

Article

The Development of a Data-Driven Surrogate Model for Enhancing Electric Vehicle Cabin Airflow Analysis

Mirza Popovac ^{*}, Thomas Bäuml , Dominik Dvorak  and Dragan Šimić

Electric Vehicle Technologies, Center for Transport Technologies, AIT—Austrian Institute of Technology, Giefinggasse 2, 1210 Vienna, Austria

* Correspondence: mirza.popovac@ait.ac.at

Abstract

This paper presents a data-driven surrogate model for predicting cabin airflow and its integration into system-level electric vehicle simulations for energy management analysis. The model employs a graph-based neural network with a mirror-symmetric predictor–corrector architecture and is trained on a dataset generated using computational fluid dynamics (CFD) covering a defined range of inlet velocities and temperatures. The surrogate appropriately reconstructs temperature fields and captures the dominant airflow structures at significantly lower computational cost than CFD. Quantitative evaluation shows high accuracy in passenger-relevant regions, while localized discrepancies remain confined mainly to shear-layer zones. The model enables near-real-time inference and is coupled with a system-level modeling framework for control-oriented simulations that are impractical with CFD. The study is tailored to a specific geometry and operating range, showing that targeted training strategies and physics-based extensions improve robustness, particularly under limited data conditions.

Keywords: Indoor Comfort; OpenFOAM; CFD; graph neural networks; PyTorch

1. Introduction

The energy efficiency of thermal conditioning systems in electric vehicles (EVs) is one of the main factors affecting overall vehicle performance and user acceptance [1,2]. Unlike conventional vehicles with internal combustion engines that exploit waste heat for cabin conditioning, EVs rely entirely on battery energy for both traction and auxiliary loads. Consequently, thermal management has a direct impact on vehicle range, operating cost, and passenger comfort [3]. Among these auxiliary loads, the heating, ventilation, and air conditioning (HVAC) system is a major contributor to energy consumption, especially under extreme ambient conditions.

A key challenge in simulating EV thermal performance is the lack of fast, system-compatible models that provide detailed cabin airflow information [4,5]. Passenger thermal comfort depends strongly on local air velocities and temperatures within the cabin. However, resolving these quantities with high fidelity typically requires computational fluid dynamics (CFD) simulations. While CFD delivers accurate flow and temperature fields, its computational cost makes it impractical for vehicle-level simulations, optimization loops, or control-oriented frameworks where repeated or real-time evaluations are needed.

This limitation motivates the development of surrogate models that reproduce CFD-level cabin airflow predictions at a fraction of the computational cost. Such models are particularly valuable when integrated into system-level EV simulations, where cabin



Academic Editor: Tim Persoons

Received: 20 March 2026

Revised: 20 April 2026

Accepted: 23 April 2026

Published: 25 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

airflow information supports HVAC control strategies, passenger comfort assessment, energy management decisions, and overall vehicle energy optimization.

In this work, a data-driven surrogate model that employs scientific machine learning methods is presented to predict cabin airflow in EVs. High-fidelity CFD solutions are used as training data for learning velocity and temperature fields inside the passenger compartment. The CFD simulations are performed with the widely used open-source library suite *OpenFOAM* [6], and the obtained data are used to train a convolutional neural network in *Python* [7]. The *PyTorch* library [8] is used for implementing a mirror-symmetric network inspired by the U-Net architecture [9]. Specifically, the U-Net structure was first introduced for biomedical image segmentation, as particularly suitable for learning spatially resolved fields owing to its multiscale feature extraction and skip-connection design. In the present context, the adopted design aims to reproduce dominant cabin flow structures and temperature distributions while maintaining low inference times and modest computational requirements. The present work does not aim to introduce fundamentally new machine learning methodologies. Instead, its contribution lies in the integration of established approaches into a coherent framework tailored for system-level EV simulations. In particular, the following aspects constitute the main contributions of this study: (i) development of a graph-based surrogate model operating directly on unstructured CFD meshes, (ii) implementation of a predictor–corrector training strategy for improved convergence behavior, and (iii) establishment of a complete workflow from CFD data generation to surrogate deployment, including the integration of the surrogate model into a system-level simulation environment.

Previous studies have demonstrated that artificial neural networks (ANN) can efficiently approximate flow fields under domain-specific constraints, particularly for steady-state and laminar regimes on structured grids. In that view, [10] employs convolutional neural networks to predict steady laminar flows with high efficiency, highlighting the potential of data-driven models for accelerating CFD tasks. Similarly, [11] extends surrogate modeling approaches to both steady and unsteady flows, showing that machine-learned models can capture dominant flow dynamics while significantly reducing computational cost. A broader overview is provided by [12] that reviews recent advances in machine learning-based CFD surrogates, emphasizing their applicability in the built environment and identifying limitations related to generalization, data requirements, and physical consistency. In parallel, developments in geometric deep learning have enabled the extension of such approaches beyond structured grids toward unstructured and mesh-based representations. Provided by [13] is a comprehensive framework for learning on non-Euclidean domains, establishing the theoretical foundation for graph-based representations of physical systems. Furthermore, [14] demonstrates that graph networks can learn to simulate complex physical dynamics governed by partial differential equations, accurately capturing interactions across discretized domains. Finally, [15] extends this concept to mesh-based simulations, showing that message-passing architectures can directly operate on computational meshes and generalize across varying geometries and resolutions. These approaches highlight the capability of graph-based models to preserve spatial connectivity and numerical structure, making them very suitable for CFD-derived datasets.

Building on these foundations, the present work investigates surrogate modeling for complex three-dimensional cabin geometries defined on unstructured meshes, where flow features such as jet development, recirculation, and buoyancy-driven stratification must be captured simultaneously. Complementing existing approaches, the proposed framework focuses on a problem-specific yet practically relevant setting, developing a surrogate model tailored to a defined cabin geometry and operating range. The primary objective is to bridge the gap between high-fidelity CFD-based airflow modeling and system-level

EV simulations, where computational efficiency and real-time capability are essential (e.g., energy optimization, design-phase studies, control-oriented applications).

To concentrate on the essential aspects of surrogate model development and assessment, a simplified electric truck cabin geometry is employed as the reference test case (Figure 1). This simplification allows systematic exploration of model accuracy and robustness without excessive focus on geometrical complexity, while maintaining all relevant flow features and required surrogate model complexity. Nevertheless, the proposed methodology remains extensible to passenger cabins of other EV configurations, including electric minibuses, trucks, or passenger cars.

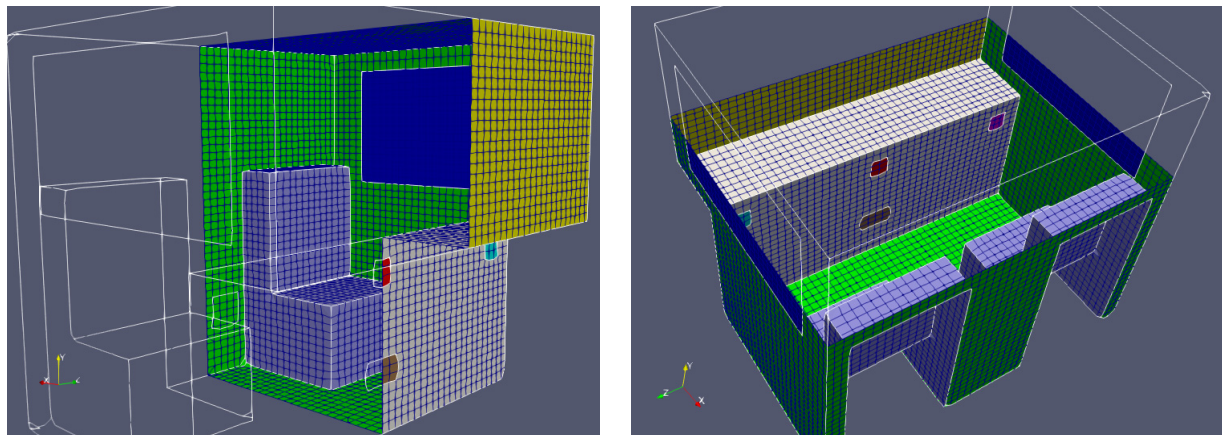


Figure 1. EV passenger cabin geometry and numerical mesh used for the present study (left: cut view to the side, right: cut view from the top) with color scheme denoting different boundaries.

