Physics-Based Region Clustering to Boost Inference on Computational Fluid Dynamics Flow Fields

Riccardo Margheritti[1] (✉), Onofrio Semeraro[2], Maurizio Quadrio[3], and Giacomo Boracchi[4]

[1] DEIB, Politecnico di Milano, Milano 20133, Italy
riccardo.margheritti@polimi.it
[2] Lab. Interdisciplinaire des Sciences du Numérique (LISN), CNRS, Univ. Paris-Saclay, Orsay 91400, France onofrio.semeraro@universite-paris-saclay.fr
[3] DAER, Politecnico di Milano, Milano 20133, Italy maurizio.quadrio@polimi.it
[4] DEIB, Politecnico di Milano, Milano 20133, Italy giacomo.boracchi@polimi.it

Abstract. The high dimensionality and variability of Computational Fluid Dynamics (CFD) data pose a significant challenge for Machine Learning (ML) models. The only solutions in the literature addressing inference from CFD flow fields are based on expert-driven features, which consist of fluid dynamic quantities averaged on specific regions of the entire computational domain. However, using handcrafted features can limit the scalability and portability of existing methods, and result in the loss of critical flow field information that might be essential for capturing non-linear patterns inherent in the CFD data. We propose a method to replace handcrafted features with features defined on regions obtained by clustering. Our approach combines: *i*) physics-based clustering, to identify meaningful regions within the flow field, *ii*) cluster-based feature extraction, to capture localized fluid dynamics properties, and *iii*) set-learning models to process the extracted information. Our solution allows integrating physics-based modeling with ML, and provides a portable and flexible pipeline capable of effectively dealing with the variability and dimensionality of CFD flow fields. We validate our method on publicly available CFD datasets (from the aerospace domain) and apply it to a realistic scenario, that is, the classification of pathologies in real 3D human upper airways extracted from CT scans, acquired in collaboration with a medical hospital. Experimental results demonstrate the accuracy and scalability of our method, and highlight its potential for leveraging CFD data in ML frameworks for other scientific and engineering applications.

Keywords: Computational Fluid Dynamics · Machine Learning · Physics-Based Clustering · Set Learning · Features Extraction.

1 Introduction

Computational Fluid Dynamics (CFD), i.e., solving the numerical version of differential equations of the fluid motion, plays a crucial role in a large number of applications [1]. By solving the Navier–Stokes equations over a discretized domain, CFD provides detailed information on velocity, pressure, and several other flow variables, simulating real-world problems in a wide spectrum of fields, from aerodynamics [2] and weather prediction [3] to biomedical engineering (e.g., airflow analysis in respiratory diseases), and energy applications (e.g., wind farms and turbomachinery). From CFD, experts can evaluate performance, optimize designs, and gain a deeper understanding of fluid behavior in complex systems. The high dimensionality of CFD outputs and the computational cost of simulations, however, pose significant challenges for data analysis through Machine Learning (ML). Gathering annotated CFD datasets, in fact, is particularly difficult, as generating labeled data often requires running expensive simulations and relying on expert domain knowledge. Moreover, the size and complexity of CFD flow fields, which often involve millions of spatial points and tens of flow variables per point, make it impractical to directly train ML models on raw CFD data. This underscores the need for compact and informative data representations. Yet, ML represents a natural approach to CFD, as it can reduce computational burden, provide valuable insights, and extract meaningful patterns from fluid dynamics.

The classical approach for feeding CFD data to a ML model consists of a handcrafted feature extraction, where domain experts or CFD specialists design features to capture specific flow properties within selected regions. These regions are spatial subdomains of the flow field, defined based on prior knowledge, whose selection process acts as an information filtering step that substantially reduces the dimensionality of the CFD data. Despite being intuitive, this procedure introduces a critical limitation: the same selected regions must be consistently identified and propagated to all the samples in the training and test sets, which may not always be straightforward. As an example, in the context of airflow analysis within the human upper airways, Schillaci et al. [4] proposed a CFD classifier that serves as a precursor to our work. In particular, they define cross-sectional planes a priori within simplified human geometries (visible in the middle of Figure 1) to extract averaged flow features. While this approach reduces data dimensionality, it also imposes rigid constraints that may limit generalization to new samples with different anatomical variations or airflow conditions. To the best of our knowledge, this is also the only method currently addressing the inference of non-computable quantities from CFD data, and therefore constitutes the current state of the art for this task. Although effective, handcrafted features can limit portability, lead to information loss, and require intensive domain expertise. Recent advances in ML, particularly in unsupervised learning and deep learning models, offer new opportunities to address these limitations [2,3].

In this work, we propose a method to overcome the drawbacks of handcrafted feature extraction by making a step towards data-driven methods. Our idea is to identify adaptive regions inside the flow field through clustering and extract features from these, without relying on any a priori engineered region selection. We ground our approach on three main components: *i*) a physics-based clustering, inspired by [3], that identifies meaningful regions in the flow field by leveraging

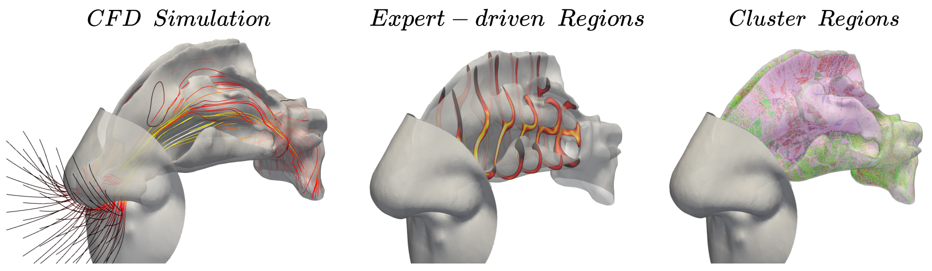


Fig. 1. On the left, the figure shows the flow streamlines in the upper airways. The middle highlights Schillaci et al.'s [4] cross-sections for CFD feature extraction, while the right shows regions from our clustering algorithm.

the physical principles inherent in the governing equations; *ii*) in-cluster features extraction, which captures localized fluid dynamic properties along with geometrical and statistical quantities; and *iii*) set learning models [5] to process the information extracted from clusters.

