields are the CFD solution interpolated onto the Cartesian grid, all errors are measured against this gridded reference rather than at the original solver mesh points. The interpolation from mesh to grid is common to the direct and end-to-end evaluations and therefore does not enter the difference between them.

Dataset-level values are means of per-design errors. The per-design error curves shown in Chapters 5 and 6 are retained because the mean alone can hide substantial variation between geometries.

Geometric accuracy

Stage 1 reconstructs a signed distance field, with primary emphasis on the zero level set. A low volumetric field error does not necessarily indicate accurate recovery of the vehicle surface. To address this limitation, two surface-oriented metrics are employed: the Dice coefficient [74], which quantifies the overlap of the enclosed solid regions, and the Hausdorff distance, which measures the difference between the reconstructed and reference surfaces [47].

The reconstructed surface is defined from the trilinearly interpolated SDF by

$$\Gamma(\varphi) = \{\mathbf{x} \in \Omega \mid \tilde{\varphi}(\mathbf{x}) = 0\}, \quad (3.3)$$

where $\tilde{\varphi}$ denotes the trilinear interpolant of φ on Ω_h .

Definition 3.2 (Dice coefficient) Let $\mathcal{S}(\varphi) = \{\mathbf{x} \in \Omega_h : \varphi(\mathbf{x}) < 0\}$ denote the set of voxels enclosed by the surface. The Dice coefficient of the reconstruction $\hat{\varphi}$ with respect to the reference φ is the normalized overlap of the two enclosed volumes [74],

$$\text{Dice}(\hat{\varphi}, \varphi) = \frac{2|\mathcal{S}(\hat{\varphi}) \cap \mathcal{S}(\varphi)|}{|\mathcal{S}(\hat{\varphi})| + |\mathcal{S}(\varphi)|} \in [0, 1]. \quad (3.4)$$

The Dice coefficient quantifies the overlap between reconstructed and reference solids; however, it does not account for the largest local surface errors. Consequently, a small region with substantial deviation may have minimal impact on the Dice score, despite significant geometric error. To mitigate this limitation, the Hausdorff distance is employed as a complementary measure of surface distance.

Definition 3.3 (Hausdorff distance) For two surfaces $A, B \subset \mathbb{R}^3$, let

$$h(A, B) = \sup_{\mathbf{x} \in A} \inf_{\mathbf{y} \in B} \|\mathbf{x} - \mathbf{y}\| \quad (3.5)$$

denote the directed Hausdorff distance, the largest distance from a point of A to its nearest point of B . The Hausdorff distance between the reconstructed and reference surfaces is its symmetrization,

$$d_H(\Gamma(\hat{\varphi}), \Gamma(\varphi)) = \max\{h(\Gamma(\hat{\varphi}), \Gamma(\varphi)), h(\Gamma(\varphi), \Gamma(\hat{\varphi}))\}. \quad (3.6)$$

The classical Hausdorff distance is sensitive to isolated outliers [47]. For the vehicle experiments, the thesis therefore reports the 95th-percentile Hausdorff distance,

$$h_{95}(A, B) = P_{95}\left(\left\{\inf_{\mathbf{y} \in B} \|\mathbf{x} - \mathbf{y}\| : \mathbf{x} \in A\right\}\right), \quad (3.7)$$

$$HD_{95} = \max\{h_{95}(A, B), h_{95}(B, A)\}.$$

We use the 95th percentile of the two directed nearest-surface distance distributions. Distances are calculated using the physical voxel spacing for each dataset and then divided by the diagonal of the computational box, making the reported values dimensionless. For example, in the Ahmed domain, the diagonal is about 2.62 m, so the mean normalized $HD_{95} = 0.0044$ mentioned later is roughly 11.5 mm.

MSE and MAE are also monitored during geometry and field reconstruction:

$$\text{MSE}(\hat{f}, f) = \frac{1}{|\Omega_h|} \sum_{\mathbf{x} \in \Omega_h} \left(\hat{f}(\mathbf{x}) - f(\mathbf{x})\right)^2, \quad (3.8)$$

and

$$\text{MAE}(\hat{f}, f) = \frac{1}{|\Omega_h|} \sum_{\mathbf{x} \in \Omega_h} \left|\hat{f}(\mathbf{x}) - f(\mathbf{x})\right|. \quad (3.9)$$

These metrics facilitate the optimization process. For geometric evaluation, the primary metrics are the Dice coefficient and the 95th percentile Hausdorff distance (HD_{95}). For aerodynamic evaluation, the principal metric is the relative L^2 error.

For drag assessment, relative error and mean absolute error in drag counts are employed as the primary engineering metrics. The coefficient of determination is additionally reported as a supplementary statistical measure.

Definition 3.4 (Coefficient of determination) For test-set drag values $C_{D,i}$ with mean $\bar{C}_D$ and predictions $\hat{C}_{D,i}$,

$$R^2 = 1 - \frac{\sum_i (C_{D,i} - \hat{C}_{D,i})^2}{\sum_i (C_{D,i} - \bar{C}_D)^2}. \quad (3.10)$$

The interpretation of R^2 depends on the variance of the reference drag coefficients. When target values are closely clustered, even minor absolute errors may result in a moderate R^2 . Since this thesis uses subsets of the AhmedML and DrivAerML datasets, we use relative error and drag-count mean absolute error (MAE) as the primary metrics for evaluating drag.

Evaluation and verification practice

We define all main splits based on unique vehicle designs rather than on individual voxels or slices. The flow experiments use the test set groups listed in Table 3.1. During evaluation, inference mode is used and stochastic training operations are turned off.

Qualitative verification is conducted alongside numerical metrics. Reference and predicted geometries are evaluated using zero-level-set contours and isosurfaces, while aerodynamic fields are compared on standard cut planes. These visual assessments facilitate the identification of local geometric errors, coordinate misalignment, wake discrepancies, and other issues that may not be apparent in dataset-level averages.

3.4 The two-stage prediction architecture

Design rationale

Figure 3.1 illustrates the implementation of the two-stage operators defined in Section 2.6. Stage 1 provides the reconstructed signed distance field that acts as the geometric interface, and Stage 2 maps this field to the aerodynamic quantities. The experiment-specific choice of Stage 1 entry path follows the protocol in Table 3.1.

3. METHODOLOGY

Table 3.1: Task-specific training and evaluation protocol. Training and evaluation groups are defined separately for geometry reconstruction, direct flow prediction, end-to-end flow prediction, and drag regression. The AhmedML geometry task uses all 380 configurations; its flow and drag tasks use the 299 retained after excluding incomplete field data.

Dataset	Task	Input	Model training	Evaluation
AhmedML	Geometry	Parameters	Stage 1: 285 train + 38 validation, of 380	57 designs
AhmedML	Direct flow	Reference SDF	Stage 2: 224 train + 30 validation, of 299	45 designs
AhmedML	End-to-end flow	Reconstructed SDF	Stage 2 E2E: 224 train + 30 validation, of 299	45 designs
AhmedML	Drag	Pooled encoder features	224 train + 29 held aside, of 299	46 designs
DrivAerML and Siemens	Geometry	SDF	Stage 1: 88 DrivAerML + 18 Siemens	9 DrivAerML designs + 2 Siemens designs
DrivAerML and Siemens	Direct flow	Reference SDF	Stage 2 direct: 34 DrivAerML + 10 Siemens	10 DrivAerML designs
DrivAerML and Siemens	End-to-end flow	Reconstructed SDF	Stage 2 E2E: 86 DrivAerML + 10 Siemens	10 DrivAerML designs
DrivAerML	Drag	Pooled encoder features	Leave-one-out over 97 designs	97 out-of-sample predictions
Siemens	Geometry	SDF	Separate industrial fine-tune: 15 configurations	5 designs
Siemens	Direct flow	Reference SDF	Stage 2 direct	10 designs
Siemens	End-to-end flow	Reconstructed SDF	Stage 2 E2E	10 designs

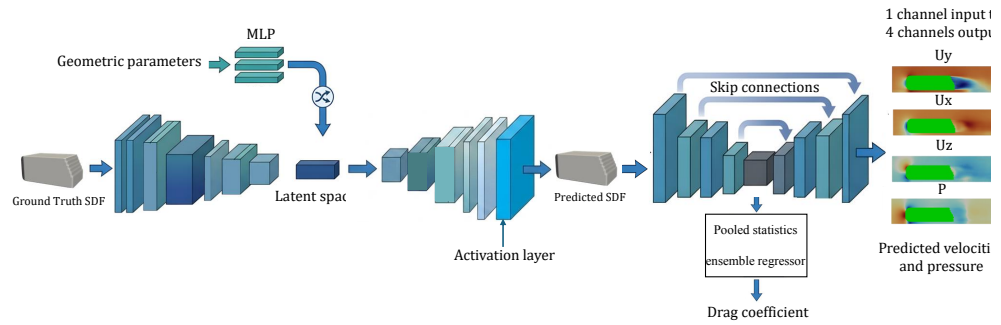


Figure 3.1: The two-stage surrogate. Stage 1 predicts the signed distance field. In the E2E configuration, reconstructed Stage 1 SDFs are used already during Stage 2 U-Net training/fine-tuning and again at inference, so the flow network is exposed to the same type of geometric input in both phases.

Stage 1: hybrid geometric network

Stage 1 consists of a dual-branch geometric network with a shared latent representation and decoder, as illustrated in Figure 3.2. The volumetric branch employs a 3D convolutional encoder to compress a signed distance function (SDF) into a latent vector of size d . The parameter branch utilizes a multilayer perceptron to map a design vector into the same latent space.

AhmedML uses a latent dimension of $d = 32$ and reconstructs directly at 128^3 . For DrivAerML and Siemens, the configuration uses $d = 128$ and reconstructs at 64^3 , then resamples to 128^3 when Stage 2 needs that input size.

When paired parameter and SDF descriptions are used in the source geometric training, a latent-alignment term encourages the parameter and volumetric branches to produce nearby codes, $z^{(p)}$ and $z^{(s)}$ respectively, for the same geometry. The shared decoder can therefore operate on either representation.

Hidden layers use LeakyReLU activations [70]. The final decoder layer is linear so that the network predicts a signed distance value without constraining its range.

An optional smoothed hyperbolic-tangent transformation places additional emphasis on the zero-level-set neighbourhood [20]:

$$\chi_{\varepsilon_{\text{act}}}(r) = \frac{1}{2} \left[1 - \tanh \left(\frac{r}{\varepsilon_{\text{act}}} \right) \right]. \quad (3.11)$$

Here, $\varepsilon_{\text{act}} > 0$ determines the thickness of the smooth transition. A smaller value results in a sharper occupancy transition, while a larger value broadens the region, causing reconstruction errors to influence the auxiliary objective over a broader area. This transformation is applied exclusively to the loss function. The decoder continues to output the raw reconstructed signed distance function (SDF).

The Stage 1 objective is

$$\mathcal{L}_{\text{rec}} = \lambda_{\text{SDF}} \mathcal{L}_{\text{SDF}} + \lambda_{\text{latent}} \mathcal{L}_{\text{latent}} + \lambda_{\text{act}} \mathcal{L}_{\text{act}}, \quad (3.12)$$

with the SDF reconstruction term

$$\mathcal{L}_{\text{SDF}} = \text{MSE}(\hat{\Phi}, \Phi), \quad (3.13)$$

the latent-alignment term

$$\mathcal{L}_{\text{latent}} = \left\| z^{(p)} - z^{(s)} \right\|_2^2, \quad (3.14)$$

and, when activated, the auxiliary surface-focused term

$$\mathcal{L}_{\text{act}} = \text{MSE}(\chi_{\varepsilon_{\text{act}}}(\hat{\Phi}), \chi_{\varepsilon_{\text{act}}}(\Phi)). \quad (3.15)$$

The MSE quantities are averaged over the spatial grid and mini-batch. The reconstruction term applies to outputs generated from both Stage 1 branches. Table 3.2 details the reconstruction network architecture. The parameter branch consists of a multilayer perceptron with two hidden layers of widths 128 and 256, each followed by a leaky rectified linear activation, and a linear output layer projecting onto the latent space. The volumetric encoder compresses the input field using a sequence of strided three-dimensional convolutions, followed by a linear projection. Both branches map to a latent space of dimension 32. The shared decoder expands the latent vector via a linear projection to a feature volume of 64×16^3 and restores the full resolution through three upsampling stages: 16^3 to 32^3 , 32^3 to 64^3 , and 64^3 to 128^3 . The output layer is a linear convolution without activation, resulting in an unbounded distance field whose zero level set defines the surface.

Stage 2: volumetric flow prediction

Stage 2 is implemented as a three-dimensional U-Net [97, 19], receiving the SDF as its single input channel and outputting four target fields: the velocity components (U_x, U_y, U_z) and pressure (P), as shown in Figure 3.3.

The U-Net is used in two separately trained configurations.

For the direct model,

$$(\phi_i^{\text{ref}}, \mathbf{q}_i^{\text{CFD}}), \quad \mathbf{q}_i^{\text{CFD}} = (U_x, U_y, U_z, P)_i$$

forms the training pair.

For the end-to-end model, Stage 1 first generates

$$\hat{\phi}_i = \text{Stage 1}(\text{design}_i),$$

and the training pair becomes

$$(\hat{\phi}_i, \mathbf{q}_i^{\text{CFD}}).$$

The predicted CFD quantities remain the same in both configurations, while the geometric input follows the corresponding direct or end-to-end protocol. The end-to-end U-Net is therefore trained on the same type of reconstructed geometric input that it receives during end-to-end inference.

The U-Net structure combines both local and global information. Skip connections keep important high-resolution details near the body and shear

3.4. The two-stage prediction architecture

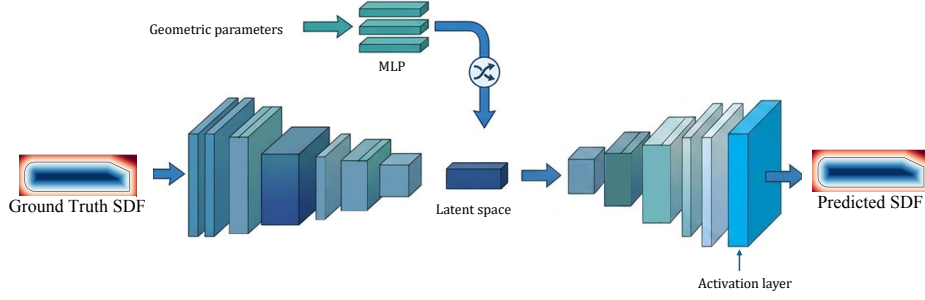


Figure 3.2: The geometric stage.

Table 3.2: Layer structure of the reconstruction network. Output shapes are given as channels $\times$ spatial resolution for a single design.

Component	Layer	Output shape	Notes
<i>Parameter branch</i>			
Input	design vector	8	standardised on the train split
Hidden 1	linear	128	LeakyReLU
Hidden 2	linear	256	LeakyReLU
Latent	linear	32	no activation
<i>Volumetric encoder</i>			
Input	distance field	1×128^3	
Contraction	strided convolutions	coarse volume	normalisation and LeakyReLU
Latent	linear	32	no activation
<i>Shared decoder</i>			
Projection	linear	64×16^3	
Upsampling 1	upsampling + convolution	32^3	
Upsampling 2	upsampling + convolution	64^3	
Upsampling 3	upsampling + convolution	16×128^3	
Output	convolution 3^3	1×128^3	no activation
Parameter branch, trainable parameters			42,400

layers. The contracting path lets the model see a larger area, which helps capture the downstream wake.

The flow loss is evaluated only in the fluid domain. Let $\mathcal{Q} = \{U_x, U_y, U_z, P\}$ and let $M_i(\mathbf{x})$ be one in the fluid region and zero inside the body. The number of contributing voxels is

$$N_{f,i} = \sum_{\mathbf{x} \in \Omega_h} M_i(\mathbf{x}), \quad (3.16)$$

and the multi-field loss is

$$\mathcal{L}_{\text{flow}} = \frac{1}{B|\mathcal{Q}|} \sum_{i=1}^B \sum_{q \in \mathcal{Q}} \frac{1}{N_{f,i}} \sum_{\mathbf{x} \in \Omega_h} M_i(\mathbf{x}) (\hat{q}_i(\mathbf{x}) - q_i(\mathbf{x}))^2, \quad (3.17)$$

where B denotes the number of designs in a mini-batch. The four channels receive equal weight after training-time normalization. Masking prevents solid voxels, whose field values are not physically part of the fluid solution, from contributing to the objective.

Table 3.3 shows the aerodynamic network’s layer structure. The model is a 3D U-Net with three contracting blocks and a bottleneck. Each contraction step halves the spatial resolution and increases channels from 32 to 128. The expanding path is the reverse of the contracting path. Skip connections link encoder feature maps to decoder inputs at the same resolution, helping to recover details lost during contraction. Group normalization (eight groups) and dropout (rate 0.1) are used throughout. The output layer has four channels, one for each predicted field, and does not use an activation function. The network has 3,699,941 trainable parameters. Table 3.4 lists the optimization and loss settings for this network and the reconstruction network.

Drag prediction

The drag coefficient is not calculated by integrating the predicted volumetric fields. To perform a physically consistent surface integration, both pressure and wall shear stress are needed at high near-wall resolution. However, Stage 2 predicts volumetric velocity and pressure on the Cartesian grid and does not include a specific wall-shear-stress field.

As a result, drag is handled as a separate scalar regression problem. The regressors are fitted after the U-Net has been trained on the flow fields, on features extracted with its weights held fixed, so no gradient from the drag target reaches the network and the field training is unaffected by the drag task. Features are taken from the contracting blocks and bottleneck of the Stage 2 U-Net. These internal representations cover larger spatial regions and include information about both local geometry and flow interaction, as well as the global wake.

3.4. The two-stage prediction architecture

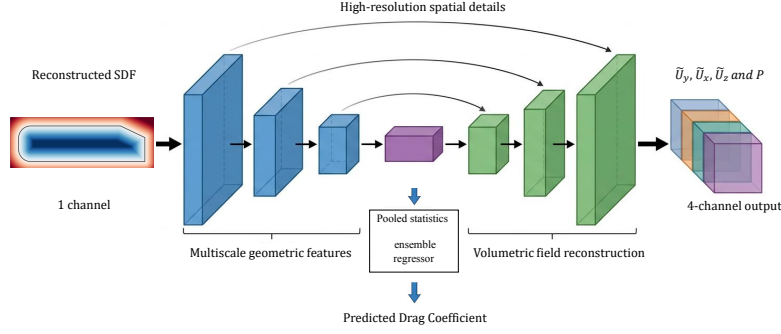


Figure 3.3: The predictive stage. The signed distance field enters as a single channel; multiscale geometric features are extracted by the contracting path and returned to full resolution by the expanding path, with skip connections preserving high-resolution spatial detail, yielding the four-channel volumetric output. The drag coefficient is regressed from pooled encoder features.

Table 3.3: Layer structure of the aerodynamic network. Output shapes are given as channels $\times$ spatial resolution for a single design.

Block	Operation	Output shape	Notes
<i>Contracting path</i>			
Input	distance field	1×128^3	reconstructed SDF
Encoder 1	convolution block	32×128^3	
Pool 1	downsampling	32×64^3	
Encoder 2	convolution block	64×64^3	
Pool 2	downsampling	64×32^3	
Encoder 3	convolution block	128×32^3	
Pool 3	downsampling	128×16^3	
Bottleneck	convolution block	128×16^3	
<i>Expanding path</i>			
Decoder 3	upsampling + skip	32^3	concatenates Encoder 3
Decoder 2	upsampling + skip	64^3	concatenates Encoder 2
Decoder 1	upsampling + skip	128^3	concatenates Encoder 1
Output	convolution	4×128^3	U_x, U_y, U_z, p ; no activation
Normalisation / dropout			GroupNorm, 8 groups / 0.1
Trainable parameters			3,699,941

Let $\mathbf{F}^{(k)} \in \mathbb{R}^{C_k \times N_k}$ denote the feature map returned by the k -th of these blocks, with $k = 1, 2, 3$ for the three contracting blocks and $k = 4$ for the bottleneck, where C_k is the number of channels of that block and N_k the number of voxels at its resolution. The entry $F_{cj}^{(k)}$ is the value of channel $c \in \{1, \dots, C_k\}$ at voxel $j \in \{1, \dots, N_k\}$. For feature map $\mathbf{F}^{(k)}$, the channel-wise mean, maximum, and standard deviation are

$$a_c^{(k)} = \frac{1}{N_k} \sum_j F_{cj}^{(k)}, \quad m_c^{(k)} = \max_j F_{cj}^{(k)}, \quad s_c^{(k)} = \sqrt{\frac{1}{N_k} \sum_j (F_{cj}^{(k)} - a_c^{(k)})^2}, \quad (3.18)$$

and these statistics are concatenated over the three contracting blocks and the bottleneck:

$$\mathbf{g} = [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(4)}, \mathbf{m}^{(1)}, \dots, \mathbf{m}^{(4)}, \mathbf{s}^{(1)}, \dots, \mathbf{s}^{(4)}], \quad (3.19)$$

where $\mathbf{a}^{(k)} = (a_1^{(k)}, \dots, a_{C_k}^{(k)})$, and $\mathbf{m}^{(k)}$ and $\mathbf{s}^{(k)}$ are defined likewise. With channel widths C_k of 32, 64, 128 and 128, the descriptor has dimension $3(32 + 64 + 128 + 128) = 1056$. The descriptor summarizes each feature channel by calculating its mean, maximum, and standard deviation. It does not keep the exact spatial location of the activations.