An important aspect of the presented approach is the integration of the developed surrogate model into a system-level EV simulation environment. The respective EV model is implemented in *Modelica/Dymola* [16,17], which provides a physics-based, equation-oriented environment for modeling vehicle subsystems such as the HVAC, battery, and vehicle dynamics. In such environments, HVAC operation is typically governed by control strategies (e.g., PID controllers) coupled with multiple subsystems, including heat exchangers and ventilation units. Executing CFD simulations within this loop is computationally infeasible, making reduced-order or surrogate representations essential for practical implementation. By embedding the surrogate model within this system-level simulation, cabin airflow information becomes available on demand during energy management and thermal comfort analyses. This capability enables evaluation and optimization of HVAC control strategies while explicitly accounting for passenger comfort, without the burden of repeated CFD calculations. Therefore, this surrogate model is trained for a specific cabin configuration and operating range: the intention is not to construct a universally generalizable model, but rather to establish a reproducible workflow that can be applied to other configurations by retraining on corresponding CFD datasets.

The remainder of this paper is organized as follows: Section 2 describes the CFD dataset generation with *OpenFOAM* (geometry definition, numerical setup, data preparation, characteristic cabin flow patterns); Section 3 presents the surrogate model development (neural network architecture, input–output formulation, training procedure); Section 4 evaluates the surrogate predictions (main airflow outputs and quantitative accuracy assessment); Section 5 describes the coupling of the Python-based surrogate with the *Modelica/Dymola* EV system model (communication protocol and performance considerations); Section 6 summarizes the main conclusions and directions for future work.

2. Cabin Airflow Dataset

The dataset used for training and validating the presented data-driven surrogate model was generated from high-fidelity numerical simulations of airflow and heat transfer inside an EV passenger cabin. The primary objective of the CFD campaign is to provide physically consistent, spatially resolved velocity and temperature fields representative of typical cabin conditioning scenarios. These fields serve as reference ground-truth data for subsequent data-driven modeling.

2.1. Governing Equations

The cabin airflow is simulated as a steady-state, incompressible, turbulent flow of a Newtonian fluid with heat transfer. This flow is governed by the conservation equations for mass, momentum, and energy, which in the present case take the form [18]:

$$\begin{aligned}\nabla \cdot \mathbf{u} &= 0 \\ \rho \mathbf{u} \cdot \nabla \mathbf{u} &= \nabla \cdot [(\mu + \mu_t) \nabla \mathbf{u}] - \nabla p \\ \rho c_p \mathbf{u} \cdot \nabla T &= \nabla \cdot [(\mu / \text{Pr} + \mu_t / \text{Pr}_t) \nabla T]\end{aligned}\quad (1)$$

where $\mathbf{u}$, p , and T denote the velocity vector field, and pressure and temperature scalar fields, respectively; ρ is the air density; c_p is the specific heat capacity; μ and μ_t are the molecular and turbulent dynamic viscosities, while Pr and Pr_t are the corresponding Prandtl numbers.

To model the effects of turbulent fluctuations on the mean flow, the Reynolds-averaged Navier–Stokes (RANS) approach is employed, where turbulent viscosity μ_t is defined based on the additional turbulence scalars [19]. Here, the k – ε turbulence closure framework is used [20], which requires two turbulence scalar transport equations to be solved: for the turbulent kinetic energy k , and for its dissipation rate ε . The final set of turbulence equations in steady-state form reads

$$\begin{aligned}\mu_t &= \rho C_\mu k^2 / \varepsilon \\ \rho \mathbf{u} \cdot \nabla k &= \nabla \cdot [(\mu + \mu_t / \sigma_k) \nabla k] + P_k - \rho \varepsilon \\ \rho \mathbf{u} \cdot \nabla \varepsilon &= \nabla \cdot [(\mu + \mu_t / \sigma_\varepsilon) \nabla \varepsilon] + \varepsilon / k (C_1 P_k - \rho C_2 \varepsilon)\end{aligned}\quad (2)$$

where C_μ is the turbulent viscosity constant, P_k is the production of turbulent kinetic energy, σ_k and σ_ε are turbulent Prandtl numbers. Owing to its robustness, accuracy, and computational efficiency, for this work, the realizable k – ε model variant is selected [21], in which the model constants C_1 and C_2 are sensitized for wall-bounded shear and jet-dominated flows.

For the air properties, $c_p = 1005 \text{ J}/(\text{kg}\cdot\text{K})$ and $\mu = 1.8 \times 10^{-5} \text{ Pa}\cdot\text{s}$ were assumed constant. The density variation was evaluated using the ideal gas law, while the Prandtl analogy was used for the thermal conductivity with $\text{Pr} = 0.71$ and $\text{Pr}_t = 0.85$. These assumptions are standard for indoor airflow simulations and ensure consistency across the dataset.

2.2. CFD Setup

The simulations were performed using open-source CFD code *OpenFOAM*, where steady-state solutions were obtained with the solver *buoyantSimpleFoam*, which accounts for buoyancy effects and employs the SIMPLE algorithm for pressure–velocity coupling [22].

The computational mesh was generated starting from a structured background grid created with *blockMesh*. The cabin geometry, including dashboard and seats, was introduced using *snappyHexMesh*. The final mesh (Figure 1) comprises 61,947 cells, intentionally

kept compact to enable a large parametric study while still capturing the dominant cabin flow structures relevant for passenger comfort.

The entire CFD setup was intentionally maintained within the *OpenFOAM* environment in order to simplify and accelerate dataset generation. In this context, the boundary surfaces were defined using *topoSet* and *createPatch* utilities, extracting patches for the cabin construction (in addition to the seats and dashboard), windows, and windshield to be treated as walls, as well as one centrally located air outlet patch, and three air inlet patches (left, center, and right). In addition to mesh resolution, standard quality metrics such as non-orthogonality and skewness were monitored to ensure numerical stability, with values remaining within recommended limits for reliable simulations (from the *checkMesh* utility: aspect ratio 3.09, skewness 2.06, non-orthogonality 7.42).

2.3. CFD Verification

The CFD results are treated as ground-truth data for the surrogate model, meaning that the learned mapping directly reflects the numerical solution of the governing equations under the chosen modeling assumptions. However, before using the CFD results as ground-truth for surrogate model training, the numerical setup underwent a dedicated verification step to assess the reliability and consistency of the simulations. This verification comprised two complementary analyses: a grid independence study and turbulence model sensitivity testing.

Mesh sensitivity was assessed by uniformly refining the computational grid in all three spatial directions. For each refinement level, velocity and temperature profiles were evaluated along the spanwise centerline of the cabin domain (Figure 2). With increasing mesh resolution, the profiles converged consistently. Beyond 60 background mesh divisions per characteristic direction in *blockMesh*, changes in both velocity and temperature became negligible, with profiles tending toward a single curve. Consequently, the mesh resolution corresponding to approximately 60 background mesh divisions was considered sufficient to ensure grid-independent results, and this resolution was adopted for the entire simulation campaign.

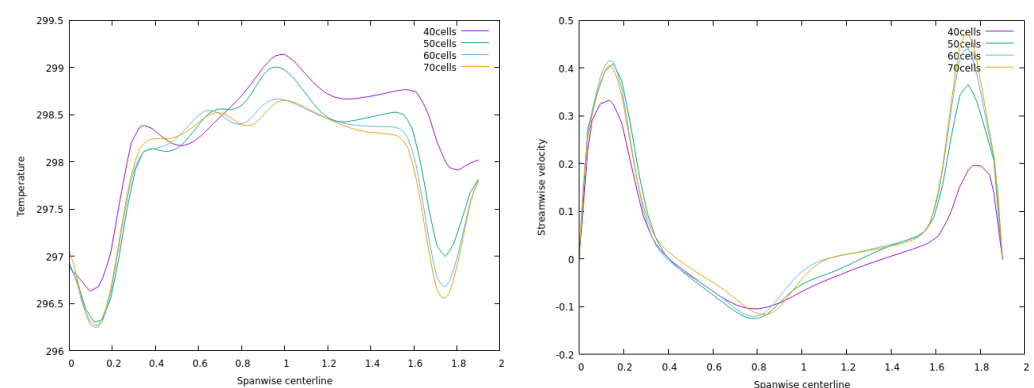


Figure 2. The comparison of the flow characteristics for different grid refinement levels.