Unlike traditional expert-driven region selection, our clustering algorithm adopts a data-driven approach grounded on the governing equations of fluid dynamics to define the regions on which we compute features. An example of this can be seen on the right-hand side of Figure 1, which illustrates the application of such clustering to human upper airways. By defining these regions using physical principles rather than heuristic choices, we ensure greater generalizability and adaptability across different scenarios, making our method inherently more portable than expert-driven approaches. However, replacing predefined regions with clusters introduces new challenges: since the number and order of clusters are not fixed a priori, directly comparing clusters across different samples becomes challenging. To address this, we adopt two complementary strategies: one directly processes the unordered clusters using set-learning models, which operate on unordered sets of features and do not require a predefined cluster order, while the other restores cluster comparability through a propagation strategy, mapping clusters from a reference sample onto new samples to preserve consistency in cluster definition. These strategies enable our method to effectively handle variability in cluster structures while retaining the advantages of a physics-based, data-driven approach. Another key advantage of our method is that we can extrapolate information from the whole CFD data without relying on sub-portions of the computational domain, which inevitably leads to a loss of information.

We conduct our experiments on different scenarios of increasing complexity. First, we validate our method on two large datasets of 2D flow fields from the aerospace domain, including both publicly available data [6] and an extended version we generated. In this setting, where a large amount of data is available, we focus on two regression tasks: *i*) predicting the 4-digit NACA (National Advisory Committee for Aeronautics) airfoil code as in [4], and *ii*) estimating parametrized geometric defects (e.g., bumps, cavities, or cut trailing edges) on the surface of NACA airfoils. To further assess the robustness of our approach in a more challenging real-world scenario with limited data, we apply it to *iii*) the classification of pathologies in 3D human upper airways extracted from patients' CT scans. Despite the reduced number of samples in this setting, our method maintains consistent performance, demonstrating its ability to generalize across different flow domains and data availability conditions.

The main contributions of our work are: *i*) We extract informative features from CFD data by identifying physics-based regions in the flow field; *ii*) We take a step toward end-to-end processing of CFD data for ML models, moving beyond existing expert-driven methods; *iii*) We validate our approach on datasets of varying complexity, showing improved performance over the state-of-the-art method by Schillaci et al. [4].

2 Related Work

Over the past 5-10 years, the application of ML in the field of fluid mechanics has experienced significant growth. This is evident from the increasing quantity and quality of published material [1,7,8]. The dominant use of ML for CFD focuses on bypassing the solution of differential equations that govern fluid motion, e.g., by physics-informed neural networks (PINNs) [9,10]. Another prominent research direction addresses turbulence modeling, where for instance, Ling et al. [11] used deep neural networks to refine the Spalart-Allmaras turbulence model [12]. Fukami et al. [13] explored ML for regression tasks, such as flow field reconstruction and estimation, and applied convolutional neural networks for super-resolution tasks, training models to extract key flow features. However, these studies primarily aim to improve or accelerate existing CFD capabilities by using fluid dynamic quantities as both input and output of their ML models. In contrast, our work focuses on a less-explored area: extracting meaningful information from CFD data to predict quantities that CFD alone cannot compute (e.g., pathologies). This problem is particularly challenging because of the dimensionality of CFD data and the difficulty in finding public datasets of annotated CFD simulations.

Given the large amount of data returned from CFD, feeding a ML model directly with flow fields is computationally prohibitive, whereby feature extraction methods are crucial to reduce complexity and focus on the most relevant information. A classical approach to feature extraction relies on expert-driven, handcrafted features designed to capture specific flow properties in predefined regions. For instance, Schillaci et al. [4] extracted predefined spatial regions from the flow field, and averaged fluid dynamic quantities within each region. This method was used to infer geometric properties in two different scenarios: *i*) predicting the NACA code from CFD data around airfoils, and *ii*) identifying pathologies from the internal flow in a simplified human nose. While effective, this approach relies heavily on domain expertise, is restricted to small sub-portions of the flow field, which may lead to the loss of critical information, and lacks portability across different scenarios.

Recent advances in ML, particularly in unsupervised learning and deep learning models, offer new opportunities to overcome the limitations of expert-driven approaches and highlight a growing trend toward end-to-end methods. For instance, Callaham et al. [3] employ Gaussian mixture models (GMM) and sparse principal component analysis (SPCA) to identify dominant physical processes in CFD flow fields. Their method segments flow regions based on local balance relationships in the governing equations, enabling the discovery of physically meaningful subspaces where certain terms in the Navier–Stokes equations can be neglected. Similarly, Saetta et al. [2] employ GMMs to segment homogeneous flow regions in CFD solutions, segmenting boundary layers, shock waves, and external inviscid flow. Their approach eliminates the need for case-dependent thresholds and human intervention, achieving results comparable to reference methods in aerodynamics. These works exemplify the increasing adoption of ML for structuring CFD data and enhancing analysis beyond traditional heuristic techniques. However, while [2] and [3] focus on data exploration and unsupervised structure identification, our approach leverages physics-based clustering explicitly for feature extraction and supervised inference. To the best of our knowledge, end-to-end trainable solutions have not yet been adopted in flow fields, and the method proposed by Schillaci et al. [4] remains the only published approach that uses CFD data to predict quantities that cannot be directly computed from the flow variables themselves. Inspired by [2] and [3], we target the prediction of such non-computable quantities (e.g., airfoil defects, pathologies), integrating flow segmentation into a trainable pipeline for regression and classification tasks. By grounding clustering on the governing equations rather than heuristics, we enhance portability and mark a significant step toward end-to-end ML methods for CFD analysis.

3 Problem Formulation

The flow field returned by a CFD simulation consists of a set of scalar and vector fields defined over a spatial domain $\Omega \subset \mathbb{R}^3$, discretized into n cells. These fields are derived by solving the discretized Navier–Stokes equations with boundary conditions applied to the geometrical boundary $\Gamma \subset \mathbb{R}^3$. CFD simulations typically produce several output flow variables, which are usually time-dependent. To simplify the analysis, we remove the dependency on time by considering steady-state solutions or time-averaged quantities, whereby the generic vectorial flow quantity $\boldsymbol{\phi}$ can be written in Cartesian coordinates as:

$$\boldsymbol{\phi}(\mathbf{x}) = \left[\phi_x(\mathbf{x}), \phi_y(\mathbf{x}), \phi_z(\mathbf{x})\right]^T ,$$

where $\mathbf{x} = (x, y, z) \in \Omega$.