Because each geometry has only one drag value, two lightweight regressors are used instead of another neural network. An ExtraTrees model [34] and a ridge regressor [45] are each trained on the standardized descriptor $\tilde{\mathbf{g}}$, and their predictions are averaged equally,

$$\hat{C}_D = \frac{1}{2} [f_{\text{ET}}(\tilde{\mathbf{g}}) + f_{\text{R}}(\tilde{\mathbf{g}})]. \quad (3.20)$$

The tree ensemble models nonlinear relationships, and ridge regression adds a stable linear part. This combination helps capture both complex and straightforward patterns in the data.

AhmedML drag is tested using a fixed split for training, validation, and testing. For DrivAerML drag, we use leave-one-out evaluation across all 97 designs [114]. For each design, we scale the descriptors and train both scalar regressors on the other 96 designs, then test on the one left out.

The DrivAerML procedure is therefore leave-one-out with respect to the scalar drag regressor, while the learned neural feature extractor remains fixed.

Training protocols

All neural models are trained with AdamW [65] and gradient-norm clipping [84]. In Stage 1, we start with a learning rate of 5×10^{-4} , use a 20-epoch

linear warmup, then apply cosine annealing [66], and set gradient clipping at 1.0. In Stage 2, the initial learning rate is 5×10^{-5} , with cosine annealing and the same clipping threshold.

We use transfer learning to move from AhmedML to the more complex DrivAerML and Siemens settings [82, 125]. For Stage 1 transfer, the AhmedML geometric model is used for initialization. For Stage 2, the AhmedML aerodynamic model provides the starting representation.

For the end-to-end aerodynamic configuration, the order of operations is:

1. train Stage 1;
2. retain and freeze the selected Stage 1 weights;
3. run Stage 1 on every Stage 2 training or fine-tuning design to produce reconstructed SDFs;
4. train the end-to-end Stage 2 U-Net using those reconstructed SDFs and the corresponding CFD targets;
5. freeze both stages and evaluate the complete chain on the held-out flow designs.

This step-by-step approach replaces joint backpropagation through the entire two-stage model.

For the realistic-vehicle transfer, DrivAerML and Siemens are adapted together rather than by training separate neural networks. Stage 1 follows the joint geometry protocol in Table 3.1.

Stage 2 starts from the AhmedML aerodynamic model and is fine-tuned according to the direct or end-to-end protocol listed in the same table. The two configurations are trained separately and use their corresponding geometric inputs and task-specific DrivAerML partitions.

If the network capacity changes within a geometric domain, a partial warm start is applied. Compatible convolutional parameters are kept, but layers next to a changed latent bottleneck are reset.

3. METHODOLOGY

Table 3.4: Reference configurations of Stage 1, Stage 2, and the drag regressor.

Configuration Parameter	Stage 1 (Geometry Network)	Stage 2 (Flow Network)
<i>Architecture</i>		
Input	AhmedML: design parameters or SDF branch as required; realistic DrivAerML/Siemens transfer: SDF through the volumetric branch	SDF (128^3)
Output	AhmedML: reconstructed SDF at 128^3 ; DrivAerML/Siemens: reconstructed SDF at 64^3	U_x, U_y, U_z, P fields at 128^3
Latent Dimension	AhmedML: $d = 32$; DrivAerML/Siemens: $d = 128$	Bottleneck size: 128×16^3
Base Channels	32	32
<i>Training & Optimization</i>		
Batch Size	2	4
Learning Rate	5×10^{-4}	5×10^{-5}
Optimizer	AdamW	AdamW
LR Schedule	20-epoch warmup + cosine annealing	Cosine annealing
Max Epochs	300	200 (Early stopping patience: 80)
Gradient Clipping	1.0 (max norm)	1.0 (max norm)
<i>Loss Objectives</i>		
Loss Terms	SDF reconstruction, latent alignment, and optional auxiliary activation	Fluid-masked multi-field MSE
<i>Drag regressor</i>		
Features	Mean, maximum, and standard deviation pooled channel-wise over the three contracting blocks and the bottleneck of Stage 2	
Model	Extremely randomized trees and ridge regression, averaged with equal weight	
Evaluation	Fixed train-validation-test partition for AhmedML; leave-one-out evaluation over the 97-design working subset for DrivAerML	

Learning Geometry: A Latent Representation of Vehicle Shapes

This chapter provides an independent evaluation of Stage 1, separate from the aerodynamic predictor. The objective is to determine whether the learned representation maintains the physical surface with adequate fidelity before the reconstructed geometry is transferred to Stage 2.

This chapter addresses Research Question 1 introduced in Chapter 1: whether a compact learned representation can reconstruct geometries of increasing complexity while preserving the zero level set that defines the physical surface. Independent evaluation of the geometric stage provides a reference point for interpreting the aerodynamic results that follow.

The experiments advance from analytically controlled shapes to progressively more complex forms. Initially, spheres and ellipsoids are used to verify the reconstruction pipeline. AhmedML [6] introduces planar regions, sharp edges, a variable rear slant, and modifications to the principal body dimensions. DrivAerML [7] adds wheels, wheel arches, underbody structures, and features distributed across multiple geometric scales, and finally the industrial Siemens geometries are considered.

The canonical experiments emphasize volumetric reconstruction since an analytical reference field is available. Beginning with AhmedML, the physical surface becomes the primary focus, and the evaluation therefore combines the Dice coefficient with the normalized 95th-percentile Hausdorff distance, HD_{95} , as defined in Section 3.3. Qualitative contour and isosurface comparisons are presented alongside these metrics to indicate where the remaining discrepancies are observed.

All results presented in this chapter concern the quality of the reconstructed SDF and its zero level set.

4.1 Evaluation of the Reconstructed Surface

Vehicle reconstructions are evaluated using the surface metrics defined in Section 3.3. The Dice coefficient quantifies solid region overlap, while the Hausdorff distance provides additional information regarding local surface displacement [74, 47]. The classical Hausdorff distance is used only in the Ahmed activation-loss sensitivity study. All final evaluations for AhmedML, DrivAerML, and Siemens employ the normalized HD_{95} metric.

The geometry evaluation groups correspond to the task-specific partitions described in Table 3.1. In this framework, the 57 AhmedML designs are used to evaluate and select model configurations trained with different values of ε_{act} , serving as a validation set rather than an independent post-selection test set. For realistic vehicle evaluations, performance is assessed on the Stage 1 test-set groups using the same protocol.

For qualitative analysis, the pointwise absolute signed distance function (SDF) difference is mapped directly onto the reconstructed surface,

$$E_{\text{abs}}(\mathbf{x}) = |\hat{\varphi}(\mathbf{x}) - \varphi(\mathbf{x})|. \quad (4.1)$$

The error field indicates where the volumetric representation differs, while the overlaid reference and reconstructed zero contours show whether those differences actually displace the represented surface.

4.2 Pipeline Validation on Canonical Geometries

The first experiments use spheres and ellipsoids because their shapes are easy to analyze mathematically. We can control how complex these shapes are without causing confusion about how their surfaces are handled. A sphere is the simplest smooth 3D shape. Its curvature is always the same, and its geometry is the same in every direction. An ellipsoid, on the other hand, has different lengths along its axes and its curvature changes. Using both shapes helps test whether Stage 1 can preserve position, scale, and shape differences before adding vehicle-specific features.

For a sphere of radius r centred at $\mathbf{c}$, the signed distance function is

$$\varphi_{\text{sphere}}(\mathbf{x}) = \|\mathbf{x} - \mathbf{c}\|_2 - r. \quad (4.2)$$

An axis-aligned ellipsoid centred at $\mathbf{c} = (c_x, c_y, c_z)$ with semi-axes (a, b, c) is defined by

$$\frac{(x - c_x)^2}{a^2} + \frac{(y - c_y)^2}{b^2} + \frac{(z - c_z)^2}{c^2} = 1. \quad (4.3)$$

For both shape families, the reference signed distance function (SDF) is generated directly from the analytical geometry. This approach avoids the

general surface-to-SDF pipeline. As a result, the canonical study serves as a controlled validation of the reconstruction network itself.

The sampled design ranges are presented in Table 4.1. All parameters are expressed in the same units as the fixed computational box.

Table 4.1: Design parameters of the canonical families and the intervals from which they are sampled. All lengths are expressed in the units of the fixed computational box.

Family	Parameter	Symbol	Range
Sphere	Radius	r	$[1.0, 2.5]$
	Centre coordinates	$\mathbf{c}$	$[-1.0, 1.0]^3$
Ellipsoid	Semi-axes	a, b, c	$[0.8, 2.5]^3$
	Centre coordinates	$\mathbf{c}$	$[-1.0, 1.0]^3$

4.2.1 Three-dimensional reconstruction of spheres and ellipsoids

Figure 4.1 compares reference and reconstructed zero-isosurfaces for two spheres and one ellipsoid from the evaluation data. The same physical coordinate system and viewpoint are used for reference and prediction.

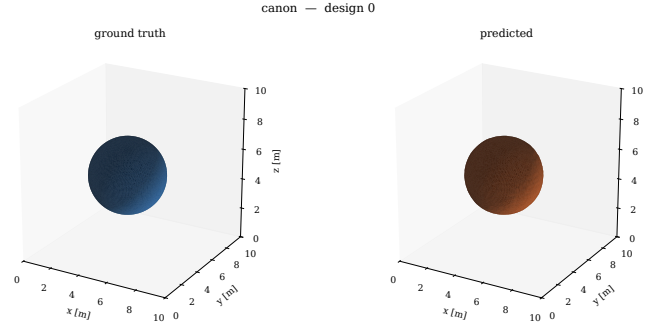
The spherical cases maintain their center position and radius, and exhibit a smooth, continuous surface. The ellipsoid presents greater complexity, as errors along any principal axis affect both its proportions and curvature distribution. Despite these challenges, the predicted ellipsoid surface preserves its elongated shape and orientation.

4.2.2 Cross-sectional reconstruction

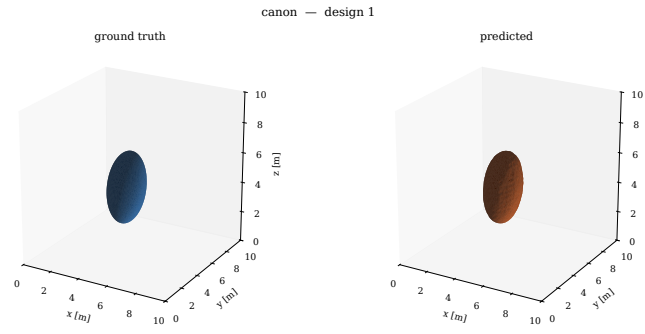
Although three-dimensional surface visualizations effectively represent overall shape, they may obscure localized discrepancies. To overcome this limitation, Figure 4.2 presents representative cross-sections from the same canonical examples. Each row shows the ground-truth signed distance function (SDF), the reconstructed SDF, and their pointwise absolute difference.

For the spheres, the network accurately reproduces the radial distance structure and the contour. The largest discrepancies occur away from the interface, at the centre of the body and in the far field, while the zero level set itself is reproduced closely. This distribution is favourable, since the geometry is determined by the zero level set alone, and errors in the distance values far from the surface do not change the recovered shape. The ellipsoid requires the model to preserve distinct scales along its principal directions. The zero contour remains close to the reference, although the absolute-error field shows a more localized difference around the narrower central region.

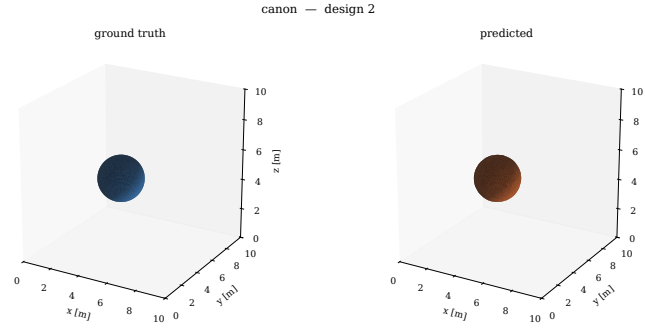
4. LEARNING GEOMETRY: A LATENT REPRESENTATION OF VEHICLE SHAPES



(a) Test set spherical design.



(b) Test set ellipsoidal design.



(c) Test set spherical design, second example.

Figure 4.1: Three-dimensional ground-truth and predicted surfaces for representative canonical geometries.

The cross-sectional analysis therefore provides additional information that is not apparent from the three-dimensional surface alone. The model reconstructs not only a plausible object but also the local distance-field structure that determines the zero level set.

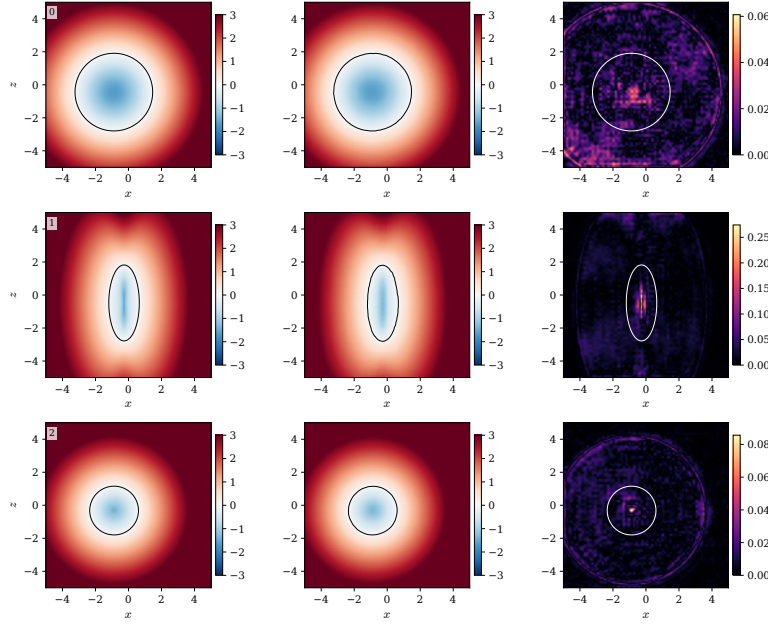


Figure 4.2: Cross-sectional reconstruction for representative canonical geometries. The first and third rows correspond to spherical designs of different sizes, while the second row corresponds to an ellipsoid. The columns show the reference field, reconstructed field, and absolute difference. The black and white contours indicate the reference and predicted zero level sets.

4.2.3 Training behavior

Figure 4.3 shows the training and validation histories for the sphere experiment. On the logarithmic scale, the optimization begins with a rapid decrease over several orders of magnitude and then enters a much slower convergence regime. The two curves remain close rather than separating progressively. The checkpoint used for evaluation is the one with minimum validation error.

4.2.4 Architecture comparison

The canonical study also compares a fully connected network (FNN) with a convolutional network (CNN) using progressively larger fractions of the available training geometries. The reported quantity is the mean absolute voxel-wise reconstruction error defined in Section 3.3.

Figure 4.4 shows two curves for each trained model: the MAE of the raw SDF prediction and the MAE after a smooth transformation is applied to that same prediction. Both curves are therefore computed from one trained network, and no activation layer is part of the architecture: the transformation is applied to the raw output after prediction. The two curves represent the same prediction in two different forms, and the vertical distance between them reflects the change of scale introduced by the transformation, not a

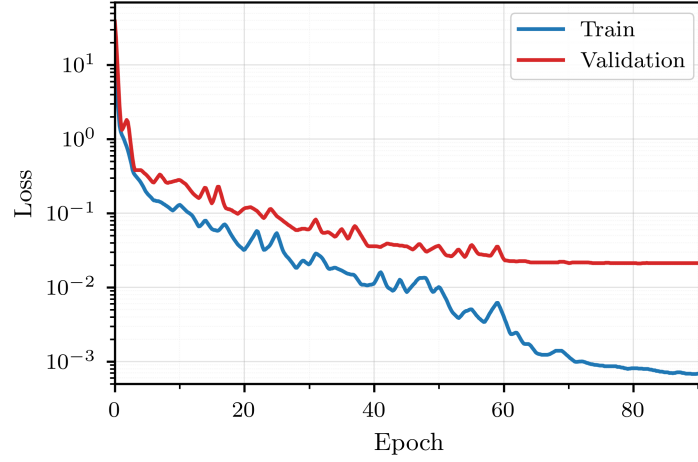


Figure 4.3: Training and validation losses for the spherical geometries reconstruction experiment on a logarithmic scale.

difference in reconstruction quality.

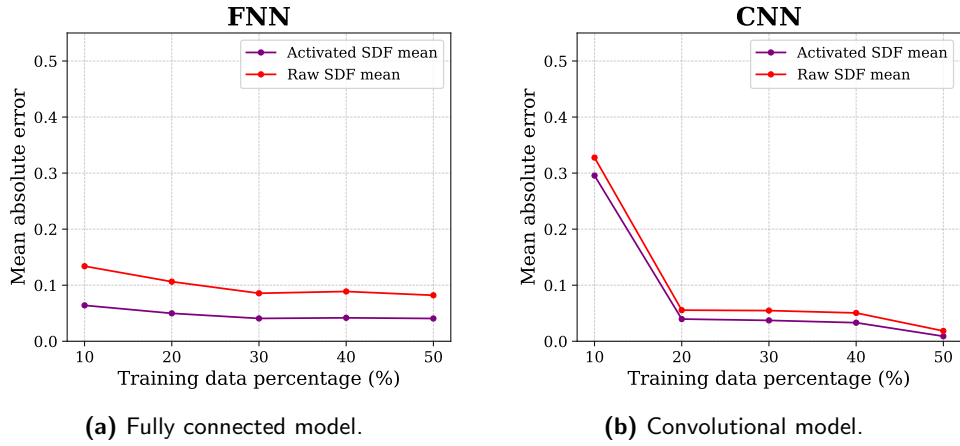


Figure 4.4: Architecture comparison on spherical geometries shape prediction.

The FNN is comparatively stable when little training data are available, but its error changes only modestly as more shapes are added. The CNN is more variable in the smallest-data regime and benefits more clearly from additional geometric examples. This behavior supports the use of volumetric convolutional encoders and decoders for the more complex families that follow.

4.2.5 Generalization across the ellipsoid family

Figure 4.5 shows how ellipsoid reconstruction changes as the amount of training data increases. Training, validation, and test errors decrease together, and the gaps between them become relatively stable as more geometries are included.

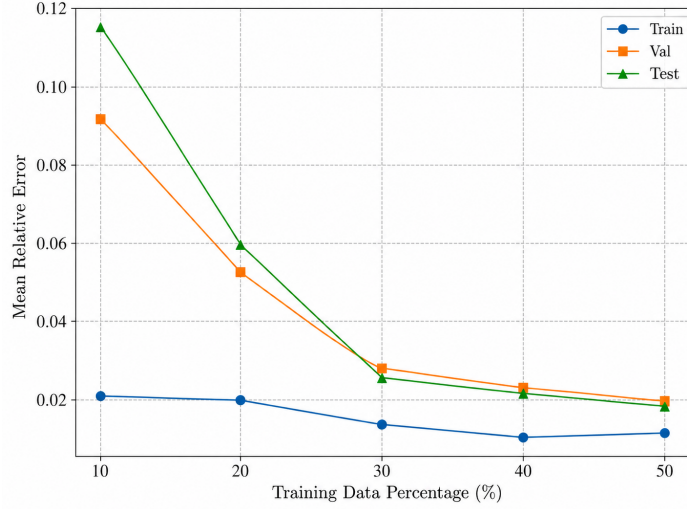


Figure 4.5: Generalization across the ellipsoid family. Training, validation, and test relative errors are reported as a function of the amount of training data.

When more than 40% of the available training geometries are included, the learning curves begin to plateau. Although adding further examples continues to reduce the test error, the improvement becomes marginal. The incremental gain is relatively small. Thus, within this canonical family, the learning curve indicates diminishing returns after a moderate portion of the sampled shape space has been represented.

4.2.6 Canonical geometry findings

The canonical experiments validate the fundamental reconstruction pipeline prior to incorporating vehicle-specific geometry. Stage 1 maintains center position, overall scale, and anisotropy for the smooth analytical shapes evaluated. Cross-sectional analyses demonstrate that the corresponding zero level sets are accurately reproduced, with no significant disconnected components observed in the reconstructions.