To evaluate the impact of turbulence modeling on predicted cabin airflow, the realizable $k-\epsilon$ simulations were compared with the $k-\omega$ SST model [23], since both are widely used for indoor and wall-bounded turbulent flows. Figure 3 illustrates the qualitative and quantitative comparison, showing that the spatial distributions of velocity and temperature obtained with the two models exhibit only minor quantitative differences, and that the overall flow topology and thermal stratification remain consistent. The agreement between grid-refined solutions and across turbulence models establishes a satisfactory level of numerical fidelity: differences between the realizable $k-\epsilon$ and $k-\omega$ SST models remained

within a few percent over representative cutplanes ($<5\%$), confirming consistency of the dataset. The CFD results are therefore regarded as a reliable reference dataset for training and evaluating the surrogate model.

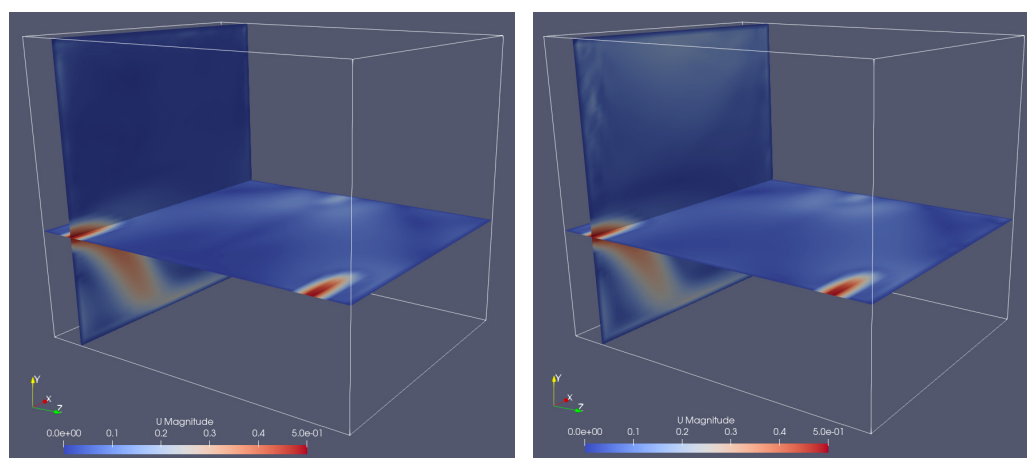


Figure 3. The airflow pattern over representative horizontal and vertical cutplanes, obtained with realizable $k\text{--}\epsilon$ (left) and $k\text{--}\omega$ SST (right) turbulence models.

2.4. Boundary Conditions

At the air inlets, the fixed value (Dirichlet) boundary condition was imposed for the velocity, temperature, and turbulence quantities, while for the pressure, the zero-gradient (Neumann) boundary condition was applied. Conversely, at the outlet, the pressure was prescribed as Dirichlet and Neumann for all other variables. All solid surfaces were treated as no-slip walls with adiabatic thermal boundary conditions.

Only the left inlet acted as the main control for the operating regime, while the values at the right and center inlets were kept constant. To represent typical cabin conditioning scenarios, the values of velocity and temperature at the left inlet (U_{in} and T_{in} , respectively) were systematically varied over a specified range, and Table 1 summarizes the applied values. The goal was to analyze parameter combinations defining two characteristic operating regimes: one with low blowing intensity and heating level (referred to as the L-case), and one with high blowing intensity and heating level (referred to as the H-case).

Table 1. Imposed velocity and temperature inlet values.

	Right	Center	Left
U_{in} (m/s)	0.7	0.7	0.5 0.7 0.9 1.1 1.3 1.5 1.7 1.9 2.1 2.3 2.5 2.7
T_{in} (K)	290	290	285 288 291 294 297 301 304 307 310 313 316

The parameter space defined by Table 1 represents the intended operating envelope for the surrogate model. The dataset is therefore designed to ensure sufficient coverage within this range rather than to span all possible HVAC operating conditions. The imposed boundary condition values formed the input parameter set for the main simulation campaign, executed through an automated script generating a dataset comprising 132 simulations. Running in the decomposed mode, the total simulation time amounted to 18.2 h.

2.5. Cabin Flow Characteristics

Representative results for the L-case and H-case are shown in Figure 4. The velocity fields (Figure 4, top) reveal jet-dominated flows issuing from the inlets, traversing the seating region, and recirculating toward the outlet after impingement on the rear cabin wall.

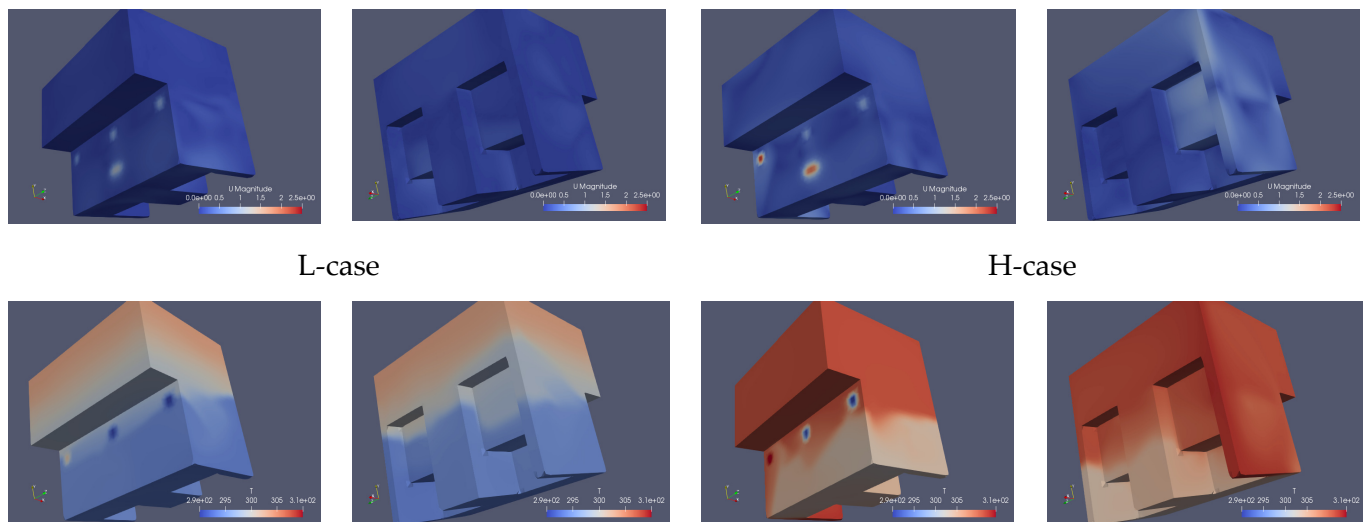


Figure 4. Front and back views of the velocity (**top**) and temperature (**bottom**) distributions, for low blowing intensity and heating level (L-case) and high blowing and heating level (H-case).

The temperature fields (Figure 4, bottom) exhibit a clear stratification characterized by a supply-air-dominated zone around the seats and an accumulation of warmer air near the cabin roof due to buoyancy effects. Capturing both airflow structure and temperature stratification is essential for assessing passenger thermal comfort and forms the basis of the surrogate model outputs.

3. Surrogate Model Architecture

The surrogate model is implemented in *Python* (v3.10.11) using *PyTorch* (v2.5.1). It is designed to provide spatially resolved predictions of cabin airflow at low computational cost, thereby enabling its deployment within system-level EV simulations. Because the CFD solution data are defined on an unstructured mesh, the surrogate adopts a graph-based domain representation. In this formulation, message passing between nodes is established through edges that reflect the cell connectivity of the CFD mesh. On this basis, a graph-based neural network composed of convolutional layers is constructed, organized as a mirror-symmetric feed-forward architecture with skip connections.

This model design preserves geometric adjacencies and supports multiscale feature extraction directly on the CFD mesh, avoiding the memory overhead and geometric distortion associated with voxelization. The approach is consistent with developments in geometric deep learning and learned mesh-based simulators, leveraging message-passing networks on graphs which approximate PDE operators and generalize across different geometries and resolutions [16–18]. The implementation further benefits from the research-oriented programming paradigm of *PyTorch* and mature optimization tools.

3.1. Model Structure and Implementation

The surrogate-modeling workflow consists of four stages: (i) dataset preparation, (ii) ANN definition, (iii) training and validation, and (iv) model storage and post-processing. For the dataset preparation, the *OpenFOAM* case is read directly from its file structure: mesh connectivity from *constant/polyMesh* is used to construct the computational graph; input fields are taken from the initial directory (0/); target fields from the converged time directories (1/, 2/, etc.). As a standard practice for stabilizing ANN training, per-channel statistics are computed to normalize input and target quantities (min–max scaling).