Each cell i in the discretized domain Ω is associated with a vector $\mathbf{Q}_i \in \mathbb{R}^D$ of D flow variables, such that $\mathbf{Q}_i = \left[q_1, q_2, \ldots, q_D\right]^T$, where each q_i represents either a spatial coordinate, a scalar quantity (e.g., pressure), or a component of a vectorial flow variable (e.g., velocity components u_x, u_y, u_z). The complete CFD output can be represented as a matrix $\mathbf{F} \in \mathbb{R}^{n \times D}$, containing all flow quantities across the discretized domain:

$$\mathbf{F} = \left[\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_n\right]^T,$$

In some applications (e.g., human upper airways), n can easily range in the order of $n \approx 10^7$, while the dimensionality of each cell, D, typically spans tens of flow variables.

Our objective is to train a model $\mathcal{K}$ that predicts a target variable Y from the CFD data $\mathbf{F}$, $\mathcal{K} : \mathbf{F} \mapsto Y$, where Y may represent categorical (e.g., pathology classification) or continuous values (e.g., defects parameters). This requires a labeled training dataset $\{(\mathbf{F}_j, Y_j)\}$, where $\mathbf{F}_j$ represents the input features extracted from the j-th simulation, and Y_j denotes the corresponding target variable.

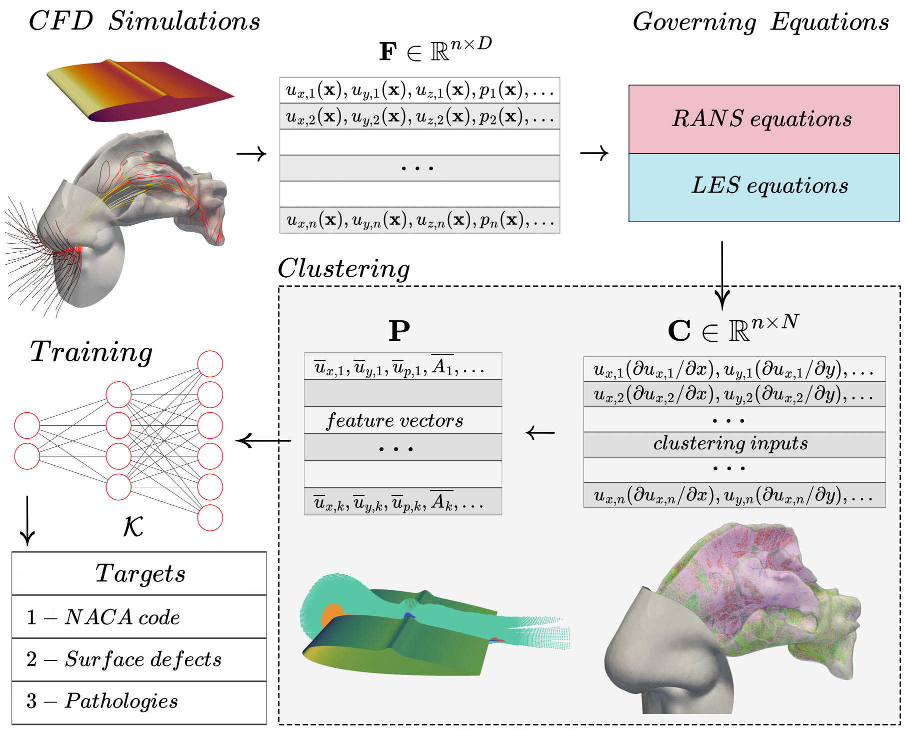
4 Method

Our method applies a clustering algorithm to process the matrix $\mathbf{C} \in \mathbb{R}^{n \times N}$, derived from the CFD data matrix $\mathbf{F} \in \mathbb{R}^{n \times D}$ through the governing equations of the CFD solver (top of Figure 2). The clustering step groups the flow field into meaningful regions based on these equations (top-right corner of Figure 2), capturing the underlying physical properties of the flow. Thus, within each cluster, characterized by a particular physical phenomenon (illustrated at the bottom of Figure 2), we compute fluid dynamic and geometric features, such as average flow quantities, cluster areas, and centroids. Clustering reduces the dimensionality of each simulation, condensing the data into a compact set of vectors $\mathbf{P}$, where each vector represents a cluster and contains features extracted from it, associated with the cluster centroid. This set of feature vectors is then used as input to an ML model trained to predict different target quantities (bottom-left of Figure 2).

4.1 Physics-Based Clustering

To reduce the dimensionality of $\mathbf{F}$ and extract meaningful structures, we apply clustering to the CFD data. The clustering process is based on the physics of the problem, as it leverages terms derived from the momentum equations of motion of the model we use to perform CFD simulations, such as the Reynolds-averaged Navier-Stokes (RANS) equations [14], or Large Eddy simulations (LES) equations, as inputs of the clustering algorithm.

Clustering Inputs Inspired by [3], we apply clustering not directly to the CFD matrix $\mathbf{F} \in \mathbb{R}^{n \times D}$, but to a transformed matrix $\mathbf{C} \in \mathbb{R}^{n \times N}$, obtained by a mapping of $\mathbf{F}$ through the governing equations. In CFD, each cell in the computational domain is represented by a vector of flow variables $\mathbf{Q}_i \in \mathbb{R}^D$ (as mentioned in Section 3), containing raw CFD quantities such as velocity, pressure,