The architecture study indicates greater benefit from increased geometric diversity for the convolutional neural network (CNN) than for the fully connected neural network (FNN). The ellipsoid learning curve demonstrates that reconstruction performance improves as a larger portion of the sampled fam-

ily is utilized for training. These findings support selecting the convolutional Stage 1 architecture for the vehicle experiments.

Surface metrics are not reported for the canonical shapes. Their reference field is analytic and each geometry is fully determined by its centre and its radius or semi-axes, so the volumetric error and the cross-sections of Figure 4.2 already characterize the reconstruction. The failure mode that surface metrics are designed to reveal, a localized displacement at a sharp or thin feature that barely changes the overall overlap, has no counterpart in these smooth convex shapes. Beyond this stage, however, volumetric error alone is insufficient for comprehensive evaluation. Consequently, the vehicle results emphasize Dice coefficient and normalized HD_{95} metrics, supplemented by contour and surface visualizations.

4.3 Ahmed-Body Geometry

AhmedML is the first group of vehicles used to test the learned geometry representation. The topology stays the same, and the parameterization is kept simple, but the shapes have flat surfaces, sharp lower edges, different rear slants, and changes in main body size. These features make it harder to achieve accurate zero-level-set results than in standard experiments.

4.3.1 Design parameters of the family

The AhmedML family has six geometric parameters presented in Table 4.2 and illustrated in Figure 4.6 [6]. Length, height, and width control the global body proportions. The front radius modifies the leading-edge curvature, while the rear-window angle and tilted-surface length define the rear treatment that is strongly associated with the wake regimes discussed in Section 2.1.1. Ground clearance is fixed within the family.

Table 4.2: Geometric parameters of the AhmedML family and the intervals from which they are sampled [6].

Parameter	Symbol	Minimum	Maximum
Rear-window angle	φ	10°	60°
Tilted surface length	ℓ_s	150 mm	250 mm
Body length	L	800 mm	1200 mm
Body height	H	250 mm	315 mm
Body width	W	300 mm	500 mm
Front-body radius	R	80 mm	120 mm

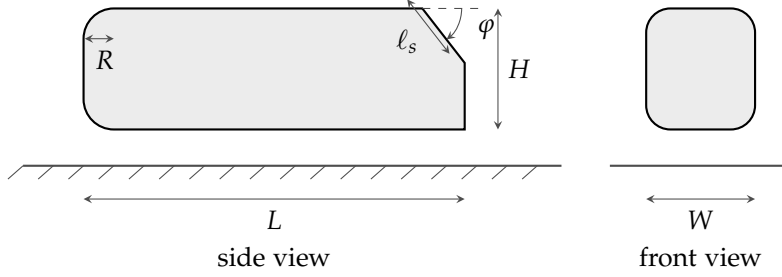


Figure 4.6: The six AhmedML design parameters visualization.

4.3.2 Effect of the activation-loss parameter

The loss introduced in Section 3.4 is examined through a sensitivity study of ε_{act} , which controls the width of the transition around the zero level set. For each tested value, Stage 1 is trained as an independent configuration while the architecture and data partition remain unchanged.

The tested values are

$$\varepsilon_{\text{act}} \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}.$$

We also trained a separate baseline model that does not include the activation term.

We ran the sensitivity study at 64^3 resolution to keep the computational cost manageable while testing a wide range of values. The goal was to compare the activation-loss settings, not to find the lowest possible Ahmed reconstruction error.

Qualitative behaviour. Figure 4.7 presents a representative evaluation geometry for all tested settings. Intermediate values of ε_{act} maintain the primary body surfaces and the rear slant. In contrast, setting $\varepsilon_{\text{act}} = 1$ results in visible surface irregularities due to an excessively wide smooth transition region.

Quantitative selection. Figure 4.8 reports the mean Dice coefficient and classical Hausdorff distance over the 57 Ahmed model-selection/evaluation designs for each separately trained configuration. Dice is computed from the overlap of the reference and reconstructed solid regions defined by $\varphi < 0$ and $\hat{\varphi} < 0$.

The raw-loss baseline achieves a good average Dice score of approximately 0.975, establishing that an accurate global volume can be reconstructed without an activation penalty. Instead, the activation term serves primarily to refine local surface placement. Dice scores improve for intermediate values and reach about 0.994 at $\varepsilon_{\text{act}} = 10^{-3}$, with 10^{-4} and 10^{-2} giving similar

4. LEARNING GEOMETRY: A LATENT REPRESENTATION OF VEHICLE SHAPES

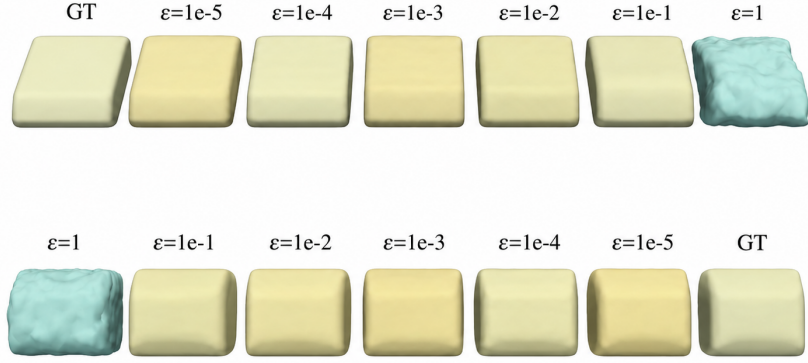


Figure 4.7: Reconstruction of one Ahmed-body evaluation design under each activation-loss setting, together with the ground-truth (GT) geometry.

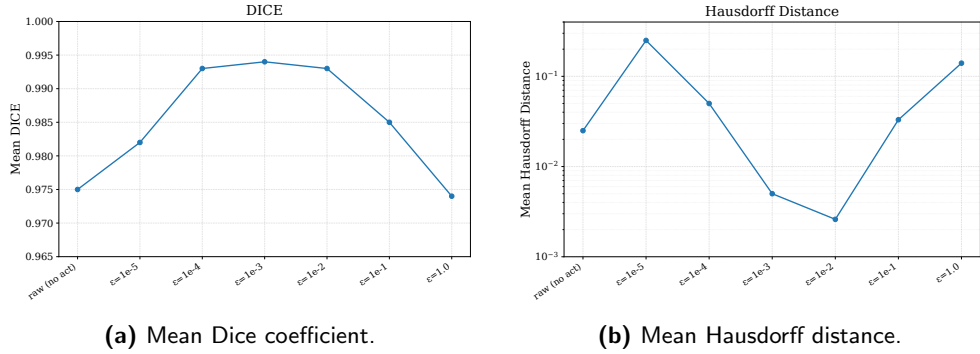


Figure 4.8: Sensitivity of the Ahmed-body reconstruction to the activation-loss parameter ϵ_{act} . Each value corresponds to a separately trained Stage 1 configuration. The metrics are averaged over the 57 Ahmed model-selection/evaluation designs.

results. Performance drops at 10^{-1} and 1. The Hausdorff measure ranks the results a bit differently. Its lowest value is near $\epsilon_{\text{act}} = 10^{-2}$, and 10^{-3} is also low. Both very small and very large transition widths lead to higher local surface errors.

The difference between these two metrics is important. For example, $\epsilon_{\text{act}} = 10^{-5}$ gives a surface that looks plausible and a Dice score close to the best settings, but its Hausdorff distance is one of the highest. This shows that high overlap can still happen alongside a small area with large displacement.

A surface-distance metric is needed to reveal this kind of issue.

Retained value. Analysis of both metrics across the 57 Ahmed model-selection and evaluation designs indicates that ε_{act} values between 10^{-3} and 10^{-2} are most favorable. The Dice score reaches its maximum at 10^{-3} , whereas the Hausdorff distance is minimized at 10^{-2} . The latter value is therefore retained as the preferred width of the interface term, because it reduces the largest local surface discrepancy while maintaining the Dice score near its maximum.

The retained setting governs only how the interface is supervised during training. No activation is applied to the network output in any configuration: the smooth transformation of Equation (3.11) acts inside the loss alone, and the decoder always emits the raw signed distance field. It is this raw decoder output that is passed to Stage 2 and that is displayed in every reconstruction panel of this chapter.

4.3.3 Surface accuracy over the Ahmed-body model-selection/evaluation set

Figure 4.9 shows Dice and normalized HD_{95} scores for the 57 Ahmed model-selection/evaluation designs using the retained setting.

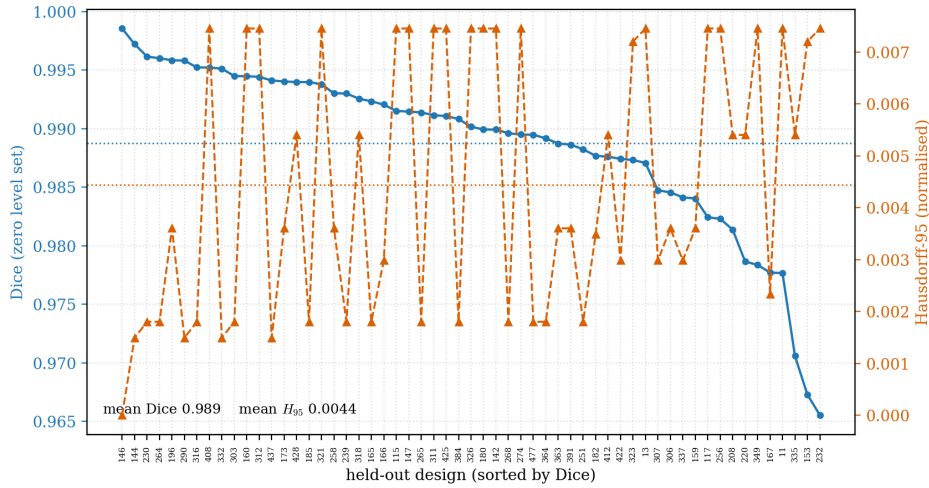


Figure 4.9: Dice coefficient and normalized 95th-percentile Hausdorff distance, HD_{95} , over the 57 Ahmed model-selection/evaluation designs, sorted by Dice. The mean values are 0.989 and 0.0044.

Most designs lie close to or above 0.99 Dice, indicating that the zero level set overlaps the reference body over nearly the entire geometry. The tail of the sorted distribution contains a smaller number of more difficult cases.

4.3.4 Longitudinal reconstruction

Figure 4.10 shows designs 146, 153, and 337 from the same 57-design model-selection/evaluation group. They were selected to illustrate different positions in the Dice distribution rather than to define a separate subset.

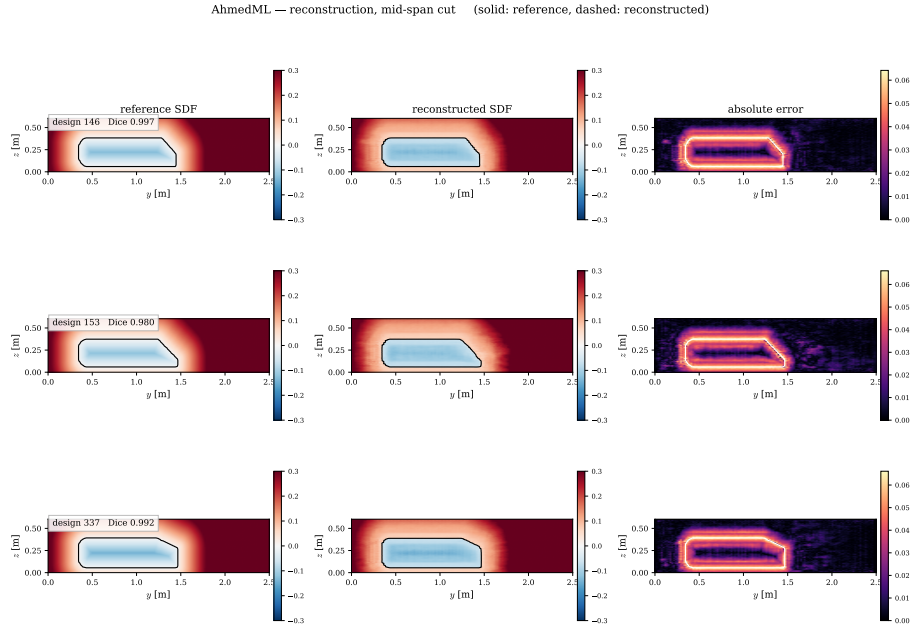


Figure 4.10: Longitudinal reconstruction for Ahmed evaluation designs 146, 153, and 337. Each row shows the reference SDF, reconstructed SDF, and absolute error with reference and predicted zero contours. The adopted sign convention assigns negative values to the solid interior, zero to the body surface, and positive values to the exterior fluid region.

The reconstructed zero contours stay close to the reference along the roof, sides, and base. Most differences appear where the surface changes direction sharply, especially near the rear, lower corners, and short vertical edges. Design 146 matches the reference contours most closely. Of the three designs, design 153 has a bigger difference near the rear slant and lower body, while design 337 falls in between.

Small differences also appear away from the surface. These errors change the SDF values but matter less for geometry if the zero contour stays the same.

Effect of the grid resolution. The residuals in Figure 4.10 are concentrated where the representation is coarsest. The computational box is padded downstream to contain the wake, so on the 128^3 grid a cell measures approximately 19.5 mm in the streamwise direction against 3.9 mm spanwise and 4.7 mm vertically. The Ahmed body therefore spans only about 55 cells along its length, and a rear slant between 150 and 250 mm long is resolved by fewer

than fifteen streamwise cells. The mean HD_{95} of about 11.5 mm is smaller than one streamwise cell, so on this family the reconstruction is already close to the resolution limit of the grid. The limitation becomes more pronounced on DrivAerML (Section 4.4), where the vehicle is larger relative to its box.

Refining the grid would lower this limit, at a cost that grows with the number of cells: doubling the resolution to 256^3 multiplies the number of voxels, and approximately the memory of the three-dimensional activations, by eight, so the batch size would have to fall accordingly. A cheaper alternative is anisotropic, refining only the streamwise direction or reducing the downstream padding at the cost of truncating the wake, since this targets the coarsest spacing directly. No Stage 1 model was trained at a higher resolution, so slow convergence and the architectural limit are not separated experimentally here; the quantization of the DrivAerML surface error at the cell size, discussed below, nonetheless indicates that for the geometry the grid rather than the optimization is the binding constraint.

4.3.5 Three-dimensional reconstruction

Figure 4.11 compares the complete reference and reconstructed surfaces for design 146.

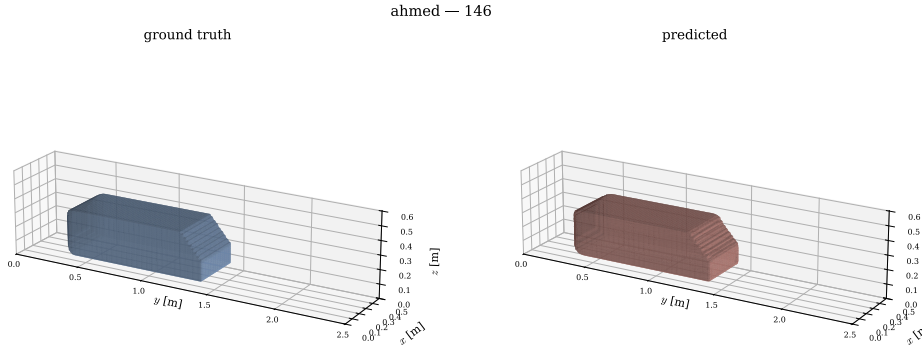


Figure 4.11: Reference and reconstructed zero-isosurfaces for Ahmed geometry-evaluation design 146. The reconstruction retains the complete three-dimensional body, its global proportions, ground clearance, and rear treatment.

The 3D view shows that the agreement seen on the longitudinal plane applies to the whole model, not just one section. The length, width, height, ground clearance, and rear geometry all match up well, and there are no major disconnected parts in the reconstructed body. Any remaining differences are small and local, not problems with the overall Ahmed Body shape.

4.4 Realistic Vehicle Geometry: DrivAer

DrivAerML has a more complex geometry. The vehicle surface now has a realistic bonnet, windscreen, roof, rear body, wheels, wheel arches, underbody structures, and small details that can be as thin as the voxel spacing.

DrivAerML uses 16 morphing parameters, compared to the six in AhmedML, to generate new shapes from a baseline notchback model with static wheels, sealed cooling inlets, and a detailed underbody [7]. Most parameters in Table 4.3 and Figure 4.12 are control-point displacements measured from the baseline, not absolute sizes. The only exception is vehicle pitch, which is a rotation. The 500 variants are created using a modified extensible lattice sequence to evenly cover the allowed parameter space.

The Stage 1 transfer protocol is defined in Chapter 3. This section reports the 9 DrivAerML test-set geometry-evaluation cases evaluated under that protocol, and all Dice scores, HD_{95} values, longitudinal panels, and three-dimensional examples correspond to this group.

The parameter-driven study described later in this section constitutes a distinct DrivAerML-only fine-tuning experiment, evaluated using a dedicated 10-design test group. This test group is separate from the 9-design geometry-evaluation group employed in the mixed DrivAerML-Siemens volumetric Stage 1 experiment.

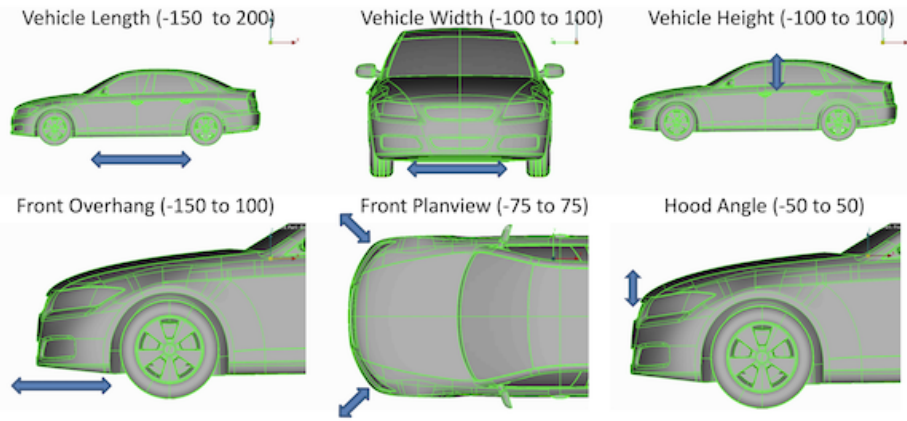


Figure 4.12: DrivAerML geometric design parameters. Adapted from the DrivAerML dataset website [16] and the dataset described in [7].

4.4.1 Surface metrics

Figure 4.13 reports Dice and normalized HD_{95} for all 9 DrivAerML Stage 1 evaluation cases.

Table 4.3: Design parameters of the DrivAerML family and the intervals from which they are sampled, expressed as changes relative to the baseline geometry [7]. All entries are control-point displacements of the morphing boxes, in millimetres, except the vehicle pitch, which is a rotation in degrees.

Parameter	Minimum	Maximum
Vehicle length	−150 mm	+200 mm
Vehicle width	−100 mm	+100 mm
Vehicle height	−100 mm	+100 mm
Front overhang	−150 mm	+100 mm
Front planview	−75 mm	+75 mm
Hood angle	−50 mm	+50 mm
Approach angle	−40 mm	+30 mm
Windscreen angle	−150 mm	+150 mm
Greenhouse tapering	−100 mm	+100 mm
Backlight angle	−100 mm	+200 mm
Decklid height	−50 mm	+50 mm
Rear-end tapering	−90 mm	+70 mm
Rear overhang	−150 mm	+100 mm
Rear diffuser angle	−50 mm	+50 mm
Vehicle ride height	−50 mm	+50 mm
Vehicle pitch, nose up positive	−1°	+1°

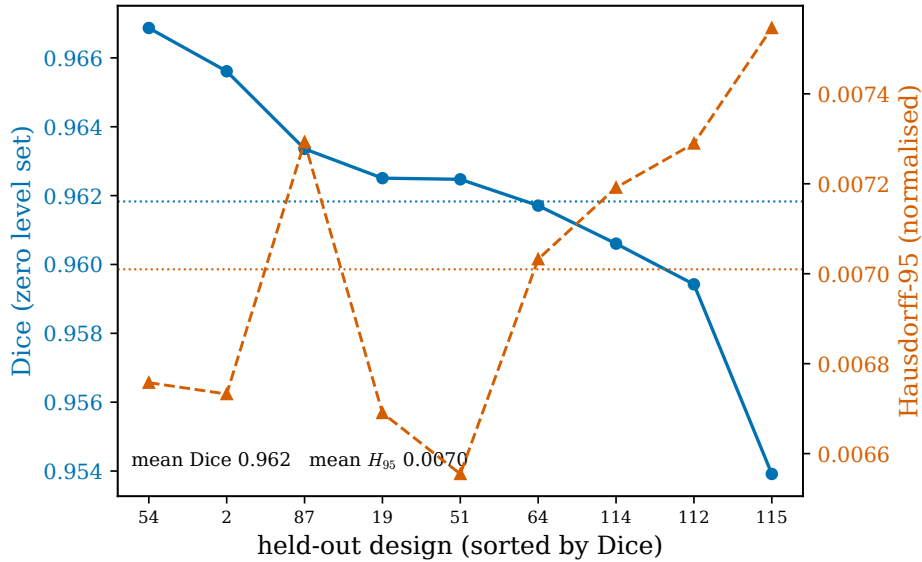


Figure 4.13: Dice coefficient and normalized 95th-percentile Hausdorff distance, HD_{95} , for the 9 DrivAerML Stage 1 evaluation cases, sorted by Dice. The mean Dice coefficient is 0.962, and the mean normalized HD_{95} is 0.0070. The latter corresponds to approximately 78 mm before normalization by the computational-box diagonal.