The model leverages a graph neural network (GNN) data representation [24] and is organized into skip-connected modules that incorporate hierarchical feature projection,

as illustrated in Figure 5. The first two sections (predictor) progressively aggregate multi-scale features through stacked graph convolutional network (GCN) blocks [25] and then reconstruct them at the original graph resolution, while lateral skip connections preserve high-frequency spatial detail. The final section (corrector) applies a linear projection to refine the four supervised targets (three velocity components and temperature). The training objective combines an MSE term applied to a hidden representation of the predictor, either with an L1 penalty on the final outputs or their cast into a physics constraint form. The resulting λ -weighted loss balance promotes a smooth latent space output and ensures sensitivity near convergence, while providing robustness to localized outliers and projecting physics-defined conditions, as applicable for CFD fields [26–28].

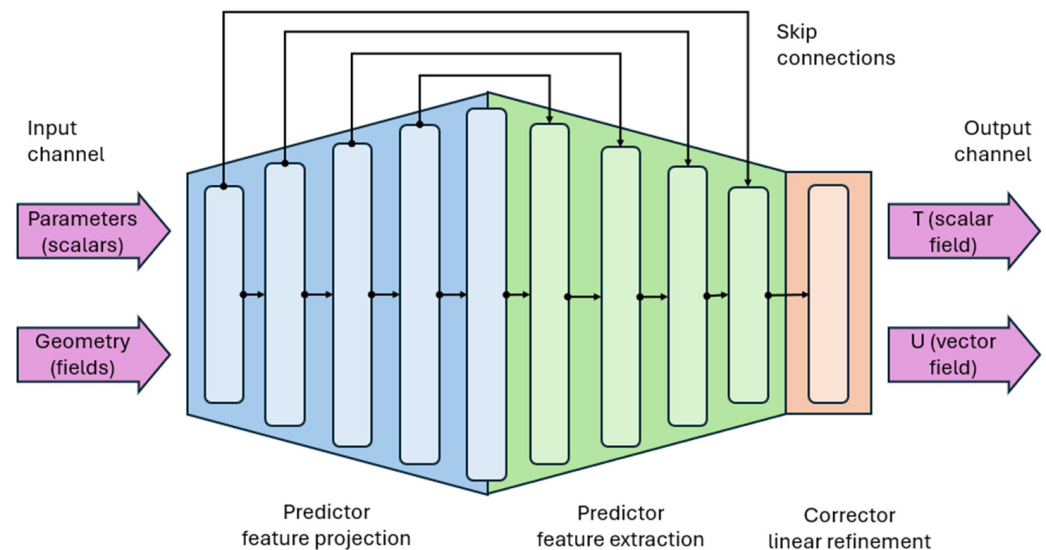


Figure 5. Predictor–corrector network architecture, with the predictor based on a mirror-symmetric design with skip connections between symmetric layers, and the corrector with a linear projection to the output fields.

In the training loop, the Adam optimizer is used, whose adaptive step-size mechanism is well-suited to the heterogeneous operating conditions encountered in the dataset [29]. An efficient *PyTorch* implementation of data-loading utilities and the model-serialization framework facilitates fast and streamlined experimentation and reproducibility. For the present configuration, the total training time was 8.1 h, which is modest relative to the computational cost of additional CFD simulations.

3.2. Input and Output Channels

Two categories of inputs are used for training the surrogate: boundary-condition parameters, which define the operating point (both L-case and H-case inlet velocities and temperatures, as listed in Table 1), and user-defined volume fields encoding the geometry and topology required to reconstruct spatial flow structures.

The operating point values T_{in} , U_{in} are embedded as global parameters, broadcast to all nodes, and concatenated with the first layer node features for learning the flow mapping across the L-case and H-case regimes. The geometric descriptors include cell-center coordinates and signed distance functions (SDFs) to selected boundaries (inlets, outlet, windows/windshield, and cabin construction with dashboard and seats), as shown in Figure 6. Coordinates are generated on-the-fly using the postProcess utility, whereas the checkMesh -writeFields "(wallDistance)" utility option is used to obtain the SDFs (it requires the wallDist subdictionary in fvSchemes to define the method for wall distance

calculation). Together, these geometric features provide a compact, mesh-aware description consistent with graph-based learning approaches used in physical simulation.

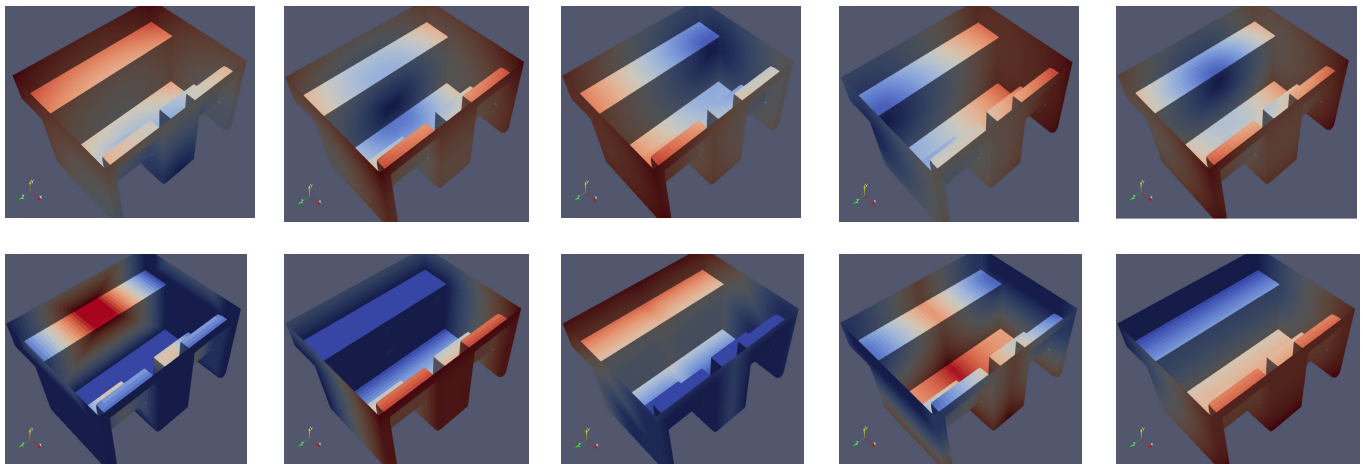


Figure 6. Surface representation of the input channel volume fields: cell-center coordinates, SDF with respect to the outlet, inlets (**left, right, center**), cabin construction, dashboard, seats, windows, and windshield (**shown from left to right, and top to bottom**).

Targets consist of the steady-state CFD solution fields: three components of the air-flow velocity vector field U and the temperature scalar field T , stored in the respective time directories. As a part of the dataset preparation, the targets from the training set are normalized (min–max scaler), and the training is performed using the normalized target values. During inference, the predicted values are renormalized (using the per-channel scaler defined at the training dataset preparation), and the physical units are recovered.

3.3. Model Architecture Rationale

The adopted predictor–correction design separates representation learning from output calibration, which improves optimization stability through residual-style refinement. This is known to improve the training in deep architectures, since regularization is applied at two complementary levels: an MSE term on the predictor’s hidden representation encourages a smooth, structured latent space, while an L1 penalty or physics-based loss on the final outputs enhances robustness to localized deviations without diminishing sensitivity to physically relevant gradients.

The choice of a palindromic feed-forward GNN with skip connections is motivated by two distinct considerations: mesh awareness and multiscale fidelity. For the first, graph message passing naturally leverages the CFD mesh connectivity, preserving geometric adjacency and local numerical stencils, which is an essential property when learning operators on unstructured discretizations. For the second, cabin flows simultaneously exhibit jet cores, recirculation zones, and buoyancy-driven stratification: the encoder–decoder architecture with skip connections maintains fine-scale detail while aggregating broader spatial context, a strength inherited from U-Net-inspired designs.

Compared with pure convolutional networks on voxelized grids, the graph-based approach retains geometric fidelity without rasterization artifacts and delivers more reliable predictions across different flow regimes. However, there is a trade-off relative to image-based workflow: computational cost grows with the number of graph edges, the processing pipeline is more complex, and the GNN architecture requires somewhat greater computational resources and longer training times (approximately 25% in this case). This is acceptable, though, given the application’s need for accuracy and geometry fidelity.