CFD Simulations
$\mathbf{F} \in \mathbb{R}^{n \times D}$
$u_{x,1}(\mathbf{x}), u_{y,1}(\mathbf{x}), u_{z,1}(\mathbf{x}), p_1(\mathbf{x}), \ldots$
$u_{x,2}(\mathbf{x}), u_{y,2}(\mathbf{x}), u_{z,2}(\mathbf{x}), p_2(\mathbf{x}), \ldots$
$u_{x,n}(\mathbf{x}), u_{y,n}(\mathbf{x}), u_{z,n}(\mathbf{x}), p_n(\mathbf{x}), \ldots$
Governing Equations
RANS equations
LES equations
Clustering
P
$\overline{u}_{x,1}, \overline{u}_{y,1}, \overline{u}_{p,1}, \overline{A_1}, \ldots$
feature vectors
$\overline{u}_{x,k}, \overline{u}_{y,k}, \overline{u}_{p,k}, \overline{A_k}, \ldots$
$\mathbf{C} \in \mathbb{R}^{n \times N}$
$u_{x,1}(\partial u_{x,1}/\partial x), u_{y,1}(\partial u_{x,1}/\partial y), \ldots$
$u_{x,2}(\partial u_{x,2}/\partial x), u_{y,2}(\partial u_{x,2}/\partial y), \ldots$
clustering inputs
$u_{x,n}(\partial u_{x,n}/\partial x), u_{y,n}(\partial u_{x,n}/\partial y), \ldots$
Training
$\mathcal{K}$
Targets
1 – NACA code
2 – Surface defects
3 – Pathologies

Fig. 2. Overview of our method. Starting from CFD simulations, we collect the flow field data in the matrix $\mathbf{F} \in \mathbb{R}^{n \times D}$ (top-left). We derive the matrix $\mathbf{C}$ by processing $\mathbf{F}$ through the governing equations (top-right), and use it as input for the physics-based clustering of the flow domain. The clustering step groups the flow field into meaningful regions, capturing the underlying physical properties. From each cluster, we extract statistical, geometric, and physical features, building a set of feature vectors $\mathbf{P}$ (bottom-right). These are then fed into a machine learning model $\mathcal{K}$ for training (bottom-left), with the goal of predicting target quantities.

and turbulence-related terms. Similarly, each column of $\mathbf{C}$ is defined cell-by-cell by a new vector of CFD quantities $\mathbf{C}_i \in \mathbb{R}^N$ which contains the terms of the equations used by the CFD solver. This transformation maps the CFD data $\mathbf{F} \in \mathbb{R}^{n \times D}$ to the matrix $\mathbf{C} \in \mathbb{R}^{n \times N}$, where $N < D$. By applying clustering directly to the columns of $\mathbf{C}$, which include the terms derived from the governing equations, we ensure that the process captures the underlying physics, allowing the model to identify meaningful regions guided by the flow dynamics.

We report in (1) the RANS equations, an analytical model for turbulent flows that decomposes flow variables into a mean component, which represents the averaged flow behavior in time, and a fluctuating component, which captures turbulence effects. For RANS equations, $\mathbf{C}_i$ includes the Cartesian components of the advection term, pressure gradient, turbulent and laminar diffusion, and turbulent kinetic energy (TKE) terms, namely:

$$\begin{aligned}\underbrace{\nabla \cdot (\mathbf{u} \otimes \mathbf{u})}_{\text{Advection}} = \underbrace{\nu \nabla^2 \mathbf{u}}_{\text{Laminar diffusion}} + \underbrace{\nabla \cdot (\nu_\mathrm{t} \nabla \mathbf{u})}_{\text{Turbulent diffusion}} \\ - \underbrace{\frac{1}{\rho} \nabla p}_{\text{Pressure gradient}} \underbrace{- \nabla \left(\frac{2}{3} k \right)}_{\text{TKE}},\end{aligned} \tag{1}$$

where $\mathbf{u}$ is the mean velocity, p is the mean pressure, ρ is the fluid density (assumed to be constant for incompressible flows), ν is the kinematic viscosity of the fluid, ν_t is the turbulent viscosity computed via a turbulence model to account for the effects of the velocity fluctuations u_i' and pressure fluctuations p', and k is the TKE, defined as $k = \frac{1}{2}(\mathbf{u}' \cdot \mathbf{u}')$. The operator ∇ denotes the gradient, while $\nabla \cdot$ and ∇^2 represent the divergence and Laplacian, respectively. For LES, we define $\mathbf{C}$ in a similar way: in this case, a spatial filtering operation replaces time averaging, separating the resolved large-scale structures from the subgrid-scale motions, which are modeled through turbulence closures.

Clustering Algorithm We leverage a Bayesian Gaussian Mixture Model (BGMM) [15] to cluster the column vectors of $\mathbf{C}$. By doing that, we identify flow regions based on the underlying physical properties of the flow (at the bottom of Figure 2), which we then use to extract features in $\mathbf{P}$. The key characteristic of the BGMM is the use of covariance matrices to model the statistical distribution of each cluster extracted from $\mathbf{C}$, effectively capturing correlations between the terms in (1). This allows the BGMM to distinguish regions governed by different flow phenomena, such as boundary layer development, shear layers, separation zones, and wake structures. Moreover, the variational Bayesian inference framework autonomously determines the optimal number of clusters k, eliminating manual tuning and allowing each simulation to return a different number of clusters. By doing that, the clustering adapts to the physics of each simulation, making our method as general as possible. To improve numerical stability and convergence, we initialize the Gaussian components using the centroids obtained from k-means clustering.

4.2 Feature Extraction

We associate each cluster with its centroid and compute a set of in-cluster quantities to be used as a feature vector. The idea is to generalize the regional averaging approach used in [4], replacing predefined spatial regions with clusters while incorporating a broader set of features. These include the averages of flow variables, turbulence quantities, and clustering inputs defined in Section 4.1 as columns of $\mathbf{C}$, weighted by the cell area or volume, along with the cluster area or volume. This process reduces the dimensionality of the clustering input $\mathbf{C} \in \mathbb{R}^{n \times N}$ to a compact set of feature vectors $\mathbf{P} = \{\ \mathbf{P}_i \in \mathbb{R}^l,\ i = 1, ..., k\}$, where $k \ll n$ is the number of clusters returned by the BGMM, and $l = \mathcal{O}(N)$ represents the number of extracted features per cluster. Notably, $\mathbf{P}_i$ also includes the coordinates of the centroids to preserve spatial information, which is fundamental for a Point Transformer, the architecture we employ for set learning. We further highlight that $\mathbf{P}$ is not directly comparable across different flow fields, as both k and the ordering of clusters in $\mathbf{P}$ can vary from case to case.