The Dice values remain within a narrow range, while HD_{95} exhibits greater variability across cases in terms of local surface displacement.

Before normalization by the computational-box diagonal, the mean HD_{95} corresponds to approximately 78 mm.

Designs 2, 19, and 51 are selected for longitudinal analysis in the following sections. Additionally, design 2 is presented as a complete three-dimensional surface.

4.4.2 Longitudinal reconstruction

Figure 4.14 presents the three selected DrivAerML evaluation cases.

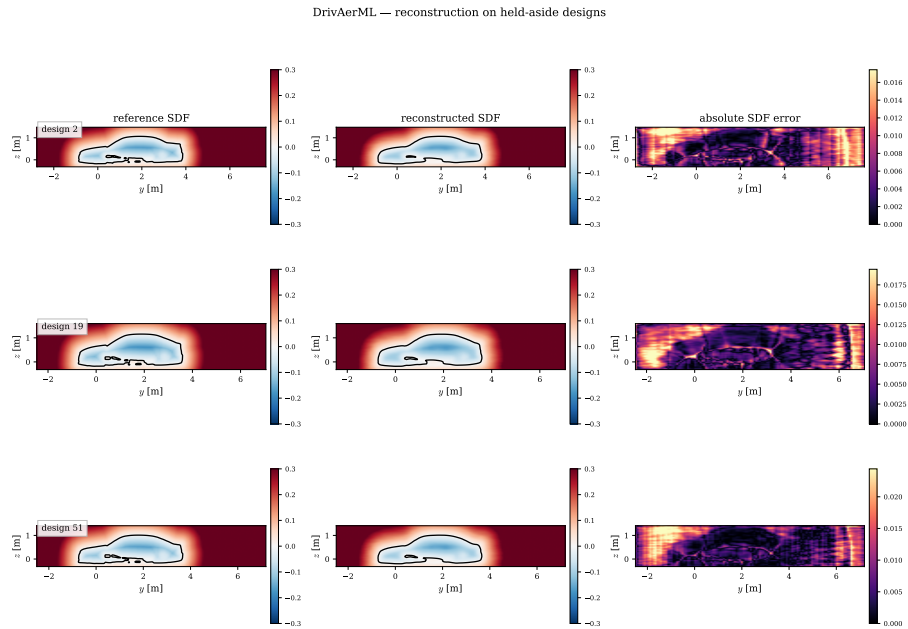


Figure 4.14: Longitudinal reconstruction for DrivAerML Stage 1 evaluation designs 2, 19, and 51. Each row shows the reference field, reconstructed field, and absolute error with zero-contour overlays. The global vehicle envelope is preserved.

In each instance, the reconstructed contour aligns with the bonnet, wind-screen, roof, rear profile, and overall vehicle position. The most significant discrepancies occur in the lower region of the vehicle, where wheels, underbody components, and narrow gaps introduce multiple interfaces within a limited number of voxels.

Two mechanisms contribute to these residuals. The first is representational: geometric features occupying only one or two voxels cannot retain fine detail on the selected grid. On the 128^3 grid a DrivAerML cell measures approximately 80 mm in the streamwise direction and 33 and 14 mm in the

other two, and the mean HD_{95} of about 78 mm corresponds to a single cell in the coarsest direction, so the surface error is quantized at the cell size. The second is learned compression: broad, smooth surfaces are more readily preserved by the encoder-decoder than thin or closely spaced structures [120].

4.4.3 Three-dimensional reconstruction of a DrivAer example

Figure 4.15 shows the complete zero-isosurface comparison for design 2.

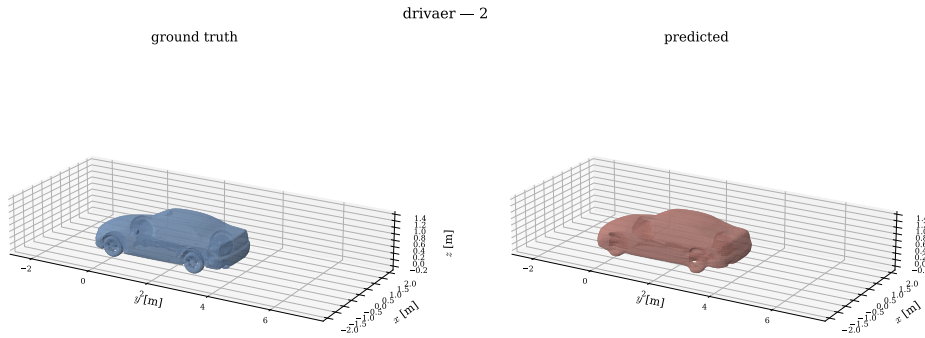


Figure 4.15: Reference and reconstructed zero-isosurfaces for DrivAerML Stage 1 evaluation design 2.

The reconstruction retains the overall length, width, bonnet, roof, rear body, and stance. The remaining disagreement concerns local fidelity around fine components rather than the continuity or global proportions of the vehicle.

4.4.4 Parameter input reconstruction of DrivAerML

In addition to the mixed DrivAerML–Siemens configuration described previously, a separate DrivAerML-only Stage 1 fine-tuning experiment was conducted using the 16 morphing parameters specific to DrivAerML. This experiment also employed a transfer-learning approach rather than training the model from scratch. The Stage 1 model was initialized with the previously learned geometry representation and subsequently fine-tuned on DrivAerML using its morphing parameters. In this setup, the parameters were provided to the parameter-driven Stage 1 branch instead of supplying the geometry as an input signed distance function (SDF). The parameter vector was mapped to the learned latent representation and decoded into the reconstructed SDF. The reconstructed field was then resampled to the Stage 2 resolution and forwarded to the end-to-end Stage 2 U-Net, to test the case of DrivAerML with its parameters as input.

4. LEARNING GEOMETRY: A LATENT REPRESENTATION OF VEHICLE SHAPES

Figure 4.16 presents the Dice coefficient and normalized HD_{95} for the ten DrivAerML test designs evaluated using this parameter-driven configuration. The mean Dice coefficient is 0.928, the mean normalized HD_{95} is 0.0072, and the mean SDF mean absolute error is 9.9×10^{-3} across these designs.

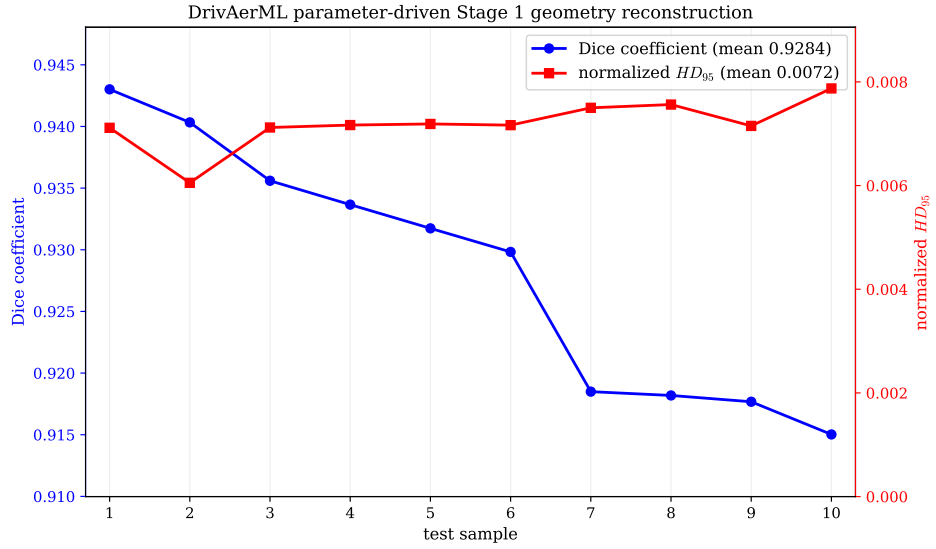


Figure 4.16: Dice coefficient and normalized 95th-percentile Hausdorff distance, HD_{95} , for the 10 DrivAerML test designs evaluated with the parameter-driven Stage 1 branch. The mean Dice coefficient is 0.928, and the mean normalized HD_{95} is 0.0072.

Figure 4.17 shows design 114 as a qualitative example of the same parameter-driven reconstruction. The reference SDF and the parameter-driven reconstruction preserve the global vehicle profile, while the absolute-error field shows the remaining local differences. The black zero-level contours provide a direct comparison between the reference and reconstructed physical surfaces.

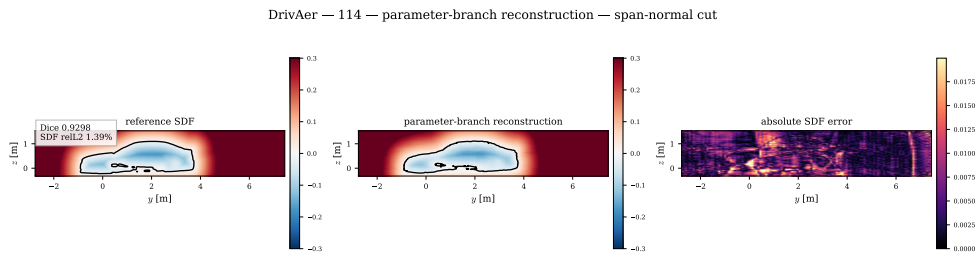


Figure 4.17: Parameter-driven Stage 1 reconstruction for DrivAerML design 114. The panels show the reference SDF, the SDF reconstructed from the DrivAerML morphing parameters, and the corresponding absolute SDF error on a span-normal cut. The black contours indicate the zero level sets defining the reference and reconstructed vehicle surfaces.

4.4.5 DrivAer findings

The DrivAerML experiment demonstrates the principal limitation of the current voxelized representation more distinctly than the canonical and Ahmed studies. Large and intermediate geometric scales are reconstructed consistently, while the largest residuals are observed around thin, sharply curved, or closely spaced features.

The additional DrivAerML evaluation using the parameter branch shows a less accurate reconstruction than the experiment using the volumetric SDF as an input, but it still presents an acceptable range of values for the Dice coefficient and the normalized HD_{95} . The two configurations are evaluated on different design groups, so the comparison indicates the direction of the difference rather than a paired measurement.

4.5 Industrial Siemens Geometries

The Siemens configurations represent the most application-oriented geometry experiment presented in this chapter. These configurations differ from the open-source datasets in styling, proportions, local components, and data provenance, and there is no available low-dimensional parameterization associated with them. The Stage 1 transfer partition is defined in Chapter 3. The metrics and qualitative examples presented here refer to the five Siemens geometry-evaluation configurations held aside from the Siemens-only Stage 1 fine-tune described in Chapter 3.

4.5.1 Surface metrics

Figure 4.18 reports Dice and normalized HD_{95} for the five Siemens evaluation geometries.

The narrow range of Dice values suggests that the reconstruction of the main industrial vehicle envelope is consistent across all evaluated configurations.

4.5.2 Longitudinal reconstruction

Figure 4.19 shows Design_5_2, Design_6_2, and Design_7_2.

All three reconstructions maintain the vehicle position, roof line, frontal region, rear profile, and ground clearance. The highest level of agreement is observed along the large, smooth surfaces. The most apparent discrepancies are found around the front treatment, lower body, and small components beneath the principal envelope, where geometric detail approaches the resolution limit of the sampled SDFs.

4. LEARNING GEOMETRY: A LATENT REPRESENTATION OF VEHICLE SHAPES

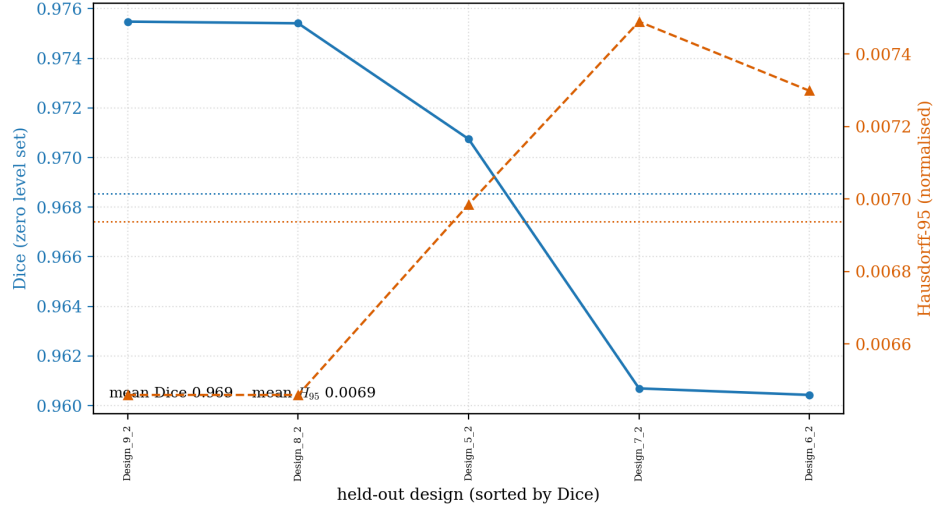


Figure 4.18: Dice coefficient and normalized 95th-percentile Hausdorff distance, HD_{95} , for the five test set Siemens designs, sorted by Dice. The mean values are 0.969 and 0.0069, respectively.

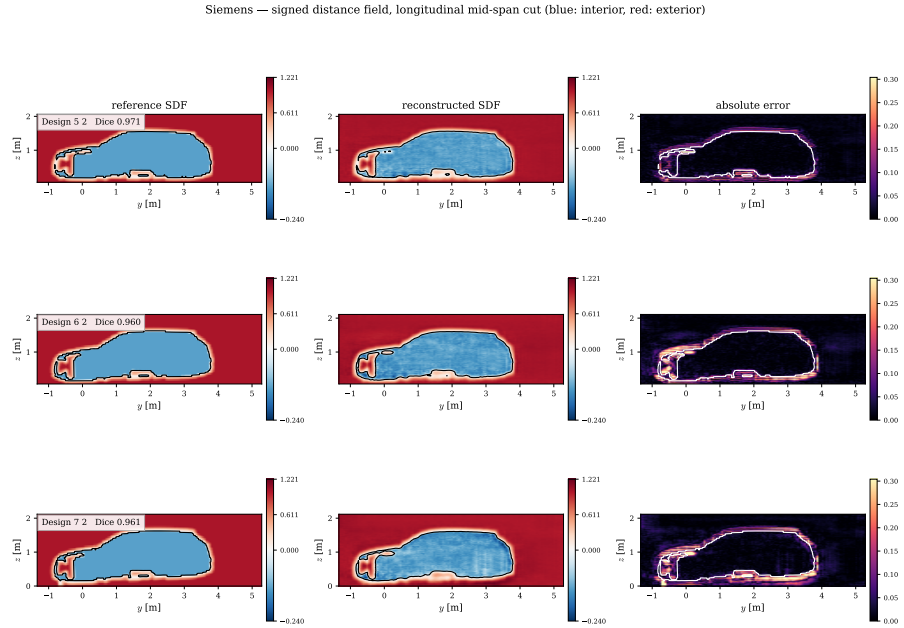


Figure 4.19: Longitudinal reconstruction for Siemens configurations Design_5_2, Design_6_2, and Design_7_2. Each row shows the reference field, reconstructed field, and absolute error with zero-contour overlays. The principal industrial vehicle profile is preserved, while the largest localized residuals occur near front structures, the lower body, and small components.

4.5.3 Three-dimensional reconstruction

Figure 4.20 compares the complete reference and reconstructed surfaces for Design_6_2.

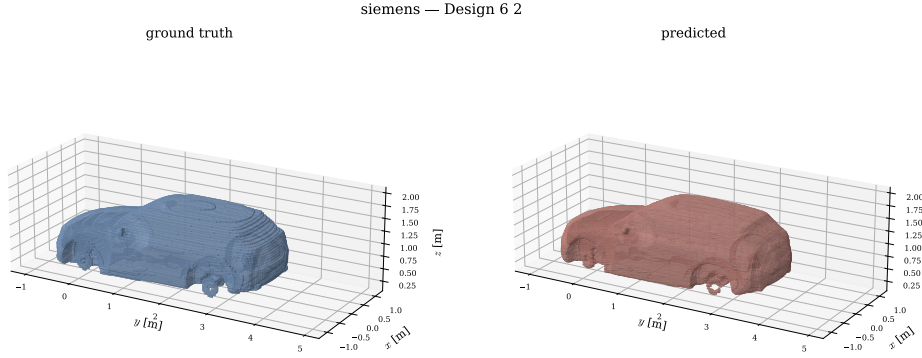


Figure 4.20: Reference and reconstructed zero-isosurfaces for Siemens configuration Design_6_2. The reconstruction preserves the complete industrial vehicle envelope and its principal exterior features.

The predicted surface maintains the overall length, width, height, front treatment, roof, and rear geometry. No significant disconnected components are observed in the inspected reconstruction. Similar to the DrivAerML case, the remaining errors are confined to detailed components and do not impact the vehicle as a whole.

4.5.4 Siemens findings

The Siemens results show that the adapted Stage 1 model preserves the primary industrial vehicle envelope, with the remaining discrepancies concentrated around smaller and more detailed structures.

4.6 Chapter Summary

This chapter evaluated Stage 1 independently before considering aerodynamic prediction. The canonical sphere and ellipsoid experiments verified that the reconstruction pipeline preserves position, scale, and anisotropy for smooth analytical shapes and supported the use of the convolutional architecture for the vehicle studies.

AhmedML introduced sharp edges and adjustable rear geometry.

The DrivAerML and Siemens experiments then tested the same geometric representation on more realistic vehicle surfaces. Across these cases, the

global vehicle envelope and the main exterior features are reconstructed consistently, while the largest remaining discrepancies occur around sharp, thin, closely spaced, or otherwise under-resolved components. The additional DrivAerML-only parameter-driven evaluation confirms that the same Stage 1 framework can reconstruct realistic vehicle SDFs directly from the available morphing parameters. These reconstructed SDFs provide the geometric inputs for the end-to-end aerodynamic experiments in Chapter 6. Chapter 5 instead evaluates Stage 2 independently using reference SDFs.

Predicting Flow Fields and Drag

This chapter evaluates Stage 2 independently using the reference signed distance field as geometric input. The objective is to assess the SDF-to-flow mapping before considering the complete Stage 1–Stage 2 composition in Chapter 6.

Stage 2 predicts U_x , U_y , U_z , and P on the 128^3 Cartesian grid. The velocity magnitude $|\mathbf{U}|$ is calculated from the three predicted velocity components, while drag is evaluated separately using the scalar regressor defined in Section 3.4. Field errors follow the evaluation procedure of Section 3.3 and are reported as means over the test set designs together with per-design curves.

The chapter follows the same progression in geometric complexity used for the geometry study: AhmedML [6], DrivAerML [7], and the industrial Siemens configurations.

For the realistic-vehicle direct experiment, Stage 2 is fine-tuned jointly on 34 DrivAerML and 10 Siemens flow cases, for a total of 44 fine-tuning cases. Evaluation is then reported separately on the 10-design DrivAerML direct test set and the 10 test set Siemens configurations.

5.1 Volumetric field prediction on the Ahmed body

The AhmedML direct-flow experiment evaluates the retained Stage 2 model on the 45 test set flow designs defined in Chapter 3. The family spans the six geometric variables in Table 4.2 and includes markedly different rear-flow regimes [1, 63, 38, 6], making it the first benchmark for the volumetric aerodynamic predictor.

Per-field accuracy. Table 5.1 shows a clear pattern: streamwise velocity and velocity magnitude are the most accurate, pressure falls in the middle, and

5. PREDICTING FLOW FIELDS AND DRAG

Table 5.1: Direct Stage 2 field-prediction results. The reference SDF is supplied to the U-Net, and the values are mean relative L^2 errors over the test set flow designs of each dataset.

Dataset	n	U_x	U_y	U_z	P	$ \mathbf{U} $
AhmedML	45	21.77%	3.76%	16.55%	13.34%	3.81%
DrivAerML	10	28.83%	4.43%	31.92%	18.44%	4.60%
Siemens	10	33.0%	10.4%	36.0%	24.2%	10.3%

the transverse components have higher relative errors. This pattern partly reflects the physics. The streamwise field contains broad, coherent structures such as the freestream, velocity deficit, and recirculating wake, whereas U_x and U_z are influenced more strongly by secondary motions, shear layers, downwash, and vortical structures.