The surrogate provides mesh-aware predictions tailored to the underlying CFD geometry and reasonably reconstructs the multiscale flow features essential for comfort-relevant metrics. At the same time, its fast inference capability makes it suitable for seamless server–client coupling with *Modelica/Dymola*, thereby enabling efficient integration within system-level EV simulations. While the present framework does not enforce strict physical conservation laws in the predictor stage, auxiliary physics-based requirements can be promoted in the corrector step through a λ -weighted loss term. In the present case, these are defined from the discretized weak form of the Navier–Stokes equations (Equation (1)) calculated as the filter applied to the CFD ground-truth and ANN-predicted fields.

3.4. Training Strategy

The dataset used for training was systematically partitioned into training and validation subsets in order to ensure robust model development and evaluation. Out of the total 132 CFD samples, 120 cases were used for training and 12 for validation, selected to ensure representative coverage of the operating parameter space defined by inlet velocity and temperature ranges. It is noted that the available dataset is limited in size and, therefore, cannot support the broad generalization of the trained surrogate model. Nevertheless, presented below is a dedicated quantification study, conducted using the available samples to assess the influence of dataset size on the resulting predictive accuracy.

The training process follows a predictor–corrector strategy, in which the initial network (predictor) learns a coarse representation of the flow field, while a subsequent corrector stage refines the prediction. The loss formulation combines statistical weighting and physics-informed components. First, a weighted loss is applied to emphasize shear-layer regions, as they exhibit steep gradients typically underrepresented during standard training. This improves the ability of the predictor to resolve localized flow features. In the corrector step, an additional physics-based loss component is introduced by recasting the predicted fields into residuals of the Navier–Stokes equations and comparing them to residuals computed from the CFD data. The quantification of this supervised learning strategy, aimed at improving consistency with the governing equations, is given below.

The training history is shown in Figure 7, illustrating stable convergence of both loss and accuracy for training and validation datasets (Figure 7, left). A zoom-in of the early training phase (Figure 7, right) highlights the contribution of the corrector stage, which accelerates convergence during approximately the first third of the training process. As training progresses, the predictor and corrector outputs become increasingly aligned. The influence of dataset size was assessed by inferring the trained model on a reduced number of training samples. Three dataset levels were considered (Figure 8 left): full dataset (3), reduced by one sixth (2), and reduced by one third (1). The results show a consistent increase in predictive accuracy with increasing training set size, indicating that the increase in data availability would yield measurable improvements in model performance. In addition, the effect of including the physics-based loss term was investigated by progressively increasing its λ -weighting relative to the direct supervised prediction, taking 10% (1), 20% (2), and 40% (3) for the weighting factor (Figure 8, right). The inclusion of the Navier–Stokes residuals was found to improve predictive accuracy (particularly for reduced dataset sizes). However, the improvement depends on an appropriate weighting level that balances data-driven fitting and physics-based regularization.

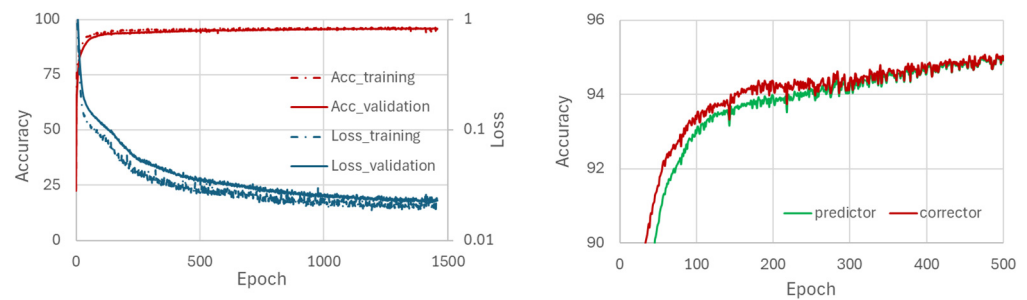


Figure 7. Model training history of the accuracy and loss for the training and validation set (**left**) and the starting phase zoom-in of the accuracy from the predictor and corrector (**right**).

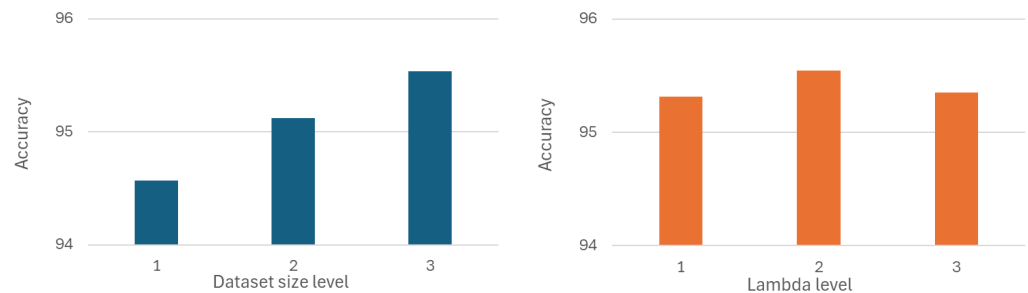


Figure 8. Model training accuracy with dataset size (**left**) and physics-based constraints (**right**).

4. Surrogate Model Predictions

The developed surrogate model is intended to enable rapid prediction of cabin air-flow velocity and temperature with an accuracy level suitable for energy-efficiency and thermal-management studies with system-level EV simulations. For a qualitative and quantitative comparison against high-fidelity CFD reference solutions, Figure 9 summarizes the temperature predictions, and Figure 10 shows velocity fields, both for representative high-intensity blowing and heating (H-case) and low blowing and heating (L-case) operating conditions. The achieved inference time of approximately 2.1 s per case confirms that the surrogate approaches near-real-time performance, making it suitable for on-the-fly use within coupled system simulations or optimization loops.

As illustrated in Figure 9, the surrogate model reproduces the principal characteristics of the steady-state temperature distribution with good qualitative agreement across both operating regimes. In both the H-case and L-case, a distinctly warmer air region forms near the cabin ceiling, driven by buoyancy effects, while the lower part of the cabin (particularly the seating area) remains dominated by the supply-air temperature. This vertical stratification, which is a key feature for passenger thermal comfort assessment, is consistently captured by the model. Differences between the CFD and surrogate predictions are limited primarily to the immediate vicinity of the air inlets, where strong temperature gradients exist.

The predicted velocity fields shown in Figure 10 demonstrate similar qualitative agreement with CFD in capturing the dominant flow topology. The surrogate successfully reproduces the acceleration of the inlet jets through the seating region and the subsequent redirection toward the outlet, as well as the presence of a relatively stagnant zone in the upper cabin region. These features are evident in both blowing regimes, although they are more pronounced in the H-case due to the higher inlet momentum. However, compared to temperature, localized deviations in velocity magnitude and jet spreading can be observed, particularly in regions of strong shear.