4.3 Clustering Strategies

The clustering process condenses the CFD data from the matrix $\mathbf{C} \in \mathbb{R}^{n \times N}$ to a more compact set representation $\mathbf{P}$. In this process, however, we cannot directly control the ordering of the clusters returned by the BGMM, that is, the order in which the clusters appear in set $\mathbf{P}$. Therefore, using a non-permutation invariant ML model $\mathcal{K}$ can lead to mutually inconsistent training samples and misclassification. We thus adopt two different strategies to build the set $\mathbf{P}$:

- *Free clustering* (*C-FREE*): We let the order and the number of the clusters vary in each simulation by applying a BGMM independently to each CFD flow field. By adopting this approach, the order of the k clusters in $\mathbf{P}$ is uncontrolled and needs to be accounted for in the choice of model $\mathcal{K}$. Nevertheless, we do not impose any fixed clustering structure, and the method is more flexible and adaptive to different flow conditions and scenarios. Here, we treat $\mathbf{P}$ as an ensemble of feature vectors where the ordering is not relevant. We use a set-learning model [5] for $\mathcal{K}$ trained on $\mathbf{P}$, providing a straightforward solution that naturally handles unordered data sets.
- *Clustering propagation* (*C-PROP*): Set-learning models are less effective than classical MLPs on ordered data. Therefore, we provide a method to consistently define clusters across different simulations. In particular, we select a reference CFD simulation where clusters have been computed as described in Section 4.1. We then propagate these clusters to all other CFD flow fields by matching the clustered data in $\mathbf{C}$ to those of the reference simulation. Specifically, given $\mathbf{C}^{\text{ref}}$ from the reference clusters and $\mathbf{C}^{s}$ from a new simulation s, we assign each cell in $\mathbf{C}^{s}_{j}$ to the closest cells $\mathbf{C}^{\text{ref}}_{i}$ based on a Euclidean distance metric $d(\mathbf{C}^{\text{ref}}_{i}, \mathbf{C}^{s}_{j})$. This matching process is implemented using a *k-d tree* [16], and the core rationale relies on preserving similarity in the space of the columns of $\mathbf{C}$.
  As the clustering inputs are derived directly from the terms of the governing equations, we ensure that the cluster propagation inherently captures the physics of the problem. With this approach, on the one hand, we preserve the number and the order of the clusters in $\mathbf{P}$ across simulations, maintaining consistency in the representation of the flow field and enabling direct comparisons between simulations. On the other hand, we constrain simulations to only capture the physical phenomena that characterize the reference simulations, reducing generality.

4.4 Inference Models

Depending on the clustering strategy we adopt, we employ different ML models: in *C-PROP*, where the clustering is propagated from the reference $\mathbf{C}^{\text{ref}}$, we train a simple MLP. However, in the case of *C-FREE*, we require a model capable of handling sets as input, i.e., unordered inputs of varying dimensions. To achieve this, in the *C-FREE* setting, we use for $\mathcal{K}$ a Point Transformer model (PT) [5] which provides a dual advantage: along with the invariance to input order, it also extracts spatial information from the clusters by leveraging self-attention mechanisms specifically designed for point clouds, modeling interactions between features extracted from clusters based on their geometric proximity and feature similarity. This ability to learn spatially-aware representations can be a fundamental aspect in a CFD context as spatial information is crucial for understanding the physics of the problem. Furthermore, as we work with relatively small yet inherently highly complex datasets, it is essential to extract as much meaningful information as possible to enhance the learning process and improve inference performance.

5 Experiments

We test our method on three distinct tasks of increasing complexity, each addressing a different application of CFD analysis: airfoil shape identification, surface defect detection on airfoils, and pathology classification in real human upper airways extracted from patients' CT scans. In each scenario, we conduct four parallel experiments in which we extract features in set $\mathbf{P}$ using different methods, as described in the following section. We directly compare our pipeline against the handcrafted-feature-based approach proposed by Schillaci et al. [4], which represents the current state of the art for the inference of non-computable quantities from CFD data, and use it as a baseline across all tasks. We first validate our approach on a publicly available dataset [6] and an extended version that we produced. These datasets consist of 2D flow fields with a large number of samples, allowing for extensive evaluation. Then, we apply our method to a more realistic real-world problem, that is, pathology identification in human upper airways directly extracted from CT scans, where data availability is significantly more limited and expensive. Additionally, this dataset consists of 3D flow fields, increasing the complexity of the problem by introducing an additional spatial dimension and geometric variability. This transition from a controlled, data-rich 2D setting to a real-world, data-scarce 3D scenario enables a comprehensive assessment of the method's adaptability and robustness.

5.1 Considered Methods

In this section, we describe the experiments we performed to evaluate our method. We consider 4 parallel approaches to extract $\mathbf{P}$, each differing in the clustering and feature selection strategy. The first approach, *HC* (Hand-Crafted features), serves as a baseline method. It relies on expert-driven feature selection proposed in [4], where CFD quantities are manually extracted based on prior domain knowledge, without leveraging clustering. In *CR+HC* (Clustering Regions with Hand-Crafted features), we introduce our clustering method to identify regions, while retaining the same feature as in *HC*. This allows us to assess how clustering influences performance without altering the feature set with respect to *HC*. A more data-driven approach is adopted in *FREE-CR+FC* (Free Cluster Regions with Full Clustering features), where both clustering regions and feature selection are determined without constraints using the *C-FREE* strategy, described in Section 4.3. In this case, we employ a PT for training and testing, as detailed in Section 4.4. Finally, in *PROP-CR+FC* (Propagated Cluster Regions with Full Clustering features), we enforce cluster consistency across simulations by propagating clusters from a reference case using the *C-PROP* strategy. Unlike *FREE-CR+FC*, this approach aligns clusters across different flow fields, allowing for direct comparisons of their feature representations. Similar to *HC* and *CR+HC*, in this method we employ an MLP for training and testing. In the following paragraphs, we provide details on the dataset structure and the process used to generate features for each experiment.

5.2 Datasets and Tasks

Airfoil Shape Identification (AirNACA) We consider the family of NACA (National Advisory Committee for Aeronautics) four-digit airfoils. We aim to train a regressor $\mathcal{K}$ that predicts the NACA code, and thus the airfoil shape, directly from the CFD solution. The shape of a NACA airfoil is defined by a four-digit code, which parametrizes the maximum camber as a percentage of the chord c (I digit, ranging from 0 to 9, where c is the straight-line distance from the airfoil leading to the trailing edge), its position along the chord in tenths of c (II digits, also from 0 to 9)), and the maximum thickness as a percentage of c (last two digits, ranging from 5 to 50). Therefore, the identification task reduces to a regression over 3 integer numbers.