The way relative error is defined also highlights this difference. Because the reference norms for the transverse components are usually smaller than for the streamwise field, the same absolute difference leads to a higher relative percentage. So, the field hierarchy should be considered along with the scale of each reference value, not just as a direct ranking of absolute errors.

Figure 5.1 shows the distribution for each design that makes up the dataset-level averages. The curve indicates that the average is not influenced by just one geometry, and the same general ranking of field difficulty appears in the test set designs.

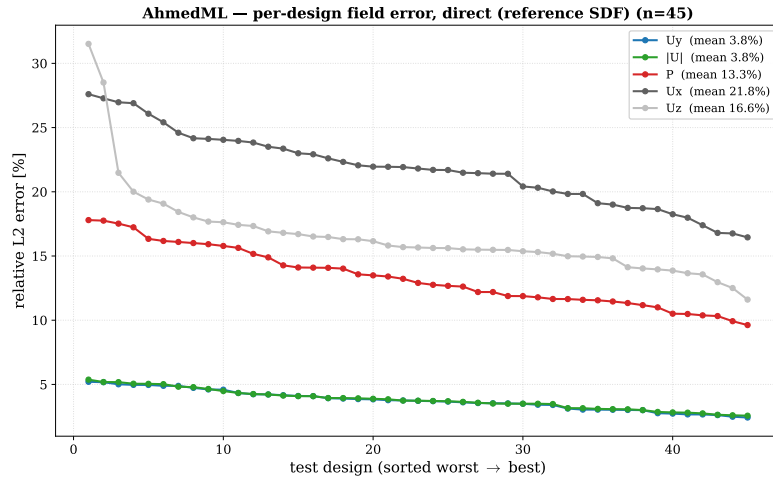


Figure 5.1: Per-design relative L^2 errors for the AhmedML direct Stage 2 evaluation. The reference SDF is used as input. The figure complements the 45-design mean values of Table 5.1 by showing the variation across individual test set designs.

Detailed inspection of a test set design. Figure 5.2 shows a test set Ahmed geometry longitudinal slice.

The prediction matches the main structures seen in the reference fields. The streamwise component shows the wake deficit and recirculation behind the body. The vertical field follows the upwash and downwash pattern linked to the front and rear slant. The pressure field keeps the frontal stagnation area, upper-surface suction, rear pressure changes, and the base-pressure structure. The biggest differences are still found close to the body, along sharp shear layers, and in the near wake.

These localized residuals are likely to cause significant pointwise errors. Even a slight shift in a steep shear layer can create a noticeable absolute difference, even if the overall wake shape stays similar to the reference. The same idea holds true for thin pressure features near the body.

Spatial localization of the error. For the test set Ahmed designs, the freestream and large attached-flow regions are matched with only small differences. Bigger errors mostly appear near the rear slant, wake shear layers, underbody, and recirculation area. These spots are where sharp gradients and changes in separation make the flow especially sensitive to the shape. The direct Ahmed result gives a helpful baseline: Stage 2 captures the main mean-flow pattern from the reference SDF, while the remaining errors are focused in the most local and rapidly changing areas.

5. PREDICTING FLOW FIELDS AND DRAG

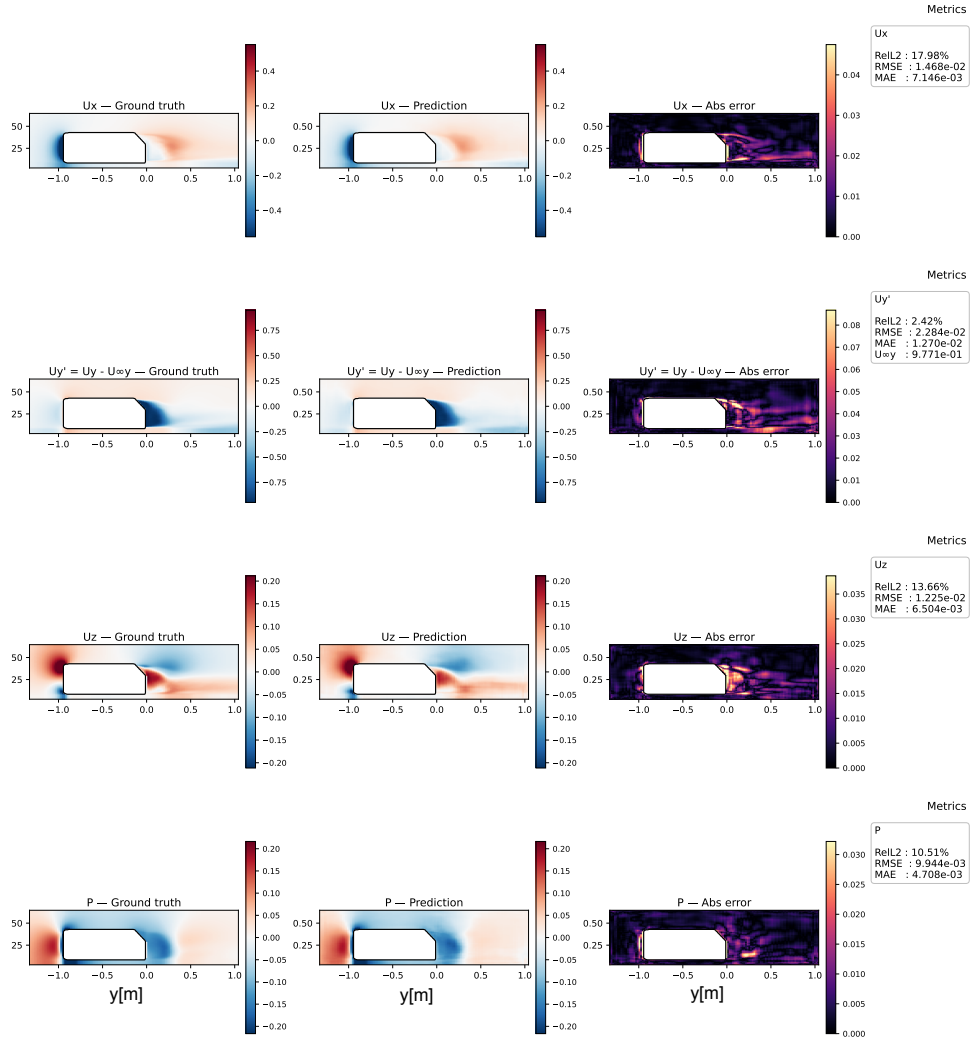


Figure 5.2: Direct Stage 2 prediction for a selected test set Ahmed design longitudinal slice. Rows: spanwise velocity, streamwise velocity, vertical velocity, and pressure. The streamwise field is displayed as $u_y - U_\infty$ to make the wake deficit visible; the reported relative error is computed on the full field. Columns show the input distance field, reference, prediction, and absolute error.

5.2 Volumetric field prediction on DrivAerML

DrivAerML presents a tougher Stage 2 challenge because both the geometric detail and aerodynamic interactions are more complex.

The direct realistic-vehicle model follows the 44-case fine-tuning protocol defined above and in Table 3.1. The results in this section refer only to the 10-design DrivAerML direct test set.

The full-vehicle geometry adds wheel wakes, underbody flow, mirrors, A-pillars, and a rear wake shaped by many parts of the car [42, 7]. The 128^3

Cartesian grid also means some small features are near the grid’s resolution limit [120]. As a result, the DrivAerML [7] experiment tests the flow model with a mix of geometric, spatial-resolution, and aerodynamic challenges.

Figures 5.3 and 5.4 show typical qualitative examples of the direct SDF-to-flow mapping. These figures help illustrate spatial behavior, not the average across the dataset. In both examples, the prediction matches the main streamwise acceleration and wake deficit, while the largest differences appear near the rear shear layer, underbody, wheels, and other specific areas. The pressure results capture the main frontal and rear loading patterns, but there are still some small differences in more complex geometric regions.

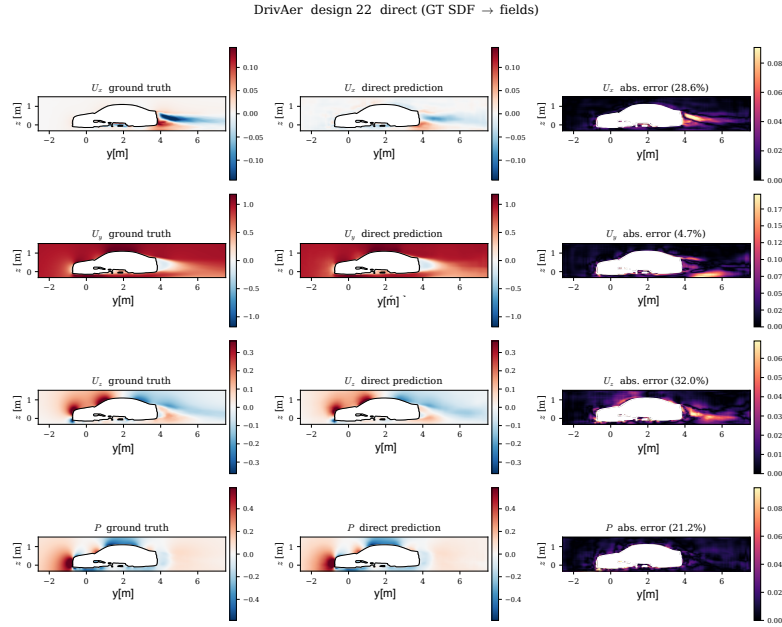


Figure 5.3: Representative DrivAerML direct-flow visualization on a longitudinal slice near the symmetry plane. Rows show the streamwise, spanwise, and vertical velocity components followed by pressure; columns show reference, prediction from the reference SDF, and absolute error. The visualization complements the quantitative test set evaluation reported in the text.

Results. The mean values in Table 5.1 show that the streamwise component remains relatively accurate even as the geometry becomes more complex. Larger errors appear in pressure and the transverse velocity components. The strongest residuals are concentrated in aerodynamically active regions such as the wheels, underbody, mirrors, near-body shear layers, and near wake.

This observation does not mean the error comes from just one cause. Geometry complexity, grid resolution, transfer from AhmedML, and the more

5. PREDICTING FLOW FIELDS AND DRAG

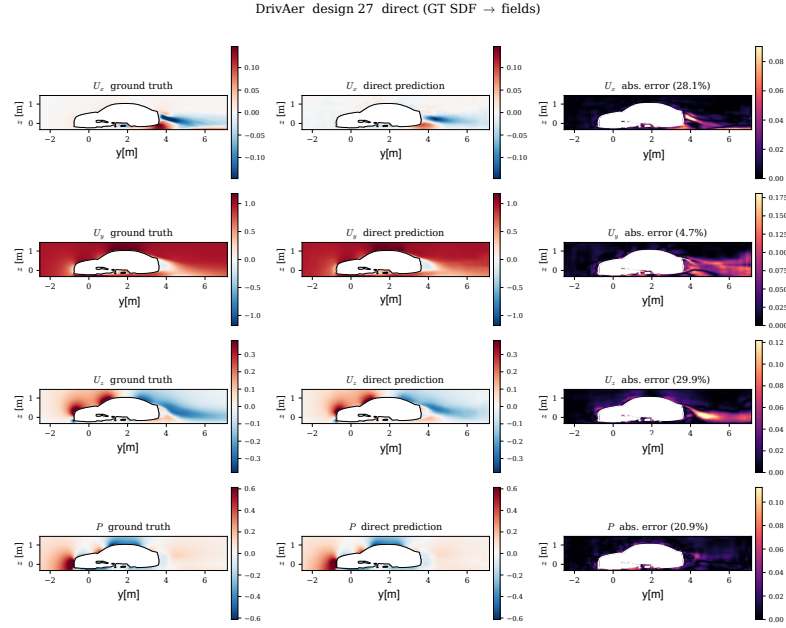


Figure 5.4: Second representative DrivAerML direct-flow visualization, using the same layout as Figure 5.3. The largest residuals are concentrated around the wake onset, rear shear layer, and underbody.

complex full-vehicle flow all change together. The experiment measures their combined effect on the direct surrogate instead of isolating each factor.

Figure 5.5 shows how the relative errors are distributed across the 10 test set designs. The curve, along with the mean values, highlights the variation within the evaluation group.

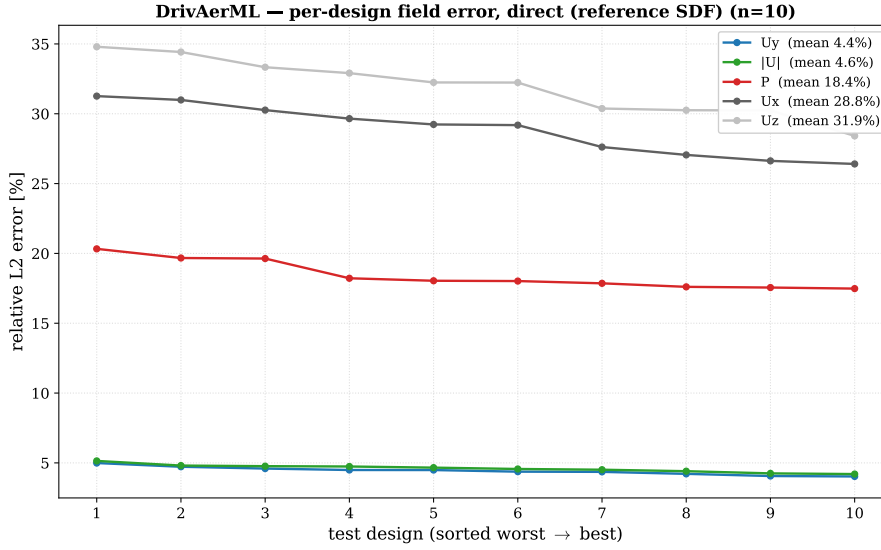


Figure 5.5: Per-design relative L^2 errors for the DrivAerML direct Stage 2 evaluation. The curve shows the variation across the 10 test set flow designs and complements the mean values reported in Table 5.1.

5.3 Volumetric field prediction on Siemens geometries

The Siemens experiment tests how well the direct realistic-vehicle model transfers to industrial data. The reference solutions use steady RANS fields, and this section reports the 10 test set Siemens flow configurations defined by the direct Stage 2 protocol in Chapter 3.

Industrial domain shift. Moving from the open datasets to Siemens changes several parts of the problem at once. The vehicle shapes differ in style, size, and detail. The CFD reference also shifts from time-averaged, scale-resolving simulations to steady RANS. In addition, the adaptation set is much smaller than the open-data training sets. Because of this, the Siemens experiment examines how geometric changes, changes in the simulation model, and limited industrial data affect the results together.

Results. Compared with AhmedML [6] and DrivAerML [7], the Siemens errors in Table 5.1 are higher, especially for streamwise velocity and pressure. The result reflects transfer to a different industrial geometry family and CFD source using a limited amount of industrial fine-tuning data.

Figure 5.7 shows the distribution for each design, which adds important context. Accuracy varies more across the industrial configurations than in the open-data evaluations. This variation matches the combined effects of

5. PREDICTING FLOW FIELDS AND DRAG

geometric, numerical, and data-volume differences discussed earlier, though this experiment does not separate their individual impacts.

Figure 5.6 shows a spatial comparison for Design_5_2, one of the ten test configurations used to compute the errors in Table 5.1 and Figure 5.7. This example highlights the type and location of prediction discrepancies. This configuration differs from the one shown for the end-to-end pipeline in Figure 6.11, so the two figures do not form a paired comparison; Section 6.3 discusses the difference.

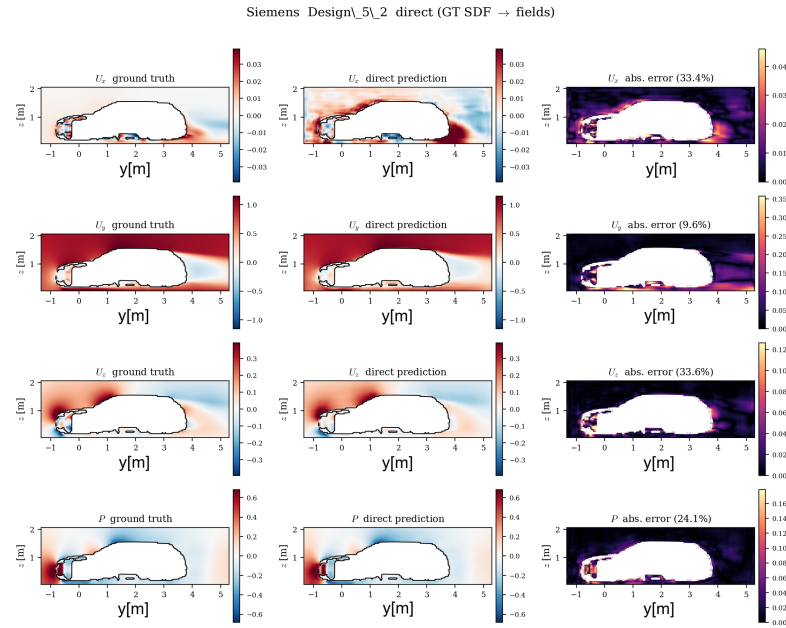


Figure 5.6: Representative Siemens direct-flow visualization for Design_5_2 on a longitudinal mid-span slice. Rows show the three velocity components and pressure; columns show the reference field, prediction from the reference SDF, and absolute error. This configuration is not among the ten test configurations of Table 5.1, and it differs from the configuration shown for the end-to-end pipeline in Figure 6.11; the two figures should therefore not be read as a paired comparison.

Our current industrial results show that the surrogate works well as a quick tool for screening and inspecting flow. These predictions help us examine large-scale flow patterns and identify designs that might need a more detailed CFD analysis [91, 26]. To confirm how well the surrogate ranks designs, we would need to test it on a bigger and more varied set of industrial cases.

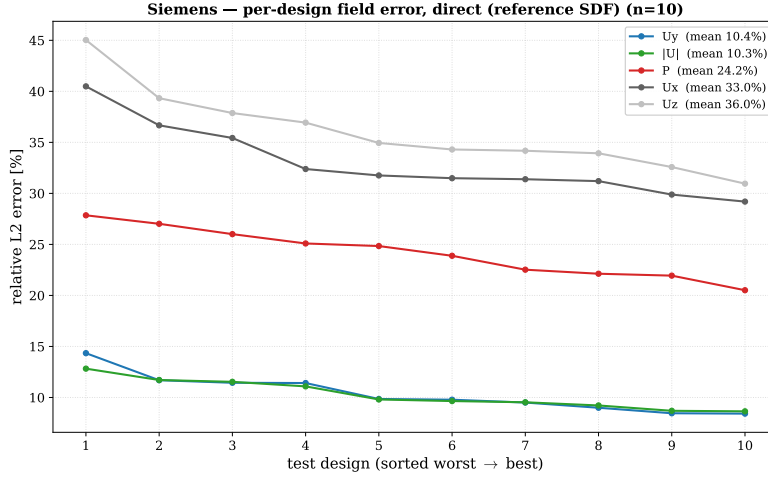


Figure 5.7: Per-design relative L^2 errors for the Siemens direct Stage 2 evaluation. The curve shows the distribution across the 10 test set industrial flow configurations and complements the mean values reported in Table 5.1.

5.4 Drag coefficient prediction

Drag is assessed separately from the volumetric fields using the scalar regression procedure defined in Section 3.4. The regressor operates on pooled features extracted from the fixed Stage 2 encoder rather than integrating the predicted volumetric fields.

Field error and drag error do not necessarily change in the same way. For example, a small error near the body might have little effect on the overall volumetric norm but could still influence aerodynamic loading. On the other hand, an error far from the body might increase the volumetric error without causing much change in drag. Also, the scalar problem uses one label per design for supervision, instead of a detailed volumetric target.

AhmedML. The Ahmed drag regressor uses the fixed scalar partition defined in Chapter 3. Feature scaling and the scalar regressors are based only on the corresponding training group.

For the test set drag group, the direct reference-SDF setup achieves a relative error of 3.25%. This equals a mean absolute error of 0.92×10^{-2} in C_D , or 9.2 drag counts. The coefficient of determination is $R^2 = 0.575$ (Figure 5.8).

The moderate R^2 value should be considered alongside the relatively narrow range of Ahmed drag coefficients. Because the denominator in the R^2 formula depends on the target variance, a narrow C_D distribution can lead to a lower coefficient of determination even if the absolute error is small. For this reason, relative error and drag-count MAE are the main metrics used here.