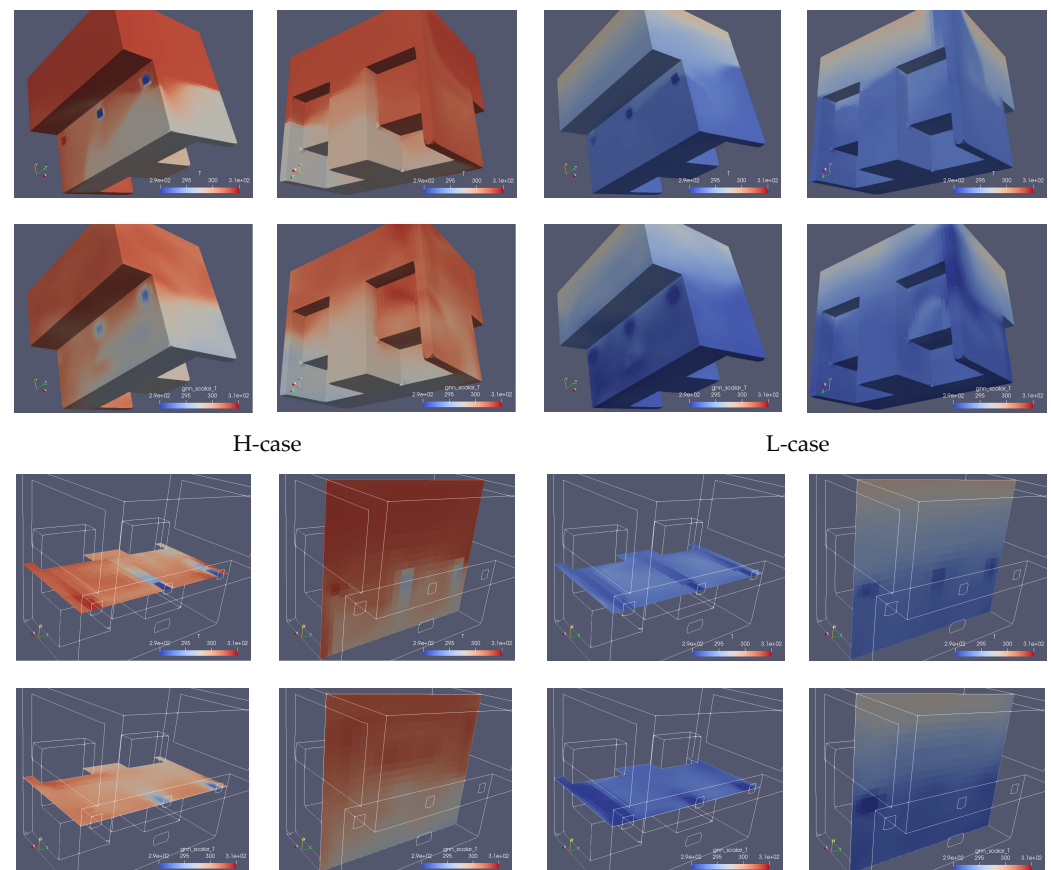


Figure 9. Cabin temperature distribution for a representative high blowing/heating case (**left**) and low blowing/heating case (**right**). **Above:** front and back views of the temperature predictions obtained with CFD (**top**) and surrogate model (**bottom**). **Below:** horizontal and vertical midplane temperature contour plots of predictions obtained with CFD (**top**) and surrogate model (**bottom**).

4.1. Interpretation of Prediction Errors

The overall discrepancies observed are acceptable from an application perspective, as system-level simulations rely primarily on spatially averaged quantities in passenger-relevant zones, where prediction accuracy exceeds 95%. Looking at the temperature distribution, the surrogate demonstrates high predictive accuracy, with volumetric mean temperature deviations remaining below 4% relative to CFD across the evaluated cases. This level of agreement is sufficient for most system-level and comfort-oriented analyses. However, the velocity magnitude displays comparatively larger discrepancies, with a mean relative error of approximately 17%. This behavior is reflected quantitatively by the higher relative error in velocity predictions. Notably, this error represents a cumulative measure over all three velocity components, further amplifying the discrepancy when compared to scalar temperature predictions. A practical consequence of the difference in predictive performance between temperature and velocity is the distinct physical role of these quantities: temperature behaves primarily as a passively transported scalar, governed by advection and diffusion, whereas velocity is the flow carrier and is far more sensitive to local flow structures and turbulence-induced mixing.

The qualitative differences between CFD and surrogate predictions become more apparent when examining local flow features. It can be observed in Figures 9 and 10 that the surrogate exhibits reduced accuracy in resolving jet shear layers, flow separation zones, and regions where turbulent mixing strongly influences jet spreading and entrainment. In CFD simulations, these phenomena are explicitly resolved through the solution of turbulence transport equations, which recover the relevant turbulent length and velocity

scales. In contrast, the surrogate model recovers these effects phenomenologically from the training data. While the adopted graph-based architecture effectively approximates large-scale flow patterns and stratification, it lacks an explicit representation of turbulence physics. Consequently, fine-scale structures (particularly those associated with sharp velocity gradients) tend to be smoothed, leading to underprediction or overprediction of local velocity magnitudes.

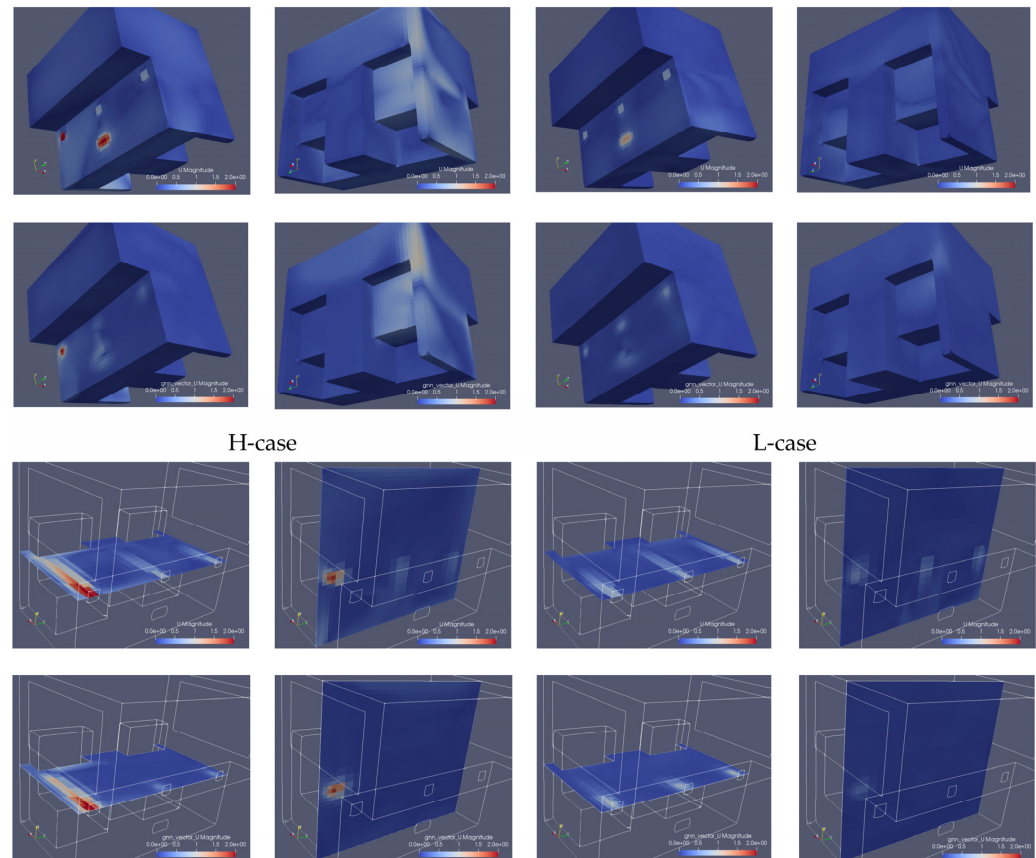


Figure 10. Cabin velocity distribution for a representative high blowing/heating case (**left**) and low blowing/heating case (**right**). **Above:** front and back views of the velocity predictions obtained with CFD (**top**) and surrogate model (**bottom**). **Below:** horizontal and vertical midplane velocity contour plots of the predictions obtained with CFD (**top**) and surrogate model (**bottom**).

4.2. Field-Based Accuracy

To analyze the prediction accuracy at the field level, a normalized accuracy measure is defined as follows:

$$\text{acc}_\phi = 1 - |\phi_{\text{CFD}} - \phi_{\text{surr}}| / |\max(\phi_{\text{CFD}}) - \min(\phi_{\text{CFD}})| \quad (3)$$

where ϕ denotes a generic flow variable (e.g., T or U). This metric evaluates the local deviation relative to the dynamic range of the CFD reference field.

Figure 11 presents the resulting accuracy distribution for temperature on both the horizontal and vertical midplanes. Across most of the cabin domain, the surrogate achieves local accuracies exceeding 95%, indicating that the dominant flow and thermal structures are well represented. Regions with significantly reduced accuracy (often falling below 90%) are confined to jet cores, shear layers, and near-inlet zones, where gradients are steepest, and the flow exhibits strong directional changes.

These regions coincide with elevated Z-score values (absolute Z-score greater than 3), identifying them as statistical outliers within the dataset. Such outliers predominantly

arise in localized, turbulence-dominated regions rather than in passenger-occupied zones, which are the primary interest for vehicle energy-management applications. As these regions correspond to strong shear layers and high-gradient zones, which act as statistical outliers within the dataset, they are inherently more difficult to approximate with purely data-driven models.