The 2D computational domain Ω is centered on the airfoil and extends over a radius of $500c$, with unitary chord c, comprising an order of $\mathcal{O}(10^5)$ discretized cells. The angle of attack is set to $\alpha = 10°$, with a freestream velocity of 30 m/s, and the RANS equations are used for turbulence modeling. The dataset comprises 3025 different flow fields, generated by varying the NACA digits and solving the corresponding CFD problems. The complete dataset of RANS simulations is publicly available on Zenodo (10.5281/zenodo.4638071) [6], and the implementation of *AirNACA* is provided at 10.5281/zenodo.15637850.

Surface Defect Detection (AirDEF) The second dataset extends the NACA airfoil dataset by introducing controlled geometric deformations to simulate manufacturing defects, structural damage, or ice accretion. The simulation setup remains identical to the previous case, maintaining the same computational domain, boundary conditions, and freestream parameters. We consider bumps, cavities, and cut trailing edges, which are parameterized and encoded in a 3-digit code. Each airfoil undergoes controlled deformations by introducing a set of 18 different surface defects applied individually or in combination, for a total of 3600 CFD flow fields. Bumps and cavities, as shown in Figure 4, are modeled with Gaussian functions at the chord midpoint: a bump raises a section of the

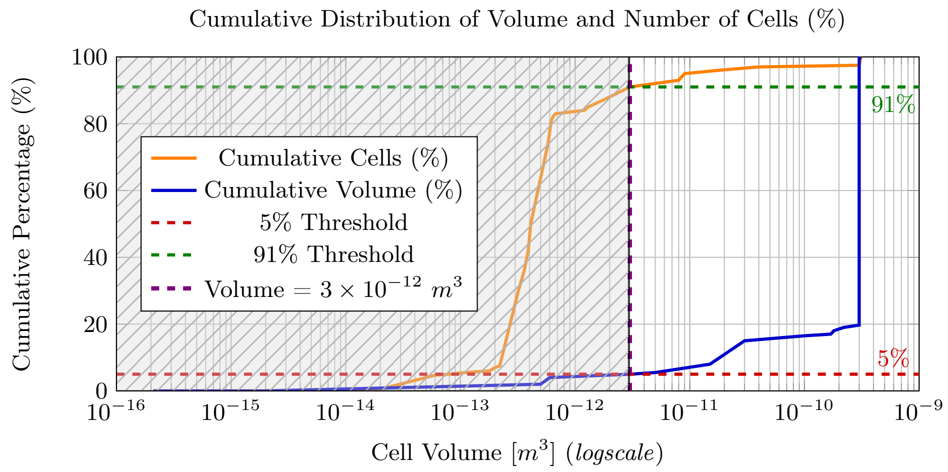


Fig. 3. Cumulative percentage distribution of Cell Volume and Number of Cells with respect to the size of the cells. The dashed lines represent the percentage of volume we lose with the filtering (red line), the percentage of memory we save (light blue line), and the volume threshold (violet line).

airfoil, while a cavity creates a small indentation. The first two digits ([-2:2]) define the orientation (bump or cavity) and the intensity of the deformation, which can reach up to the 4% of the chord c. The last digit ([0:2]) controls the intensity of the cut at the trailing edge, which can raise up to the 5% of c.

Pathology Identification in Real Upper Airways (NosePAT) The third dataset consists of LES simulations of airflows in human upper airways, focusing on the identification of septal deviations and turbinate hypertrophies on real 3D patient geometries obtained from CT scan segmentation. A septal deviation is a condition where the nasal septum is deviated to one side, potentially obstructing airflow and causing breathing difficulties. Turbinate hypertrophy refers to the excessive enlargement of the nasal turbinates, leading to nasal congestion.

The dataset is derived from 7 CT scans of healthy patients provided by *ASST Santi Paolo e Carlo*, a medical institution we are collaborating with. ENT (Ear, Nose, and Throat) specialists manually introduced geometric deformations to synthetically simulate nasal pathologies, varying their locations and severities to create a diverse set of pathological cases. This process resulted in 309 synthetic pathological geometries, on which we performed LES simulations of a steady-state inspiration. The CFD simulations were performed using OpenFOAM, employing the LES technique under incompressible flow assumptions. By using real CT scans for dataset generation, this approach captures anatomical variability while preserving realism. In addition to the 309 synthetic geometries, we extracted 10 real pathological cases (5 septal deviations, 5 hypertrophies) from CT scans to evaluate whether a model trained on synthetic data can accurately detect real conditions (we refer to this experiment as ***NoseREAL***).

The computational domain discretizes the upper airways' internal volume, enclosing the nostrils within a spherical boundary to simulate an open environment. Each mesh consists of $\mathcal{O}(10^7)$ cells, each storing tens of fluid dynamic quantities. Simulations run on 96 cores on a high-performance computing system, requiring 160GB of RAM and tens of thousands of core-hours, producing around 40GB of data each. Generating real human upper airway data demands specialized expertise and substantial computational resources.

Figure 3 shows the cumulative distribution of mesh cells and total volume by cell size. Most cells' volume range between $2 \times 10^{-13}\, m^3$ and $5 \times 10^{-13}\, m^3$, but they contribute minimally to the total volume, which is largely concentrated (80%) in cells around $3 \times 10^{-10}\, m^3$. To reduce computational costs, after LES simulations, we filter out cells with $V < 3{\times}10^{-12}\, m^3$ (violet dashed line in Figure 3), reducing data by 91% while retaining 95% of the total volume. The removed cells, located near walls (on a millimeter scale), are essential for CFD accuracy but contain little information related to pathology effects, which manifest on larger scales. This filtering is purely for computational efficiency, and including these cells would only add information, potentially improving performance.