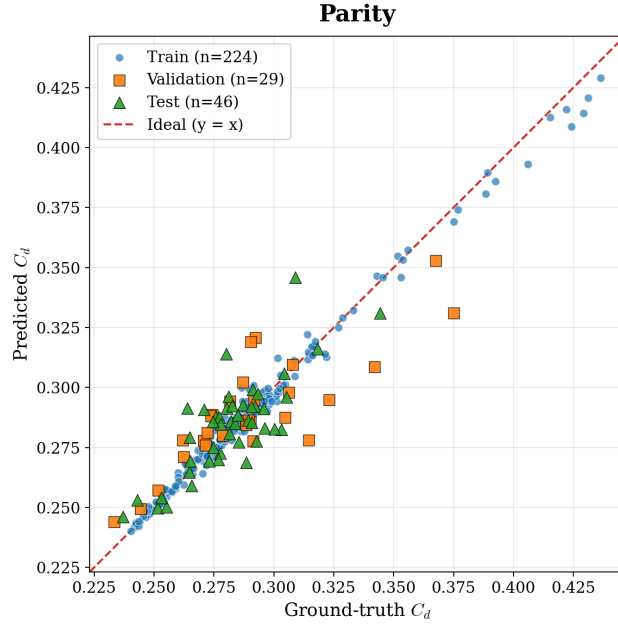


Figure 5.8: AhmedML direct drag-coefficient parity. Predicted and reference C_D values are shown for the training, validation, and test set drag groups.

DrivAerML. We evaluate DrivAerML drag using leave-one-out regression on the 97-design scalar-regression working set [114]. For each fold, we standardize the descriptors and fit both scalar regressors on 96 designs, then apply them to the remaining design. The Stage 2 feature extractor remains fixed across all folds, so the leave-one-out procedure applies only to the scalar regressor. The 97-design collection here refers to the scalar-regression evaluation set and should not be confused with the 34-design DrivAerML fine-tuning subset used by the direct Stage 2 flow model.

With this approach, the direct reference-SDF setup achieves a relative error of 2.29%, a mean absolute error of 5.1 drag counts, and $R^2 = 0.873$ (Figure 5.9).

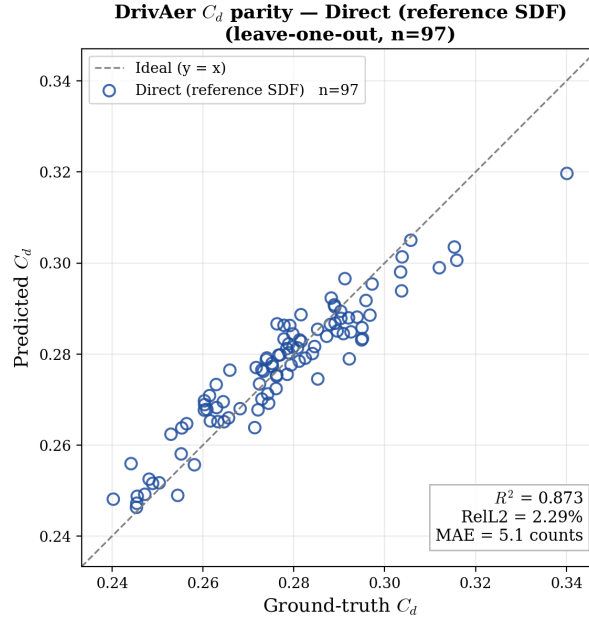


Figure 5.9: DrivAerML direct drag-coefficient parity under leave-one-out scalar regression over the 97-design scalar-regression working set. The Stage 2 feature extractor remains fixed across folds.

5.5 Discussion

The direct Stage 2 results show a consistent hierarchy across the three datasets. Streamwise velocity and velocity magnitude are predicted most accurately, pressure has intermediate error, and the transverse velocity components are more difficult in relative terms. The per-design curves show that these averages hide meaningful variation between individual geometries, particularly for the industrial Siemens cases.

The transition from AhmedML to DrivAerML and then Siemens also increases the difficulty of the prediction problem. Geometric detail, wake complexity, grid resolution, transfer learning, CFD-model differences, and available fine-tuning data change together, so the experiments measure their combined effect rather than isolating one source of error.

The common SDF input nevertheless allows the same Stage 2 architecture to be used across all three geometry families. The direct results in this chapter therefore provide the reference-SDF benchmarks for the complete surrogate evaluated in Chapter 6. Drag remains a separate scalar learning problem and is reported only for AhmedML [6] and DrivAerML [7].

The End-to-End Surrogate

This chapter evaluates the complete Stage 1–Stage 2 surrogate on the test set flow groups defined in Chapter 3. The aerodynamic model is fitted on Stage 1 reconstructed SDFs, so the geometric representation used during Stage 2 fitting is the same type used during end-to-end inference.

The evaluation includes 45 AhmedML designs, 10 DrivAerML designs, and 10 Siemens configurations. AhmedML uses the parameter-driven composition. For the realistic-vehicle transfer, the main mixed DrivAerML–Siemens experiment uses the volumetric Stage 1 route defined previously. In addition, DrivAerML is evaluated separately through a DrivAerML-only parameter-driven Stage 1 route using its available morphing parameters.

6.1 From geometry to flow field: the complete pipeline

The direct results from Chapter 5 serve as the reference-SDF benchmark. For the E2E results, a separate Stage 2 model is fitted using the SDF reconstructed by Stage 1 as input. AhmedML uses the same test set group in the two configurations. For Siemens, the direct and end-to-end models are each evaluated on ten held-out configurations, six of which are common to both. DrivAerML is different: its direct and mixed-transfer end-to-end models use different fine-tuning-set sizes and different 10-design evaluation groups. The additional DrivAerML-only parameter-driven experiment is reported separately and is not included in the direct-versus-end-to-end comparison of Table 6.1.

The effect of composition varies across the three datasets. For AhmedML, the direct and end-to-end values are computed on the same designs; for Siemens, six of the ten configurations are shared between the two evaluation groups. In both cases, the changes remain moderate, showing that the aerodynamic

Table 6.1: Direct and end-to-end field prediction. Values are mean relative L^2 errors in percent. U_{vec} denotes the relative L^2 error of the complete three-component velocity vector. For DrivAerML, the end-to-end row corresponds to the mixed realistic-vehicle configuration using the volumetric Stage 1 route. The direct and end-to-end rows use different fine-tuning sizes and different 10-design evaluation groups and therefore do not form a paired comparison. The additional DrivAerML-only parameter-driven configuration is reported separately in Section 6.2.2.

Dataset	Configuration	n	U_x	U_y	U_z	P	$ \mathbf{U} $	U_{vec}
AhmedML	Direct, reference SDF	45	21.77	3.76	16.55	13.34	3.81	4.53
	End-to-end	45	20.91	4.27	18.74	14.08	4.33	5.02
DrivAerML	Direct, reference SDF	10	28.83	4.43	31.92	18.44	4.60	5.17
	End-to-end	10	25.23	3.74	27.61	17.40	3.89	4.43
Siemens	Direct, reference SDF	10	32.99	10.39	36.01	24.18	10.27	11.46
	End-to-end	10	34.06	11.10	37.04	26.65	10.89	12.18

model continues to operate when the reference SDF is replaced by the Stage 1 reconstruction.

The DrivAerML rows require a different interpretation. The end-to-end averages are lower than the direct averages, but the two configurations use different DrivAerML fine-tuning sizes and different 10-design test sets. Their numerical difference is therefore not a controlled paired estimate of the effect of SDF reconstruction. It reflects the combined effects of geometric input representation, fine-tuning-set size, and test-set composition.

AhmedML composition. The AhmedML end-to-end evaluation uses the 45 test set flow designs defined in Chapter 3. The dataset-level means are reported in Table 6.1, while Figure 6.1 shows their distribution across individual designs.

Figure 6.2 shows a spatial view of the end-to-end prediction for Ahmed design 146. The panel compares the CFD reference and predicted velocity magnitude and pressure on the same mid-span cut, along with the corresponding absolute-error maps. For this design, the panel shows relative L^2 errors of 2.70% for $|\mathbf{U}|$ and 9.12% for pressure.

We evaluated drag using the separate scalar protocol described in Chapter 3. For the test set Ahmed drag group, the E2E configuration achieved a relative error of 5.37%, a mean absolute error of 11.3 drag counts, and $R^2 = 0.775$ (Figure 6.3).

Drag from the geometric description alone. Because the drag regressor is fitted on features of an aerodynamic network that has already been trained on the flow fields, the field task cannot be slowed by the drag task. The relevant question is instead whether the representation learned from the fields is more

6.1. From geometry to flow field: the complete pipeline

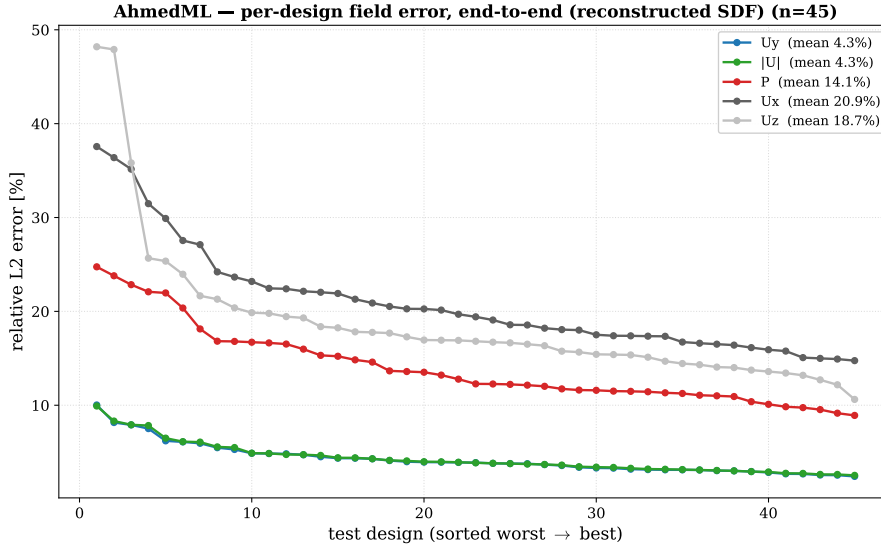


Figure 6.1: Per-design relative L^2 errors for the AhmedML end-to-end evaluation over 45 test set flow designs. Stage 2 receives the SDF reconstructed by Stage 1. The designs are displayed in the worst-to-best ordering used by the evaluation figure, and the legend reports the corresponding mean errors.

informative about drag than the geometric description itself. To answer it, drag was also regressed directly on the eight Ahmed design parameters, using the same regressor ensemble on the same partition as the learned descriptor. This comparison was carried out as a separate evaluation over the 499 designs for which both a distance field and a drag label are available, fitted on 374 and evaluated on 125, and with a Stage 1 configuration different from the one behind the result above. Its absolute values are therefore not directly comparable with that result, whereas the comparison between the two descriptors, made on identical designs, is controlled (Table 6.2).

Table 6.2: Drag regressed from the design parameters and from the learned descriptor, on identical designs and with the same regressor ensemble.

Descriptor	R^2	RelL2 (%)	MAE (counts)
Design parameters	0.597	8.23	16.0
Learned descriptor	0.712	6.96	13.3
Parameters and descriptor	0.719	6.87	13.0

The learned descriptor is the more informative of the two, raising the coefficient of determination by more than a tenth and reducing the absolute error by about three drag counts, and adding the design parameters to it changes the result very little. The representation learned from the flow fields

6. THE END-TO-END SURROGATE

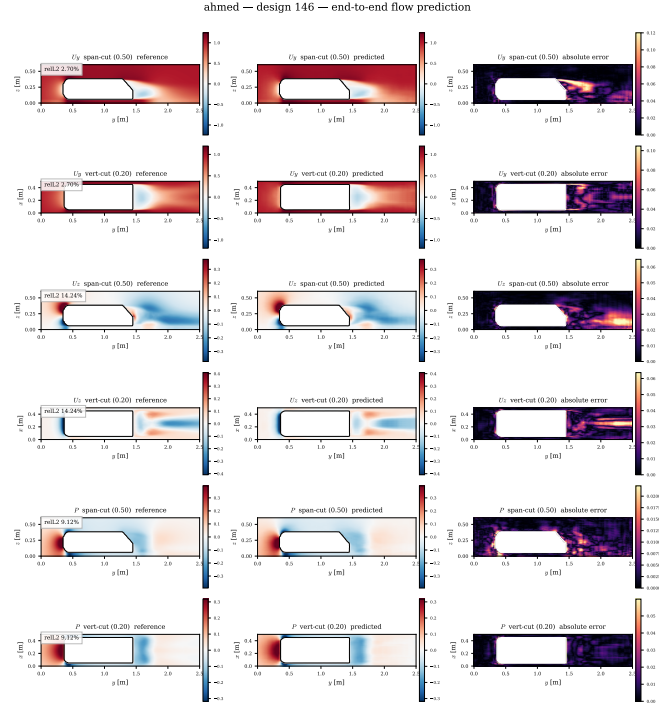


Figure 6.2: End-to-end flow prediction for Ahmed design 146. Each row contains the CFD reference, the end-to-end prediction, and the absolute-error map. The displayed relative L^2 errors are 2.70% for $|U|$ and 9.12% for P .

therefore carries the information contained in the design parameters and more. It is also the only one of the two that can be formed for a geometry with no parameter vector. A convolutional network trained end-to-end from the distance field directly to the drag coefficient was also examined during development and reached $R^2 = 0.68$, but on a different and filtered subset of designs, so it does not settle whether direct training could match the field-trained descriptor on equal terms. A cleaner isolation of the benefit of field supervision would compare against features extracted from an untrained network of the same architecture; this was not performed.

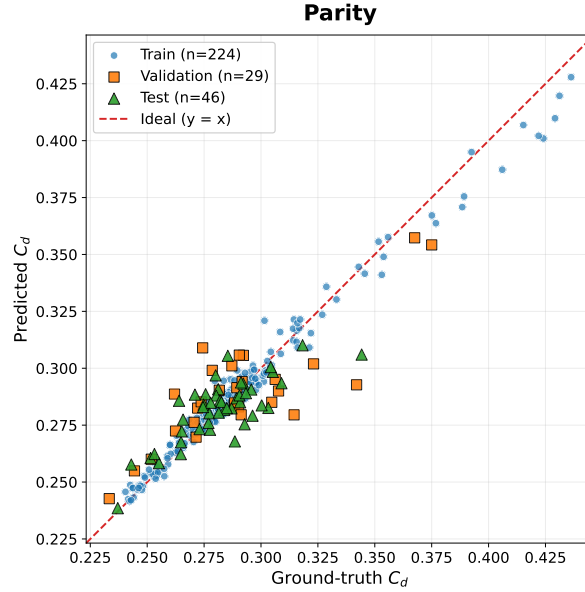


Figure 6.3: AhmedML drag-coefficient parity for the E2E configuration on the test set scalar-regression group.

6.2 End-to-end accuracy on DrivAerML

6.2.1 Mixed-transfer volumetric Stage 1 route

The DrivAerML end-to-end experiment follows the realistic-vehicle protocol defined in Chapter 3. This section reports the corresponding 10-design E2E evaluation group, which is different from the 10-design DrivAerML direct test set used in Chapter 5. The dataset-level means are listed in Table 6.1, and Figure 6.4 shows their per-design distribution.

Figure 6.5 presents the field-level results for DrivAerML design 2. Similar to the Ahmed example, the panel displays the CFD reference, end-to-end prediction, and absolute error for both velocity magnitude and pressure. The relative L^2 errors shown are 3.97% for $|\mathbf{U}|$ and 16.59% for pressure.

The end-to-end averages are lower than the direct averages for all five reported quantities, but this is not a controlled paired comparison. The E2E model uses a larger DrivAerML fine-tuning set, and the two configurations are evaluated on different 10-design groups.

6.2.2 DrivAerML-only parameter-driven Stage 1 route

In addition to the mixed DrivAerML-Siemens configuration described previously, a separate DrivAerML-only Stage 1 fine-tuning experiment was conducted using the 16 morphing parameters specific to DrivAerML. In this

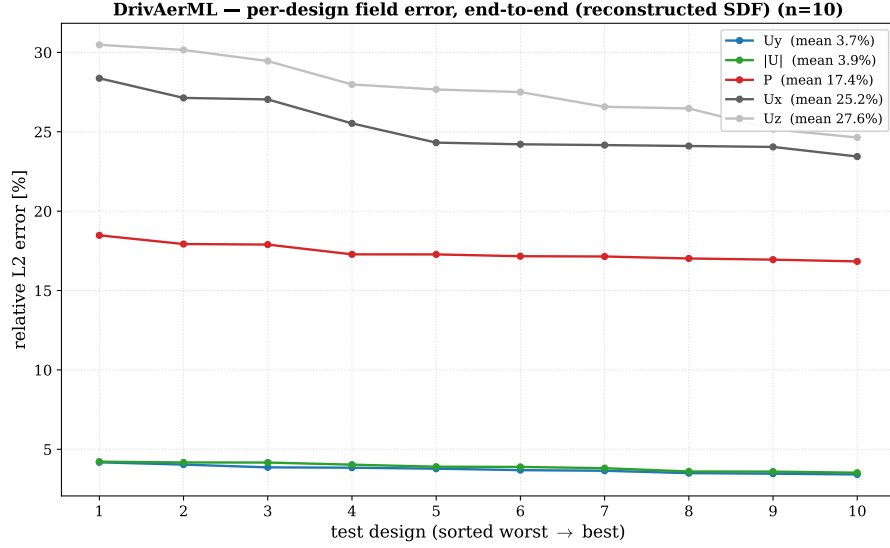


Figure 6.4: Per-design relative L^2 errors for the DrivAerML end-to-end evaluation over 10 test set flow designs. The DrivAerML geometry is represented as an SDF and enters the volumetric Stage 1 branch; the reconstructed SDF is then supplied to the E2E Stage 2 U-Net. The design-by-design errors are displayed in the sorted order used by the evaluation figure.

setup, the parameters were provided to the parameter-driven Stage 1 branch instead of supplying the geometry as an input signed distance function (SDF). The parameter vector was mapped to the learned latent representation and subsequently decoded into the reconstructed SDF. The reconstructed field was then resampled to the Stage 2 resolution and forwarded to the end-to-end Stage 2 U-Net. Siemens configurations were excluded from this Stage 1 parameter fine-tuning because they do not share the DrivAerML morphing-parameter space.

This supplementary experiment evaluates the full DrivAerML configuration, starting directly from the available design variables,

$$\mu \longrightarrow \hat{\varphi} \longrightarrow \hat{\mathbf{q}},$$

where μ denotes the DrivAerML morphing parameters, $\hat{\varphi}$ is the reconstructed SDF, and $\hat{\mathbf{q}}$ denotes the predicted aerodynamic fields.

Figure 6.6 reports the field errors over the 10 DrivAerML test designs for this parameter-driven composition. The mean relative L^2 errors are 33.96% for U_x , 7.92% for the streamwise velocity U_y , 36.56% for U_z , 28.07% for pressure, and 7.97% for the derived velocity magnitude $|\mathbf{U}|$.

Figures 6.7 and 6.8 show DrivAerML design 114 as a spatial example of the same parameter-driven composition. The reconstructed SDF obtained from the 16 design parameters is used as the Stage 2 input.

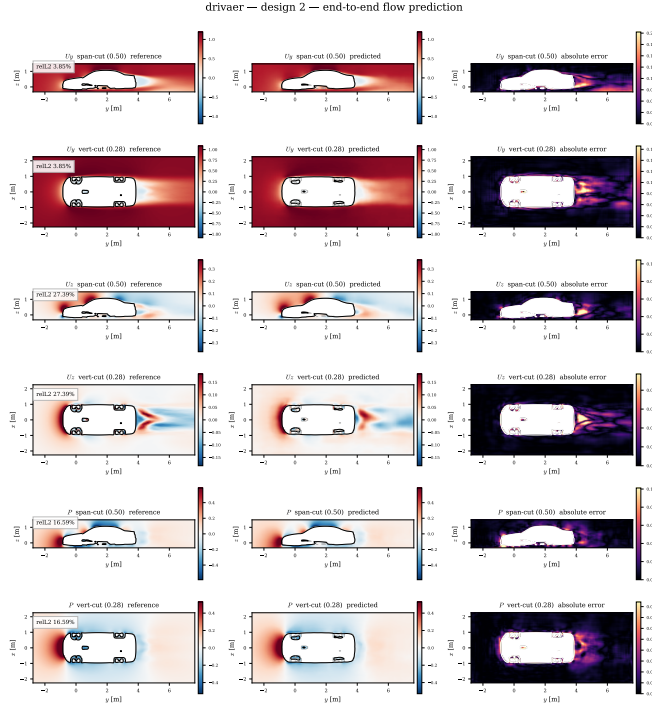


Figure 6.5: End-to-end flow prediction for DrivAerML design 2. Each row contains the CFD reference, the end-to-end prediction, and the absolute-error map. The displayed relative L^2 errors are 3.97% for $|\mathbf{U}|$ and 16.59% for P .

The parameter-driven results are reported as an additional DrivAerML-only configuration rather than replacing the mixed-transfer volumetric benchmark in Table 6.1. They demonstrate that the same complete Stage 1–Stage 2 composition can start directly from DrivAerML design parameters when such a parameterization is available.

Drag. DrivAerML drag is evaluated using leave-one-out scalar regression on the 97-design scalar-regression working set [114]. The learned Stage 1 and Stage 2 representations remain fixed across all folds. The direct setup achieves a 2.29% relative error, 5.1 drag counts MAE, and $R^2 = 0.873$, while the E2E setup gives 3.65%, 7.8 drag counts, and $R^2 = 0.677$ (Table 6.3 and Figure 6.9). These values compare two fixed feature extractors produced by their respective direct and end-to-end flow-training protocols.