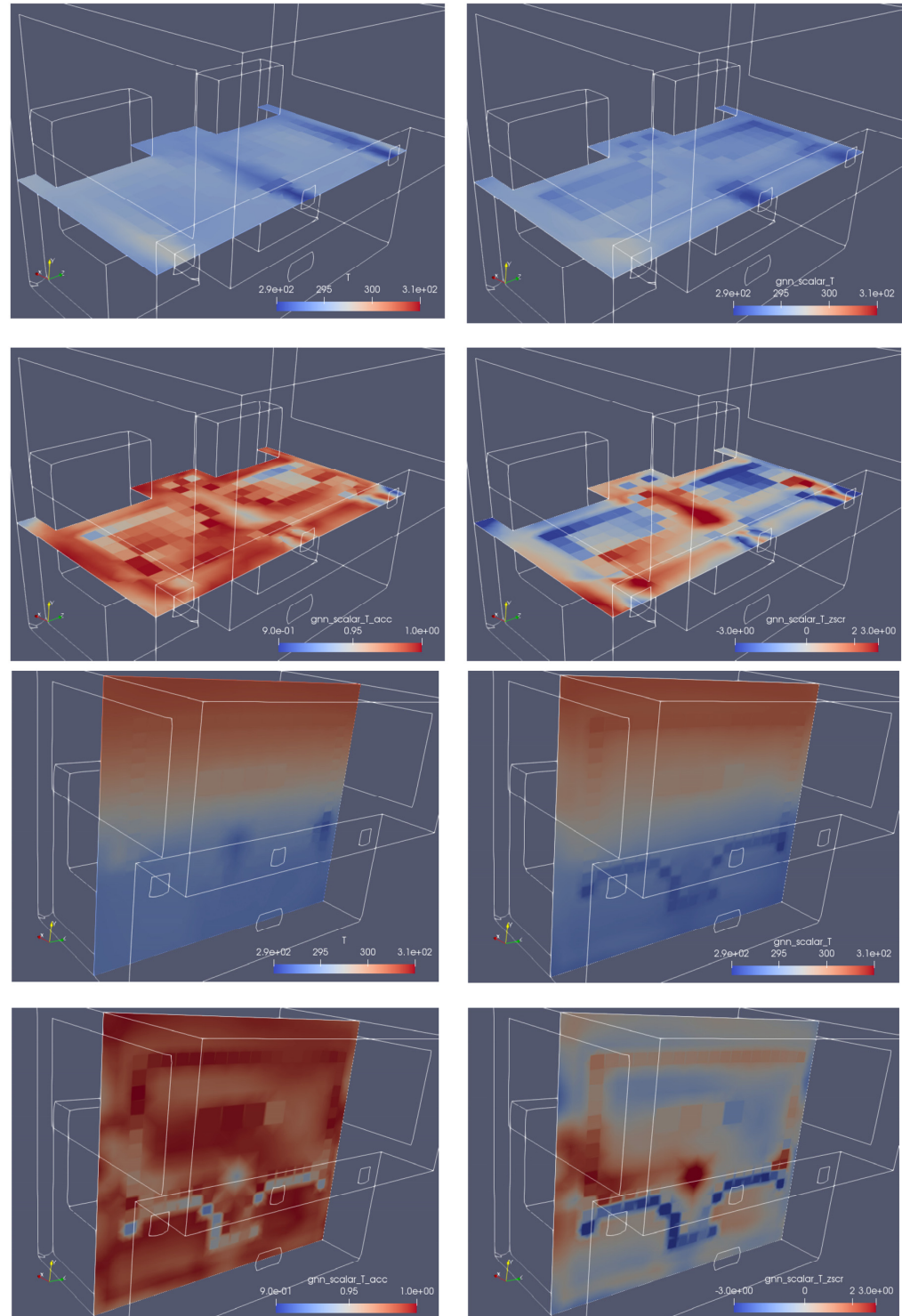


Figure 11. Temperature prediction from CFD and surrogate model (**top**), accuracy and Z-score (**bottom**). **Above:** horizontal midplane. **Bottom:** vertical midplane.

The presence of localized inaccuracies highlights a fundamental limitation of purely data-driven surrogates: without explicit knowledge of the governing Navier–Stokes equa-

tions or turbulence closure models, the neural network's extrapolation capability in high-gradient regions is limited. In the present framework, this limitation is partially mitigated through the predictor–correction architecture and the use of robust loss functions, which suppress the influence of extreme outliers and bias the model toward a physically consistent solution during training.

5. System-Level Integration

Rather than packaging the surrogate as a functional mock-up unit (FMU), which might cause version or license-related constraints, the present study employs a direct server–client communication architecture. This approach offers high flexibility and minimal overhead, enabling rapid experimentation with surrogate-model variants and input–output definitions. While the coupling via FMU remains a tool-independent alternative for surrogate-model integration, the server–client implementation used here emphasizes simplicity and transparency when embedding data-driven models into established system-simulation workflows. Notably, *Python* natively supports such communication through its built-in socket library, which is part of the standard *Python* distribution and facilitates straightforward and robust implementation of this coupling mechanism.

In the exemplary workflow illustrated in Figure 12, the surrogate model operates as an independent *Python* server process, for example, deployed on a remote compute cluster, where it hosts the trained neural network and processes incoming inference requests. The *Modelica*/*Dymola* simulation functions as a client potentially running on a separate local workstation, and communicates with the *Python* server through a predefined protocol during runtime. At prescribed simulation steps, *Dymola* transmits the current operating conditions to the *Python* server (U_{in} , T_{in}). The surrogate evaluates the request and returns the spatially averaged temperature and airflow velocity within the predefined passenger-occupied regions required for comfort-relevant metrics (U_{av} , T_{av}). The communication latency remains sufficiently low to permit the EV system-level simulation to progress without disrupting numerical stability or degrading overall simulation performance. This demonstrates that surrogate-based airflow prediction enables system-level simulations that would otherwise be infeasible using CFD due to computational constraints.

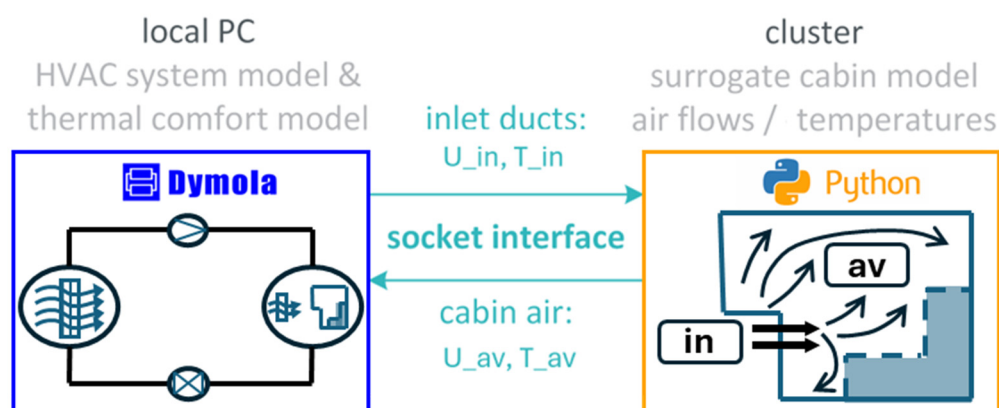


Figure 12. Schematic representation of the coupling between the system-level EV simulation (*Modelica*/*Dymola*) and the surrogate model (*Python*).

6. Discussion and Conclusions

This work presents the development, verification, and application of a data-driven surrogate model for predicting cabin airflow in electric vehicles, with the explicit objective of bridging high-fidelity CFD simulations and system-level thermal-management studies. The study is intentionally focused on a defined cabin geometry and operating

range, reflecting a practical engineering application rather than a universally generalizable modeling framework. The surrogate deploys a graph neural network structured into a mirror-symmetric architecture with skip connections, enabling direct learning from CFD data defined on unstructured meshes while preserving multiscale spatial information. The model was trained on a systematically generated CFD dataset covering a representative range of inlet velocities and temperatures, corresponding to low and high blowing/heating operating regimes.

The conducted analysis demonstrates that the surrogate model is capable of reproducing the dominant airflow topology and thermal stratification observed in CFD simulations with high fidelity. Across different operating flow regimes, the model reliably captures the formation of buoyancy-driven warm air layers near the cabin ceiling, the supply-air-dominated region around the seating area, and the principal jet structures linking inlets and outlets. Quantitative evaluation shows scenario-dependent accuracy, with local temperature deviations remaining below 4% in passenger-occupied zones, while local velocity magnitude predictions exhibit larger discrepancies, reaching a mean relative error of approximately 17%. These results are consistent with the physical roles of temperature as a passively transported scalar and velocity as the turbulence-carrying field, which is more sensitive to local shear layers and high-gradient regions.

A detailed field-based accuracy analysis further revealed that prediction errors are spatially localized, primarily in jet cores, shear layers, and near-inlet regions, where turbulent mixing and momentum exchange dominate the flow physics. In contrast, the majority of the cabin volume (particularly the regions most relevant for passenger comfort) exhibits prediction accuracy exceeding 95%. This distinction is an important finding of the study, as it confirms that the surrogate's limitations are largely confined to localized, turbulence-dominated features rather than global comfort-relevant indicators.