5.3 Feature Extraction

For the ***HC*** experiment, we compute the same features from the CFD flow fields as in [4]. In *AirNACA* and *AirDEF*, this involves computing region-averaged velocity magnitude and pressure on expert-defined regions along three vertical lines perpendicular to the airfoil chord (left-hand side of Figure 4) at $x = -c$, $x = 1c$, and $x = 10c$. Each segment contains 8 regions symmetrically distributed around $y = 0$, with boundaries defined by the y-coordinates at $[-500, -10, -1, -0.1, 0, 0.1, 1, 10, 500]c$. Each regional average is weighted over the cells' area to account for their uneven dimension. In *NosePAT*, we define 6 sections with the first and last one marking the start and end of the olfactory region (right-hand side of Figure 4), while the remaining 4 are evenly spaced between them. On these, we compute the regional average of the velocity magnitude $|\boldsymbol{u}|$ separately for the left and right semi-sections. By doing that, each LES simulation is compressed to 12 features that are used as inputs of $\mathcal{K}$.

For ***CR+HC***, we apply *C-PROP* clustering as described in Section 4.3 using as reference a NACA0012 in *AirNACA* and *AirDEF*, and one of the 7 healthy patients used to generate the training set as reference for propagating clusters in *NosePAT*. Within clusters, we compute the same regional averages as for *HC*, mentioned in the previous paragraph.

For ***FREE-CR+FC*** and ***PROP-CR+FC***, we fully apply our clustering method as described in Section 4, using for *PROP-CR+FC* the same references as for *CR+HC*.

5.4 Models Training and Evaluation

As in [4], we stick to training simple MLPs for *HC*, *CR+HC* and for *PROP-CR+FC*, while we adopt a Point Transformer model for *FREE-CR+FC*. We define models with a comparable number of parameters (around $50K$ parameters in *AirNACA* and *AirDEF*, and $30K$ for *NosePAT*) to evaluate the performance

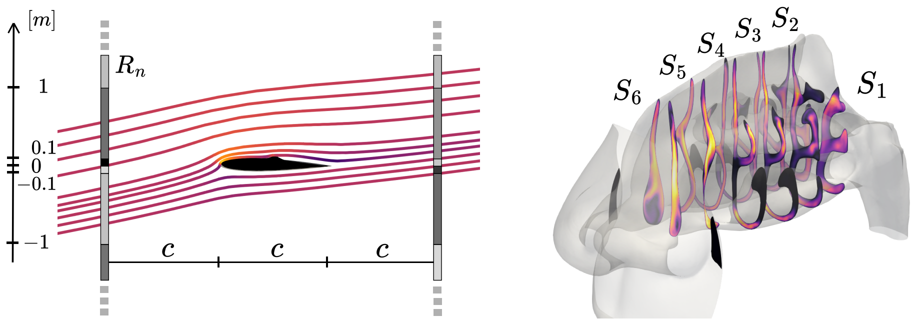


Fig. 4. Handcrafted regions in *AirNACA* and *AirDEF* (left) and *NosePAT* (right).

when considering architectures having similar learning capacities. In *AirNACA* and *AirDEF*, the dataset is standardized and shuffled before being split into five folds for cross-validation. In each fold, one subset is used for testing, while the rest is split into training and validation (80% and 20%). We train our models using the mean squared error (MSE) as loss function. As these are regression tasks and the predicted values are not necessarily integers, they are rounded to obtain the final output code. We report the averaged results across the five folds, considering the mean absolute error (MAE) over each estimated code and the classification accuracy, which represents the percentage of correctly predicted codes when rounding predictions to the nearest integer.

In *NosePAT*, we evaluate the performance of our models with a Leave-One-Patient-Out Cross-Validation (*LOPOCV*): in this approach, all the synthetic samples derived from each of the 7 healthy patients are iteratively excluded from the training set and used as test cases, ensuring that models are evaluated on anatomies that were never seen during training. For each iteration, the remaining patients' data are split into 85% for training and 15% for validation. Eventually we average the results of the test over all the excluded patients. Unlike the previous regression tasks, this problem is now framed as a binary classification task, and models are trained using categorical cross-entropy as the loss function. The primary evaluation metric is classification accuracy. In the *NoseREAL* experiment, we train our models on the complete synthetic dataset and then test on the set of real pathological samples, assessing the performance on never-seen-before anatomies and pathological forms.

5.5 Experimental Results

The results of our experiments are reported in Table 1 and 2. Table 1 shows the test accuracy of the inference models trained using features extracted with methods described in Section 5.1. We stress that in regression tasks (*AirNACA* and *AirDEF*) we round the results to the nearest integer and evaluate the accuracy. Alongside the accuracy of the *LOPOCV* for the *NosePAT* task, we also report the scores of the classifier on the set of real pathological patients (*Nose-*

	Test Accuracy			Score
	AirNACA	*AirDEF*	*NosePAT*	*NoseREAL*
HC	84.6%	65.6%	**88.8%**	**8/10**
CR+HC	85.0%	83.2%	71.5%	6/10
PROP-CR+FC	**86.5%**	**88.7%**	86.8%	**8/10**
FREE-CR+FC	85.1%	84.3%	77.5%	7/10

Table 1. Test accuracy of *HC*, *CR+HC*, *PROP-CR+FC* and *FREE-CR+FC*. On the right-hand side of the table, we report the score on the set of real pathological patients.

	Mean Absolute Error (MAE) and Standard Deviation (std)											
	AirNACA						*AirDEF*					
	I digit		II digit		III digit		I digit		II digit		III digit	
	MAE	±std	MAE	±std	MAE	±std	MAE	±std	MAE	±std	MAE	±std
HC	0.17	0.23	0.30	0.24	0.17	0.26	0.35	0.25	0.33	0.18	0.11	0.15
CR+HC	0.16	0.20	0.28	0.21	0.16	0.23	0.22	0.21	0.21	0.17	0.03	0.12
PROP-CR+FC	**0.14**	0.19	**0.24**	0.21	**0.15**	0.24	**0.18**	0.19	**0.19**	0.18	**0.01**	0.11
FREE-CR+FC	**0.14**	0.21	0.26	0.20	0.16	0.19	0.21	0.22	0.21	0.18	0.02	0.09
Range	[0:9]		[0:9]		[5:50]		[-2:2]		[-2:2]		[0:2]	

Table 2. Mean Absolute Error (MAE) and standard deviation (std) for each digit of the regression code in *AirNACA* and *AirDEF* tasks.