6. THE END-TO-END SURROGATE

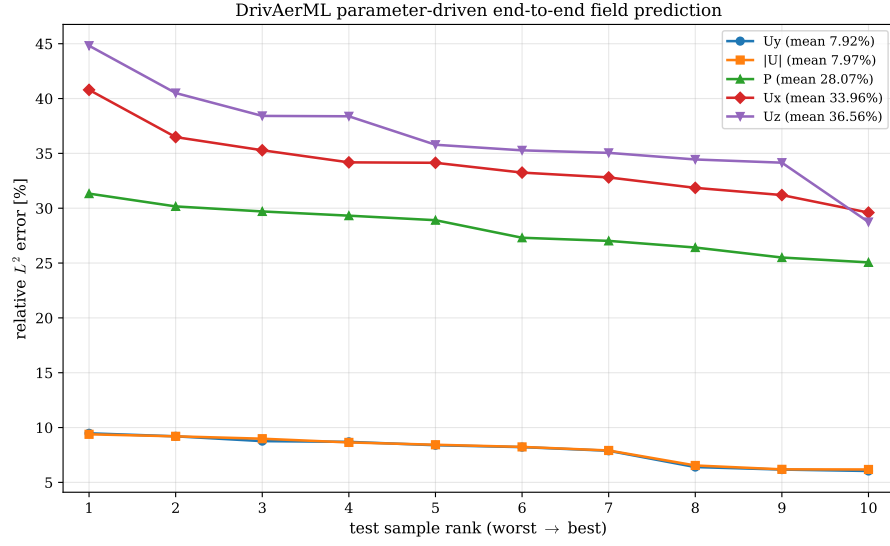


Figure 6.6: Per-design relative L^2 errors for the DrivAerML parameter-driven end-to-end experiment over 10 test designs. Stage 1 receives the DrivAerML morphing parameters and reconstructs the SDF before Stage 2 predicts the aerodynamic fields. For each reported quantity, the 10 errors are independently ordered from worst to best; consequently, the same rank on the horizontal axis does not necessarily correspond to the same design for different quantities. The mean errors are 33.96% for U_x , 7.92% for U_y , 36.56% for U_z , 28.07% for P , and 7.97% for $|U|$.

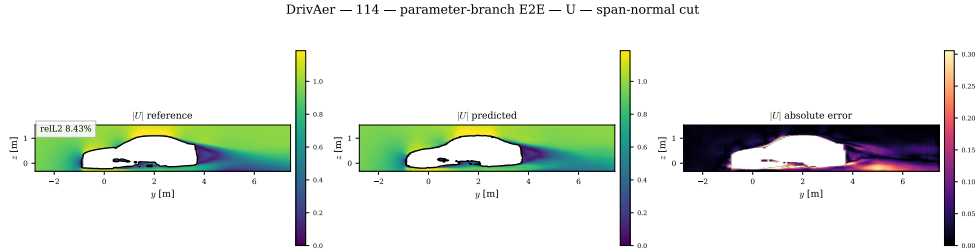


Figure 6.7: Parameter-driven end-to-end velocity-magnitude prediction for DrivAerML design 114 on a span-normal cut. The panels show the CFD reference, the Stage 2 prediction obtained from the parameter-reconstructed SDF, and the corresponding absolute error. The displayed relative L^2 error for $|U|$ is 8.43%.

Table 6.3: DrivAerML drag-coefficient accuracy under leave-one-out scalar regression over the 97-design scalar-regression working set. The scalar regressor is refitted on 96 designs in each fold, while the learned neural representations remain fixed.

Configuration	Relative L^2 error	MAE [drag counts]	R^2
Direct (reference SDF)	2.29%	5.1	0.873
E2E (reconstructed SDF)	3.65%	7.8	0.677

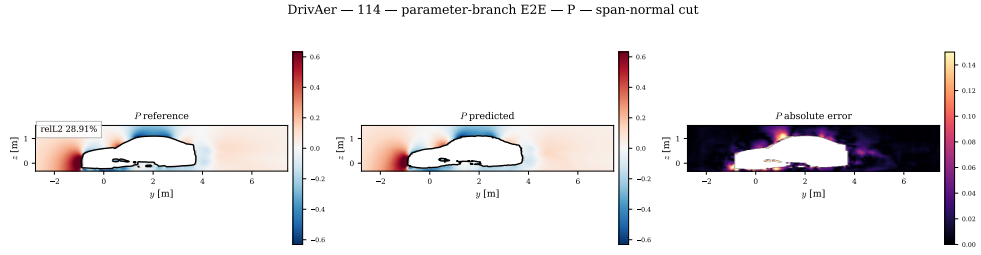


Figure 6.8: Parameter-driven end-to-end pressure prediction for DrivAerML design 114 on a span-normal cut. The panels show the CFD reference, the Stage 2 prediction obtained from the parameter-reconstructed SDF, and the corresponding absolute error. The displayed relative L^2 error for pressure is 28.91%.

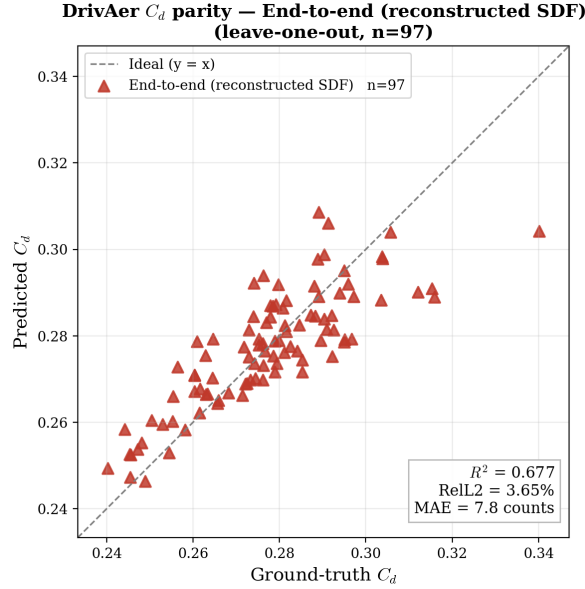


Figure 6.9: DrivAerML E2E drag-coefficient parity under leave-one-out scalar regression over the 97-design scalar-regression working set.

6.3 End-to-end accuracy on Siemens

The Siemens end-to-end evaluation adheres to the joint realistic-vehicle protocol outlined in Chapter 3. Reconstructed signed distance functions (SDFs) are generated using the pooled DrivAerML–Siemens volumetric Stage 1 model. For the separate industrial geometry-reconstruction assessment, we employ the Siemens-only Stage 1 fine-tune described in Chapter 4. The evaluation uses ten held-out industrial configurations, six of which also belong to the direct Siemens benchmark. The mean values are listed in Table 6.1. Figure 6.10 shows that the industrial configurations exhibit a wider spread of errors, especially for the transverse velocity components and pressure.

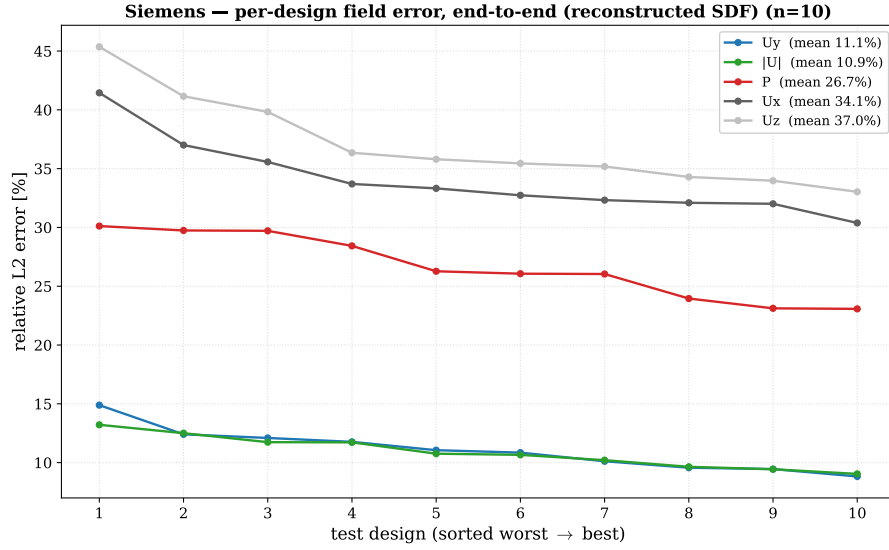


Figure 6.10: Per-design relative L^2 errors for the Siemens end-to-end evaluation over 10 test set industrial flow configurations. The geometry-derived SDF is reconstructed through the volumetric Stage 1 branch before being supplied to the E2E Stage 2 network. The configurations are displayed in the sorted order used by the evaluation figure. The legend reports mean errors of 34.1% for U_x , 11.1% for U_y , 37.0% for U_z , 26.7% for pressure, and 10.9% for velocity magnitude.

Figure 6.11 shows a field-level industrial example for Design_6.2. It compares the CFD reference with the end-to-end predicted velocity magnitude and pressure, and also includes absolute-error maps. The relative L^2 errors shown are 12.50% for $|U|$ and 26.27% for pressure.

Shape of the predicted wake. The end-to-end prediction in Figure 6.11 shows a more diffuse shear layer at the top of the wake compared to the direct example in Chapter 5. The angle at which the flow leaves the roof is also less clearly defined. However, the two panels show different vehicles: the direct example uses Design_5.2, while the end-to-end example uses Design_6.2. Because of this, they are not a direct comparison, and some of the visible differences come from the geometry of each design. Still, the pipeline itself may also play a role. For the industrial configurations, the geometry is reconstructed at 64^3 , so each streamwise cell is about 0.10 m, which is twice the spacing of the reference field. This means the sharp upper rear edge that sets the separation point is rounded in the reconstructed input. The end-to-end network is trained on reconstructed, not exact, geometry, so it may learn to make predictions that handle uncertainty in the edge position, which could blur the shear layer that starts there. The dataset-level errors in Table 6.1 support this idea: the end-to-end pressure error is about 2.5 percentage points higher than the direct one.

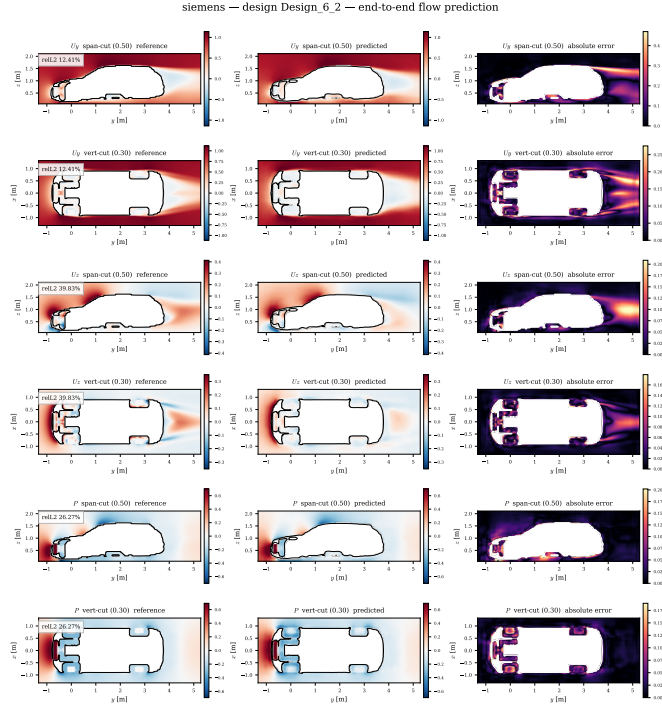


Figure 6.11: End-to-end flow prediction for Siemens configuration Design_6_2. Each row contains the CFD reference, the end-to-end prediction, and the absolute-error map. The displayed relative L^2 errors are 12.50% for $|\mathbf{U}|$ and 26.27% for P .

6.4 Discussion

The end-to-end experiments test whether the reconstructed SDF can act as the learned interface between geometry and volumetric aerodynamic prediction.

For AhmedML and Siemens, the changes in error are moderate, indicating that the complete composition remains stable when Stage 2 operates on reconstructed rather than reference geometry.

The additional DrivAerML-only experiment evaluates a different Stage 1 entry mechanism. Instead of beginning from a geometry-derived SDF, Stage 1 receives the 16 DrivAerML morphing parameters and reconstructs the SDF through the parameter-driven branch before Stage 2 inference. Its mean errors are 33.96% for U_x , 7.92% for U_y , 36.56% for U_z , 28.07% for pressure, and 7.97% for velocity magnitude. This result is used to demonstrate the flexibility of the geometric entry stage when an explicit parameterization is available, rather than to replace the mixed-transfer benchmark.

The drag results form a separate scalar-learning problem. They show that useful scalar predictions can still be obtained from fixed Stage 2 representations, but the direct and E2E DrivAerML drag results inherit the differences

between their corresponding neural feature extractors.

Overall, the reconstructed SDF functions as a practical intermediate representation in the complete surrogate. The per-design curves and field panels show how this behavior varies across individual geometries and where the remaining spatial discrepancies are concentrated.

6.5 Comparison with the published state of the art

AhmedML and DrivAerML provide the most relevant public benchmarks for placing the present field-prediction accuracy in context [6, 7]. AB-UPT [3] reports relative L^2 errors for pressure and for the complete three-component velocity vector. To make the comparison consistent, the present work therefore uses the corresponding vector quantity U_{vec} rather than the derived velocity magnitude $|\mathbf{U}|$, as listed in Table 6.4.

Table 6.4: Contextual comparison with published AB-UPT results on AhmedML and DrivAerML. Values are relative L^2 errors in percent; lower is better. U_{vec} denotes the complete three-component velocity-vector error. The DrivAerML present-work row corresponds to the mixed-transfer volumetric end-to-end configuration of Table 6.1; the additional parameter-driven DrivAerML-only experiment is reported separately in Section 6.2.2. The numerical values are not treated as a strict head-to-head ranking because the methods use different spatial discretizations and evaluation domains.

Method	Dataset	U_{vec}	P	Evaluation domain
AB-UPT [3]	AhmedML	1.90	1.98	CFD mesh points
This work, E2E	AhmedML ($n = 45$)	5.02	14.08	128^3 fluid region
AB-UPT [3]	DrivAerML	5.93	6.08	CFD mesh points
This work, E2E	DrivAerML ($n = 10$)	4.43	17.40	128^3 fluid region

Interpretation of the comparison. Using the complete velocity vector gives a more meaningful comparison than using velocity magnitude. On AhmedML, AB-UPT reports lower relative errors for both velocity and pressure. On DrivAerML, the present end-to-end result has a numerically lower velocity-vector error, 4.43% compared with 5.93%, whereas AB-UPT reports a substantially lower pressure error, 6.08% compared with 17.40%.

These numbers should not be interpreted as a strict ranking of the two methods. AB-UPT evaluates predictions on CFD mesh points, whereas the present framework predicts on a fixed 128^3 Cartesian representation and evaluates the fluid region of that grid. The models also differ in architecture, geometric representation, and experimental protocol. The comparison therefore shows the relative accuracy level reached by each approach for the same

physical output quantities, rather than establishing a controlled head-to-head benchmark.

Drag coefficient. AB-UPT also measures the DrivAerML drag coefficient. Instead of predicting C_D directly, it estimates surface pressure and wall shear stress, then integrates these surface forces to get the aerodynamic coefficients. The reported DrivAerML drag result is $R^2 = 0.963$ [3]. In contrast, our framework takes a different approach by regressing C_D from pooled Stage 2 encoder features. Using the leave-one-out scalar-regression protocol, the direct configuration achieves $R^2 = 0.873$, a relative error of 2.29%, and 5.1 drag counts MAE. The end-to-end configuration gives $R^2 = 0.677$, 3.65%, and 7.8 drag counts. While the R^2 values help provide context, they are not a direct comparison because the evaluation protocols differ. AB-UPT calculates drag using high-resolution surface-force integration on its DrivAerML test partition, while our method uses leave-one-out scalar regression on fixed learned descriptors.

Position of the present work. The comparison shows a mixed accuracy picture rather than a uniform advantage for either approach: AB-UPT is substantially more accurate for pressure on the reported open-source benchmarks, while the present DrivAerML E2E result has a lower reported velocity-vector error. The contribution of this thesis is therefore not framed as universally superior field accuracy. It is the integration of an explicit, independently evaluated SDF reconstruction stage with volumetric flow prediction and transfer to an industrial Siemens geometry family through the same geometric interface. The additional DrivAerML-only parameter-driven experiment further shows that the geometric stage can also begin directly from design parameters when a suitable parameterization is available.

6.6 Toward deployment: accuracy requirements and inference cost

Reference CFD cost. The computational comparison helps show how surrogate inference fits within the context of the simulations that were used to create the open-source datasets.

AhmedML uses a time-accurate hybrid RANS–LES approach with the Spalart–Allmaras delayed detached-eddy model in OpenFOAM v2212. The meshes have about 15 to 20 million cells, and the Reynolds number is 2.78×10^6 based on body length [6]. According to the published campaign, meshing takes about 30 minutes and solving takes 48 hours on 288 CPU cores, which is about 1.4×10^4 core-hours per design [6].

DrivAerML uses a similar type of scale-resolving method on meshes with about 10^8 cells. A typical case takes around 40 hours on 1536 CPU cores, or about 6.1×10^4 core-hours per design [7].

These costs do not directly reflect geometric complexity. They also depend on solver accuracy, mesh size, how the solution converges, and the computing setup.

Surrogate inference cost. The full learned model takes about 6.5 seconds on a single GPU to go from geometric input to reconstructed SDF and aerodynamic outputs, as summarised in Table 6.5. Compared to the published AhmedML wall-clock time, this is about four orders of magnitude faster.

This speed-up is not normalized for hardware. The CFD and surrogate timings use different machines, and the surrogate gives a learned mean-flow approximation at the accuracy described in this thesis, not a full CFD replacement. Training also has a high one-time cost and relies on a precomputed CFD database. The main benefit comes when this cost is spread over many later design queries [91, 26].

Table 6.5: Cost of one aerodynamic evaluation. Reference costs are those reported for the public datasets [6, 7]; the surrogate is timed on a single GPU. The hardware is different and the table must not be interpreted as a normalized speed-up.

	AhmedML	DrivAerML	Surrogate
Method	hybrid RANS–LES	hybrid RANS–LES	3D CNN
Mesh	$\sim 2 \times 10^7$ cells	$\sim 10^8$ cells	128^3 grid
Solution	48 h on 288 cores	40 h on 1536 cores	6.5 s on one GPU
Per design	$\sim 1.4 \times 10^4$ core-hours	$\sim 6.1 \times 10^4$ core-hours	$\sim 2 \times 10^{-3}$ GPU-hours
Output	time-averaged fields	time-averaged fields	mean fields, C_D

Use in the design loop. The surrogate is intended as a screening workflow rather than a substitute for high-fidelity CFD. Its short inference time makes it suitable for examining predicted wake and pressure patterns and identifying designs that may require more detailed simulation [91, 26]. A larger and more varied industrial evaluation set would still be required to validate design ranking reliably.

Conclusion and Perspectives

This thesis explored whether a learned geometric representation based on signed distance fields can provide a practical link between vehicle geometry and fast aerodynamic prediction. The study moved from simple analytical shapes to AhmedML, DrivAerML, and industrial Siemens configurations, allowing the same framework to be examined as geometric and aerodynamic complexity increased.

The results support the use of the SDF as an explicit intermediate representation between geometric reconstruction and volumetric flow prediction. The main findings, limitations, and possible extensions are summarized below.

7.1 Contribution

The main contribution of this thesis is an explicit two-stage surrogate in which the signed distance field provides a common geometric interface. Stage 1 can be evaluated independently from the aerodynamic model, while Stage 2 can be assessed either from the reference SDF or as part of the complete Stage 1–Stage 2 composition.

The geometry experiments show that the representation preserves the main vehicle envelope across AhmedML, DrivAerML, and Siemens. The largest reconstruction differences are concentrated around sharp edges, thin parts, narrow gaps, and other details that approach the resolution limit of the Cartesian grid. These results establish the geometric stage as a measurable component of the surrogate rather than only an internal preprocessing step.

The direct aerodynamic experiments show a consistent hierarchy across the three vehicle datasets: streamwise velocity and velocity magnitude are predicted more accurately than the transverse velocity components, while pressure has intermediate error. The realistic-vehicle experiments also show

the combined influence of more detailed geometry, transfer learning, limited industrial data, and differences in the CFD reference solutions.

The end-to-end experiments show that the reconstructed SDF can be used as the actual Stage 2 input during both training and inference.

Drag remains a separate scalar-learning problem. The results show that useful drag estimates can be obtained from pooled Stage 2 features for AhmedML and DrivAerML, while no Siemens drag benchmark is reported.