From a practical standpoint, one of the most significant outcomes of this work is the achieved near-real-time inference capability. With prediction times on the order of a few seconds per case, the surrogate enables rapid access to cabin airflow characteristics without executing full CFD simulations. This performance makes the model suitable for iterative design studies, parametric analyses, and embedded use within vehicle-level simulations. In particular, the direct server–client coupling between the Python-based surrogate and *Modelica*/*Dymola* system models allows system-level simulations to request airflow information on demand and proceed with energy-management calculations without sacrificing computational efficiency.

In an application context, the presented framework represents a pragmatic compromise between accuracy and computational efficiency. For EV thermal-management optimization and control, where airflow predictions are primarily used to assess passenger comfort and energy consumption trends rather than to resolve fine-scale turbulence, the demonstrated accuracy is sufficient and operationally valuable. The methodology therefore supports the broader goal of improving energy efficiency and driving range in electric vehicles by enabling tighter coupling between cabin comfort models and vehicle energy-management systems.

At the same time, the results highlight inherent limitations of purely data-driven surrogates. The reduced accuracy in highly turbulent regions underscores the absence of explicit physical constraints, such as conservation laws or turbulence closures, within the current learning framework. Although the adopted predictor–correction head and robust loss formulation mitigate the influence of outliers, the model fundamentally operates as an interpolator within the space covered by the training data. A preliminary investigation of physics-based loss formulations indicates that incorporating governing-equation residuals can improve prediction robustness, particularly in data-scarce scenarios.

The gained observations motivate the future work aimed at enhancing predictive fidelity through physics-defined constraints, hybrid modeling strategies, and domain decomposition approaches that treat jet and shear-layer regions with specialized subnetworks. The focus is to be on extending the present framework through systematic incorporation and improved treatment of physics-based considerations, and evaluation of alternative architectures. In addition, further studies are to investigate generalization across different geometries and expanded operating conditions, building upon the workflow established in this study.

Author Contributions: Conceptualization, M.P.; methodology, M.P.; software, M.P. and D.D.; validation, M.P., T.B., and D.D.; formal analysis, M.P.; investigation, M.P.; resources, M.P., T.B., D.D., and D.Š.; data curation, M.P.; writing—original draft preparation, M.P.; writing—review and editing, M.P., T.B., D.D., and D.Š.; visualization, M.P. and D.D.; supervision, D.Š.; project administration, D.Š.; funding acquisition, T.B. and D.Š. All authors have read and agreed to the published version of the manuscript.

Funding: This work was performed within the project Shift2Zero, which received funding through the EU Call HORIZON-CL5-2024-D5-01-06, under the Grant Agreement No. 101192375.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and analyzed during the current study can be made available upon request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following symbols and abbreviations are used in this manuscript:

Symbols

T	Temperature [K]
p	Pressure [Pa]
u	Velocity vector [m/s]
ϕ	Generic flow variable [-]
c_p	Specific heat at constant pressure [J/(kg·K)]
Pr/Pr_t	Molecular/Turbulent Prandtl number [-]
μ/μ_t	Molecular/Turbulent dynamic viscosity
ρ	Fluid density [kg/m ³]

Abbreviations

acc	Accuracy
CFD	Computational fluid dynamics
EV	Electric vehicle
FMU	Functional mock-up unit
GNN	Graph neural network
GCN	Graph convolution network
HVAC	Heating, ventilation, and air conditioning
U-Net	U-shaped neural network

References

1. Guo, R.; Li, L.; Sun, Z.; Xue, X. An integrated thermal management strategy for cabin and battery heating in range-extended electric vehicles under low-temperature conditions. *Appl. Therm. Eng.* **2023**, *228*, 120502. [[CrossRef](https://doi.org/10.3390/fluids11050107)]

2. Lajunen, A.; Yang, Y.; Emadi, A. Review of Cabin Thermal Management for Electrified Passenger Vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 6025–6040. [\[CrossRef\]](#)
3. Lajunen, A. Energy Efficiency and Performance of Cabin Thermal Management in Electric Vehicles. In *SAE Technical Paper 2017-01-0192*; SAE International Technical: Warrendale, PA, USA, 2017. [\[CrossRef\]](#)
4. Zhao, L.; Zhou, Q.; Wang, Z. A systematic review on modelling the thermal environment of vehicle cabins. *Appl. Therm. Eng.* **2024**, *257*, 124142. [\[CrossRef\]](#)
5. Dan, D.; Zhao, Y.; Wei, M.; Wang, X. Review of Thermal Management Technology for Electric Vehicles. *Energies* **2023**, *16*, 4693. [\[CrossRef\]](#)
6. Weller, H.G.; Tabor, G.; Jasak, H.; Fureby, C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* **1998**, *12*, 620–631. [\[CrossRef\]](#)
7. Python Software Foundation. Python Language Reference Manual (Version 3.10). 2021. Available online: <https://docs.python.org/3.10/reference/> (accessed on 20 October 2024).
8. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
9. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9351, pp. 234–241. [\[CrossRef\]](#)
10. Ribeiro, M.D.; Rehman, A.; Ahmed, S.; Dengel, A. DeepCFD: Efficient steady-state laminar flow approximation with deep convolutional neural networks. *arXiv* **2020**, arXiv:2004.08826. [\[CrossRef\]](#)
11. Bhushan, S.; Burgreen, G.W.; Bowman, J.L.; Dettwiller, I.D.; Brewer, W. Predictions of steady and unsteady flows using machine-learned surrogate models. *arXiv* **2023**, arXiv:2305.03823. [\[CrossRef\]](#)
12. Mao, R.; Lan, Y.; Liang, L.; Yu, T.; Mu, M.; Leng, W.; Long, Z. Rapid CFD prediction based on machine learning surrogate model in built environment: A review. *Fluids* **2025**, *10*, 193. [\[CrossRef\]](#)
13. Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [\[CrossRef\]](#)
14. Sanchez-Gonzalez, A.; Godwin, J.; Pfaff, T.; Ying, R.; Leskovec, J.; Battaglia, P. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning, Online, 13–18 July 2020*; Proceedings of Machine Learning Research (PMLR): Cambridge, MA, USA, 2020; Volume 119, pp. 8459–8468. Available online: <https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html> (accessed on 22 April 2026).
15. Pfaff, T.; Fortunato, M.; Sanchez-Gonzalez, A.; Battaglia, P. Learning mesh-based simulation with graph networks. *arXiv* **2021**. [\[CrossRef\]](#)
16. Modelica Association. Modelica Language Specification, Version 3.6 (9 March 2023). 2023. Available online: <https://specification.modelica.org/maint/3.6/MLS.pdf> (accessed on 22 April 2026).
17. Dassault Systèmes. *Dymola 2025x Release: Multi-Engineering Modeling and Simulation (Modelica/FMI; Python Scripting Support)*; Dassault Systèmes: Waltham, MA, USA, 2024. Available online: https://www.3ds.com/assets/invest/2024-12/dymola_2025x.pdf (accessed on 22 April 2026).
18. Batchelor, G.K. *An Introduction to Fluid Dynamics*; Cambridge University Press: Cambridge, UK, 2000. [\[CrossRef\]](#)
19. Pope, S.B. *Turbulent Flows*; Cambridge University Press: Cambridge, UK, 2000. [\[CrossRef\]](#)
20. Launder, B.E.; Spalding, D.B. The numerical computation of turbulent flows. *Comput. Methods Appl. Mech. Eng.* **1974**, *3*, 269–289. [\[CrossRef\]](#)
21. Shih, T.-H.; Liou, W.W.; Shabbir, A.; Yang, Z.; Zhu, J. A new $k-\epsilon$ eddy viscosity model for high Reynolds number turbulent flows: Model development and validation. *Comput. Fluids* **1995**, *24*, 227–238. [\[CrossRef\]](#)
22. OpenCFD Ltd. *BuoyantSimpleFoam Solver Documentation* (OpenFOAM v2112). 2021. Available online: <https://doc.openfoam.com/2306/tools/processing/solvers/rtm/heat-transfer/buoyantSimpleFoam/> (accessed on 22 April 2026).
23. Menter, F.R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA J.* **1994**, *32*, 1598–1605. [\[CrossRef\]](#)
24. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; pp. 1263–1272.
25. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
26. Huber, P.J. Robust estimation of a location parameter. *Ann. Math. Stat.* **1964**, *35*, 73–101. [\[CrossRef\]](#)
27. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [\[CrossRef\]](#)

28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
29. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.

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