A full inference from geometric input to reconstructed SDF and aerodynamic fields takes about 6.5 s on a single GPU. This timing is not directly comparable with the CFD wall-clock costs because the hardware and outputs differ, but it illustrates the intended many-query use of the surrogate once the CFD database and trained models are available.

Published comparisons also define the limits of the present work. AB-UPT reports lower volumetric field errors, especially for pressure [3]. Our main contribution is methodological: an independently testable geometric stage, an explicit SDF interface between geometry and flow prediction, and a common realistic-vehicle transfer route for DrivAerML and Siemens. Full software integration into the Siemens SciML Sandbox remains future work.

7.2 Limitations

Spatial resolution remains a limitation. Stage 2 uses a 128^3 Cartesian grid, while Stage 1 reconstructs the realistic vehicle geometries at 64^3 before resampling. At full-vehicle scale, mirror supports, narrow gaps, underbody parts, and other small features can approach the grid resolution [120].

The aerodynamic targets are limited to the mean quantities available in the CFD data. AhmedML and DrivAerML provide time-averaged scale-resolving solutions, while Siemens provides steady RANS fields. The surrogate therefore learns these mean-flow patterns rather than instantaneous turbulent behavior such as vortex shedding, wake oscillations, or transient separation [6, 7].

This work does not provide calibrated predictive uncertainty. Error maps indicate where predictions differ from the reference, but they do not provide a formal confidence estimate for a new design [51].

Drag is also a small-data scalar problem. Each geometry provides one coefficient, while the volumetric-flow model receives spatial supervision over the full field. The current regressor does not enforce the physical connection between predicted pressure, wall shear stress, and total aerodynamic force, and no industrial Siemens drag study is included.

The industrial conclusions are limited by the 20 available Siemens configurations and, in particular, by the 10-design test set flow evaluation. The reported transfer results therefore describe this industrial collection rather than the broader variability of production vehicle families.

The SDF removes direct dependence on the original design variables, but not on discretization or preprocessing. Bounding box, grid resolution, voxel spacing, orientation, sign convention, and surface quality still influence the sampled representation.

Finally, the latent representation is evaluated here as part of the reconstruction and prediction framework, not as a generative or optimization space which could be a next goal.

7.3 Perspectives

The next step is to validate the results more broadly in industry. Using a larger, independent Siemens or multi-vehicle dataset would let us test transfer behavior across more geometry types. This would give us a better way to judge when the surrogate is accurate enough for engineering screening. Expanding the dataset would also help us see whether any issues come from limited training data or from differences in geometry, strategy, CFD setup, or turbulence modeling. In industrial technology transfer, the surrogate model works best as an extra tool in the simulation workflow, not as a replacement for high-fidelity CFD. Its main advantage is that it can quickly provide aerodynamic feedback during design stages, when many geometries must be compared, without requiring as many resources as a full CFD calculation. These candidate geometries can be converted to the standard SDF format, run through the trained pipeline, and compared based on the predicted velocity and pressure fields. The most promising designs can then be selected for detailed, high-fidelity simulation. A logical progression for technology transfer involves integrating the developed methods into the Siemens Scientific Machine Learning (SciML) Sandbox, as described in Chapter 1. While this thesis focuses on the scientific development, software integration is planned for a subsequent phase. Employing a modular approach enables geometry preprocessing and signed distance function (SDF) construction, Stage 1 geometry reconstruction, Stage 2 aerodynamic prediction, and post-processing to be managed as distinct components. This structure maintains the stage-wise validation strategy established in the thesis and facilitates the updating or retraining of individual surrogate components as new industrial data become available. For practical industrial implementation, additional software and validation mechanisms are necessary. These mechanisms include automated geometry quality checks, standardized coordinate and normalization conventions, model and dataset versioning, monitoring of prediction errors on

newly simulated designs, and criteria for identifying configurations outside the training data range. Implementing these measures makes it possible to determine when the surrogate model is suitable for rapid screening and when a new high-fidelity computational fluid dynamics (CFD) calculation is warranted. An additional industrial perspective involves progressively integrating new computational fluid dynamics (CFD) data. As further industrial CFD simulations become available, these can be incorporated into an expanding training database to update or fine-tune the surrogate model. Consequently, the model would not remain a static predictor trained on a fixed dataset, but would gradually adapt to new vehicle families, geometric features, and operating conditions. Cases where the surrogate shows significant errors or low confidence could be prioritized for new high-fidelity simulations. This approach would establish an iterative workflow, with CFD serving as the reference source of physical information while the surrogate increasingly facilitates rapid screening and design-space exploration. Enhancing spatial resolution represents a promising research direction. Sparse voxel, adaptive grid, or hybrid surface-volume models can concentrate resolution near critical geometric and aerodynamic features without necessitating a uniformly fine three-dimensional grid [95, 18, 11]. Additionally, surface-aware objectives may help preserve thin geometric structures and near-body flow features [120]. Drag prediction can be made more physically explicit by predicting surface pressure and wall shear stress, then integrating the resulting loads. This approach could complement or replace the current compact scalar regressor when suitable surface labels are available [3]. The current surrogate model predicts deterministic mean fields. Future extensions could incorporate time-dependent or statistically richer outputs, such as flow variance, wake-oscillation amplitude, spectral quantities, or selected unsteady modes [103, 29]. Additionally, calibrated uncertainty estimation would help identify cases where interpolation is reliable and those requiring new high-fidelity computational fluid dynamics (CFD) calculations [51]. The learned latent representation may facilitate future design operations. Interpolation, sampling, and inverse design can be formulated within the compact geometric space [83, 117]. In industrial applications, this approach offers a pathway from passive prediction to assisted design exploration, contingent upon the enforcement of geometric validity and engineering constraints during the search. The SDF interface applies to a broader range of scenarios than external vehicle aerodynamics alone. Finally this representation can be integrated with various predictors to address internal flows, thermal fields, and other simulation challenges where geometry is a primary factor.

Bibliography

- [1] Syed R Ahmed, G Ramm, and Gunter Faltin. Some salient features of the time-averaged ground vehicle wake. In *International Congress & Exposition*, 1984.
- [2] DE Aljure, Joan Calafell, Aleix Baez, and Assensi Oliva. Flow over a realistic car model: Wall modeled large eddy simulations assessment and unsteady effects. *Journal of Wind Engineering and Industrial Aerodynamics*, 174:225–240, 2018.
- [3] Benedikt Alkin, Maurits Bleeker, Richard Kurle, Tobias Kronlachner, Reinhard Sonnleitner, Matthias Dorfer, and Johannes Brandstetter. AB-UPT: Scaling neural CFD surrogates for high- fidelity automotive aerodynamics simulations via anchored- branched universal physics transformers. *Transactions on Machine Learning Research*, 2025.
- [4] Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 2024.
- [5] David Amsallem, Matthew J. Zahr, and Charbel Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92:891–916.
- [6] Neil Ashton, Danielle C Maddix, Samuel Gundry, and Parisa M Shabestari. Ahmedml: High-fidelity computational fluid dynamics dataset for incompressible, low-speed bluff body aerodynamics. *arXiv preprint arXiv:2407.20801*, 2024.
- [7] Neil Ashton, Charles Mockett, Marian Fuchs, Louis Fliessbach, Hendrik Hetmann, Thilo Knacke, Norbert Schonwald, Vangelis Skaperdas,

- Grigoris Fotiadis, Astrid Walle, et al. Drivaerml: High-fidelity computational fluid dynamics dataset for road-car external aerodynamics. *arXiv preprint arXiv:2408.11969*, 2024.
- [8] T Avadiar, MC Thompson, J Sheridan, and D Burton. Characterisation of the wake of the drivaer estate vehicle. *Journal of Wind Engineering and Industrial Aerodynamics*, 177:242–259, 2018.
- [9] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.
- [10] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.
- [11] Marsha J Berger and Phillip Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics*, 82(1):64–84, 1989.
- [12] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.
- [13] Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64(2):525–545, 2019.
- [14] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.
- [15] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual review of fluid mechanics*, 52(1):477–508, 2020.
- [16] CAE ML Datasets. DrivAerML. Accessed: 01 August 2026.
- [17] Haecheon Choi and Parviz Moin. Grid-point requirements for large eddy simulation: Chapman’s estimates revisited. *Physics of fluids*, 24(1), 2012.
- [18] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 3070–3079. IEEE, 2019.

-
- [19] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.
- [20] Murray Cutforth and Shahab Mirjalili. Convolutional autoencoders for the reconstruction of three-dimensional interfacial multiphase flows. *AI Thermal Fluids*, page 100035, 2026.
- [21] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- [22] Michel C Delfour and J-P Zolésio. *Shapes and geometries: metrics, analysis, differential calculus, and optimization*. SIAM, 2011.
- [23] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual review of fluid mechanics*, 51(1):357–377, 2019.
- [24] Mohamed Elrefaie, Angela Dai, and Faez Ahmed. Drivaernet: A parametric car dataset for data-driven aerodynamic design and graph-based drag prediction. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 88360, page V03AT03A019. American Society of Mechanical Engineers, 2024.
- [25] Joel H Ferziger, Milovan Perić, and Robert L Street. *Computational methods for fluid dynamics*, volume 3. Springer, 2002.
- [26] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.
- [27] Stefania Fresca and Andrea Manzoni. Pod-dl-rom: Enhancing deep learning-based reduced order models for nonlinear parametrized pdes by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 388:114181, 2022.
- [28] Jochen Fröhlich and Dominic von Terzi. Hybrid les/rans methods for the simulation of turbulent flows. *Progress in Aerospace Sciences*, 44:349–377.
- [29] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.

- [30] Joshua Fuller and Martin A Passmore. The importance of rear pillar geometry on fastback wake structures. *Journal of Wind Engineering and Industrial Aerodynamics*, 125:111–120, 2014.
- [31] Roberto Garcia-Fernandez, Koldo Portal-Porras, Oscar Irigaray, Zugatz Ansa, and Unai Fernandez-Gamiz. Cnn-based flow field prediction for bus aerodynamics analysis. *Scientific Reports*, 13(1):21213, 2023.
- [32] Fabrizio Garotta, Nicola Demo, Marco Tezzele, Massimo Carraturo, Alessandro Reali, and Gianluigi Rozza. Reduced order isogeometric analysis approach for pdes in parametrized domains. In *Quantification of Uncertainty: Improving Efficiency and Technology: QUIET Selected Contributions*, pages 153–170. Springer, 2020.
- [33] Massimo Germano, Ugo Piomelli, Parviz Moin, and William H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids*, 3:1760–1765.
- [34] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.
- [35] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. A review of level-set methods and some recent applications. *Journal of Computational Physics*, 353:82–109, 2018.
- [36] M Grandemange, M Gohlke, and O Cadot. Turbulent wake past a three-dimensional blunt body. part 1: Global modes and bi-stability. *Journal of Fluid Mechanics*, 722:51–84.
- [37] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.
- [38] Emmanuel Guilmineau, Guo-Bin Deng, Alban Leroyer, Patrick Queutey, Michel Visonneau, and Jeroen Wackers. Assessment of hybrid rans-les formulations for flow simulation around the ahmed body. *Computers & Fluids*, 176:302–319, 2018.
- [39] Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 481–490, 2016.
- [40] Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International conference on machine learning*, pages 12556–12569. PMLR, 2023.

-
- [41] Kan He, Guglielmo Minelli, Jiabin Wang, Tianyun Dong, Guangjun Gao, and Sinisa Krajnovic. Numerical investigation of the wake bi-stability behind a notchback ahmed body. *Journal of Fluid Mechanics*, 926.
- [42] Angelina I Heft, Thomas Indinger, and Nikolaus A Adams. Introduction of a new realistic generic car model for aerodynamic investigations. In *SAE 2012 World Congress & Exhibition 192355*, 2012.
- [43] Jan S Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.
- [44] Saddam Hijazi, Giovanni Stabile, Andrea Mola, and Gianluigi Rozza. Data-driven pod-galerkin reduced order model for turbulent flows. *Journal of Computational Physics*, 416:109513, 2020.
- [45] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [46] Wolf-Heinrich Hucho, Gino Sovran, et al. *Aerodynamics of road vehicles*, volume 398. Butterworths London, 1987.
- [47] D P Huttenlocher, G A Klanderman, and W J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863.
- [48] 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, 2022.
- [49] Efthymios N Karatzas, Giovanni Stabile, Francesco Ballarin, and Gianluigi Rozza. Reduced basis, embedded methods and parametrized levelset geometry. *arXiv preprint arXiv:2301.12401*, 2023.
- [50] Kenji Kawaguchi, Masanori Hashiguchi, Ryuuichi Yamasaki, and Kunio Kuwahara. Computational study of the aerodynamic behavior of a three-dimensional car configuration. In *International Congress and Exposition*, 1989.
- [51] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

- [52] Ashok Khondge and Sandeep Sovani. An accurate, extensive, and rapid method for aerodynamics optimization: The 50: 50: 50 method. In *SAE 2012 World Congress & Exhibition 192355*, 2012.
- [53] Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.
- [54] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [55] Sinisa Krajnovic and Lars Davidson. Flow around a simplified car, part 2: Understanding the flow. *Journal of Fluids Engineering*, 127:919–928.
- [56] Siniša Krajnović and Lars Davidson. Flow around a simplified car, part 1: large eddy simulation. 2005.
- [57] Toni Lassila and Gianluigi Rozza. Parametric free-form shape design with pde models and reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1583–1592, 2010.
- [58] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [59] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.
- [60] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.
- [61] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [62] Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023.

-
- [63] Hermann Lienhart and Stefan Becker. Flow and turbulence structure in the wake of a simplified car model. In *2003 SAE World Congress*, 2003.
- [64] Ning Liu, Siavash Jafarzadeh, and Yue Yu. Domain agnostic fourier neural operators. *Advances in neural information processing systems*, 36:47438–47450, 2023.
- [65] I. Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*.
- [66] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- [67] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deepnet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.
- [68] Sophie Luckhurst, Max Varney, Hao Xia, Martin A Passmore, and A Gaylard. Computational investigation into the sensitivity of a simplified vehicle wake to small base geometry changes. *Journal of Wind Engineering and Industrial Aerodynamics*, 185:1–15, 2019.
- [69] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *arXiv preprint arXiv:2011.13495*, 2020.
- [70] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.
- [71] Florian R Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, 32(8):1598–1605, 1994.
- [72] Florian R Menter, Andreas Hüppe, David Flad, Andrey V Garbaruk, Alexey A Matyushenko, and Andrey S Stabnikov. Large eddy simulations for the ahmed car at 25° slant angle at different reynolds numbers. *Flow, Turbulence and Combustion*, 112(1):321–343, 2024.
- [73] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465. IEEE, 2019.

- [74] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.
- [75] Mehdi Mortazawy, Mukul Rao, Jonathan Jilesen, Dalon Work, and Richard Shock. Early stage vehicle aerodynamics development using a gpu based lbm cfd solver. In *WCX SAE World Congress Experience 288442*, 2023.
- [76] Takaaki Murata, Kai Fukami, and Koji Fukagata. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics*, 882:A13, 2020.
- [77] Ken Museth. Vdb: High-resolution sparse volumes with dynamic topology. *ACM transactions on graphics (TOG)*, 32(3):1–22, 2013.
- [78] Mario Ohlberger and Stephan Rave. Reduced basis methods: Success, limitations and future challenges. *arXiv preprint arXiv:1511.02021*, 2015.
- [79] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton–jacobi formulations. *Journal of Computational Physics*, 79:12–49.
- [80] Guglielmo Padula and Gianluigi Rozza. Generative models for parameter space reduction applied to reduced order modelling. In *International Workshop of Scientific Machine Learning: Emerging Topics*, pages 133–158. Springer, 2024.
- [81] Guglielmo Padula, Artem Sinitsa, and Gianluigi Rozza. A generative model-free form deformation approach for the generation of mesh motions with applications to pde. *arXiv preprint arXiv:2607.13202*, 2026.
- [82] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- [83] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepscdf: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 165–174. IEEE, 2019.
- [84] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.

-
- [85] Benjamin Peherstorfer and Karen Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.
 - [86] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.
 - [87] Stephen B Pope et al. *Turbulent flows*, volume 20. Cambridge university press Cambridge, 2000.
 - [88] 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.
 - [89] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
 - [90] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*. Springer, 2015.
 - [91] Nestor V Queipo, Raphael T Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P Kevin Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28, 2005.
 - [92] Mats Ramnefors, Rikard Bensryd, Elna Holmberg, and Sven Perzon. Accuracy of drag predictions on cars using cfd-effect of grid refinement and turbulence models. In *International Congress & Exposition 2072*, 1996.
 - [93] A. Rao, G. Minelli, B. Basara, and S. Krajnovic. On the two flow states in the wake of a hatchback ahmed body. *Journal of Wind Engineering and Industrial Aerodynamics*, 173:262–278.
 - [94] J Reiss, P Schulze, J Sesterhenn, and V Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40:A1322–A1344.
 - [95] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017.

- [96] Francesco Romor, Giovanni Stabile, and Gianluigi Rozza. Non-linear manifold reduced-order models with convolutional autoencoders and reduced over-collocation method. *Journal of Scientific Computing*, 94(3):74, 2023.
- [97] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [98] Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of computational methods in engineering*, 15(3):1–47, 2007.
- [99] Gianluigi Rozza, Giovanni Stabile, and Francesco Ballarin. *Advanced reduced order methods and applications in computational fluid dynamics*. SIAM, 2022.
- [100] Shenhui Ruan, Andreas G Class, and Gianluigi Rozza. A structured review of reduced order modeling for domain decomposition problems: State of the art and perspectives. *Archives of Computational Methods in Engineering*, pages 1–73, 2026.
- [101] Mario Rüttgers, Junshin Park, and Donghyun You. Large-eddy simulation of turbulent flow over the driver fastback vehicle model. *Journal of Wind Engineering and Industrial Aerodynamics*, 186:123–138, 2019.
- [102] Filippo Salmoiraghi, Angela Scardigli, Haysam Telib, and Gianluigi Rozza. Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation. *International Journal of Computational Fluid Dynamics*, 32(4-5):233–247, 2018.
- [103] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
- [104] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986.
- [105] Louis Serrano, Lise Le Boudec, Armand Kassaï Koupai, Thomas X Wang, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. Operator learning with neural fields: Tackling pdes on general geometries. *Advances in Neural Information Processing Systems*, 36:70581–70611, 2023.

-
- [106] Mikhail L Shur, Philippe R Spalart, Mikhail Kh Strelets, and Andrey K Travin. A hybrid rans–les approach with delayed-des and wall-modelled les capabilities. *International Journal of Heat and Fluid Flow*, 29:1638–1649.
- [107] Ashish Singh, Santhosh Kumar, and Kishor Nikam. High performance cfd computations for ground vehicle aerodynamics. *SIAT 2011 203781*, 2011.
- [108] Lawrence Sirovich et al. Turbulence and the dynamics of coherent structures. part i: Coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.
- [109] Joseph Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3):99–164, 1963.
- [110] P R Spalart, S Deck, M L Shur, K D Squires, M Kh Strelets, and A Travin. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and Computational Fluid Dynamics*, 20:181–195.
- [111] Philippe Spalart and Steven Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.
- [112] Philippe R Spalart. Detached-eddy simulation. *Annual review of fluid mechanics*, 41(1):181–202, 2009.
- [113] Giovanni Stabile and Gianluigi Rozza. Finite volume pod-galerkin stabilised reduced order methods for the parametrised incompressible navier–stokes equations. *Computers & Fluids*, 173:273–284, 2018.
- [114] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133, 1974.
- [115] Tommaso Taddei. A registration method for model order reduction: Data compression and geometry reduction. *SIAM Journal on Scientific Computing*, 42:A997–A1027.
- [116] 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.
- [117] Nobuyuki Umetani and Bernd Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics (TOG)*, 37(4):1–10, 2018.

- [118] Henk Kaarle Versteeg and Weeratunge Malalasekera. An introduction to computational fluid dynamics. 1995.
- [119] Ricardo Vinuesa and Steven L Brunton. Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2(6):358–366, 2022.
- [120] Thomas Walker, Salvatore Esposito, Daniel Rebain, Amir Vaxman, Arno Onken, Changjian Li, and Oisin Mac Aodha. Crosssdf: 3d reconstruction of thin structures from cross-sections. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30928–30937. IEEE, 2025.
- [121] Shibo Wang, Terence Avadiar, Mark C Thompson, and David Burton. Effect of moving ground on the aerodynamics of a generic automotive model: The driver-estate. *Journal of Wind Engineering and Industrial Aerodynamics*, 195:104000, 2019.
- [122] Yifei Wang, Christophe Sicot, Jacques Boree, and Mathieu Grandemange. Experimental study of wheel-vehicle aerodynamic interactions. *Journal of Wind Engineering and Industrial Aerodynamics*, 198:104062, 2020.
- [123] 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.
- [124] Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.
- [125] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.
- [126] Ding-Xuan Zhou. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124:319–327, 2020.