REAL). Table 2 contains the MAEs and the standard deviations over the test set predictions for each individual digit that composes the regression code.

Table 1, shows that the clustering is overall beneficial in almost every task. For both *AirNACA* and *AirDEF*, the clustering progressively improves the classification accuracy of *HC*, both when considering clustering regions in *CR+HC*, and when using all the in-cluster features (*PROP-CR+FC*). We argue this improvement is due to the use of physics-based identified cluster regions and the more informative features we extract from those regions, which also contain geometric information that is crucial for identifying defects. This is particularly evident in the case of *AirDEF*, where *HC* alone achieves the lowest performance. In *HC*, indeed, we extract information from regions that are distant from the airfoil, making it less effective in capturing the local aerodynamic effects of defects. Although *FREE-CR+FC* outperforms *HC* in both *AirNACA* and *AirDEF*, its accuracy remains lower than that of *PROP-CR+FC*. This result is consistent with the increased difficulty of the inference problem in *FREE-CR+FC*, where cluster ordering is not constrained.

In *NosePAT*, we report the test accuracy of the *LOPOCV* described in Section 5.4. The trend in *CR+HC*, *PROP-CR+FC* and *FREE-CR+FC* is confirmed, demonstrating that including in the training set more in-cluster features enhances the classification performance. Also, in the case *FREE-CR+FC*, the accuracy reduces with respect to *PROP-CR+FC*, reflecting the higher complexity of the problem. However, in *NosePAT*, *HC* achieves the best test performance, which is not surprising as the 6 transversal sections we use in *HC* to extract features are designed ad-hoc for this particular task, and two of these align with pathological deformations, directly capturing critical information.

Table 1 shows also the scores of the classifier on the set of real pathological patients, which represent a significant real-world application. In this experiment, we train a classifier on CFD data from synthetically pathological geometries and then test it on actual pathological conditions. Despite the relatively small dimension of the dataset, the results demonstrate that we can effectively identify real pathologies on unseen patients using solely CFD data. *PROP-CR+FC*, in fact, achieves an 8/10 score with a trend between *CR+HC*, *FC+PROP-C*, and *FREE-CR+FC* similar to *AirNACA* and *AirDEF*.

From Table 2, we observe that the II digit in *AirNACA* consistently exhibits the highest MAE across all experiments, indicating it is the most challenging parameter to predict. Conversely, the third digit, representing airfoil thickness, is well predicted, showing the lowest MAE relative to its range. Similarly, in *AirDEF*, the third digit also achieves the lowest MAE, while the first two digits yield comparable errors. Here, the last digit corresponds to a trailing-edge cut, whose high predictability is likely due to its distinct aerodynamic impact, as it generates a structured wake effectively captured by physics-based clustering.

Overall, results from *AirNACA* and *AirDEF* confirm that our method improves inference performance on aerospace datasets while maintaining consistency across experiments. *NosePAT* demonstrates that our method can be applied to a complex real-world task such as pathology identification, even when relying on limited CFD data.

5.6 Computational Considerations

Our method introduces additional computational overhead due to the clustering step and the use of more advanced learning architectures with respect to the baseline approach proposed in [4]. Clustering was performed on the Leonardo Data Centric General Purpose (DCGP) partition of the CINECA HPC system, using 3 compute nodes each equipped with 2× Intel Xeon Platinum 8480+ CPUs, with 56 cores per CPU. In our implementation, we run one clustering process per core in parallel. We apply BGMM to the matrix $\mathbf{C} \in \mathbb{R}^{n \times N}$, where n is the number of CFD cells. For the 2D airfoil simulations (*AirNACA* and *AirDEF* tasks), where $n \approx 300K$, the BGMM requires around 10–12 minutes per sample. In the *NosePAT* dataset, we filtered out the smallest cells, removing 91% of the data but retaining 95% of the total volume as described in Section 5.2, resulting in roughly 1.5-1.7 million cells per sample. Clustering on this reduced set takes about 1.5 hours per sample (consistent with the linear scaling of the BGMM with n), and this cost remains manageable and predictable. This overhead is absent in the *HC* baseline, which uses manually defined regions. However, in practice, defining such regions, especially in anatomically complex domains like human upper airways, requires expert input and manual inspection, introducing a non-negligible cost in human time and effort. Our method removes this dependency by automatically identifying meaningful flow regions in a data-driven and physics-informed manner.

In terms of inference time, the Point Transformer model used in the *FREE-CR+FC* setting is more computationally demanding than the MLPs we employed in *HC*, *HC+FC*, and *PROP-CR+FC*. While all architectures are constrained to have a similar number of parameters within the same task, the Point Transformer involves more complex operations, such as attention over sets of clusters, which lead to longer training times. In our experiments, training the Point Transformer on standard hardware with GPU acceleration takes the order of tens of minutes, whereas the MLPs typically converge in a matter of minutes. We consider this additional training overhead acceptable, given the improved flexibility and generalization achieved by the set-learning architecture.

In summary, although our method introduces additional computational costs with respect to the handcrafted baseline, it provides a scalable and general solution that eliminates manual feature engineering and enables consistent application across diverse CFD domains.

6 Conclusion

In this work, we proposed a physics-based clustering framework to extract meaningful flow structures from CFD data, enhancing the inference of non-computable quantities in aerodynamic and biomedical applications. By replacing expert-driven feature selection with a clustering strategy based on governing equations, our method takes a step toward end-to-end approaches. We validated our method on aerospace datasets for airfoil shape and defect identification and extended its application to pathology classification in human upper airways. The results demonstrated improved inference performance over traditional handcrafted features. Moreover, our approach proved robust across different datasets, highlighting its portability and scalability in diverse CFD scenarios. In future works, we will explore more end-to-end methods to directly extract relevant information and gain deeper insights into CFD data, minimizing expert-driven processing.

Reproducibility We provide the implementation of the *AirNACA* task presented in this work, based on the public dataset available at 10.5281/zenodo.4638071. The repository includes scripts for feature extraction, clustering, and model training, and is publicly available at 10.5281/zenodo.15637850.

Acknowledgments. The Authors are grateful to A. Bulfamante and C. Pipolo for the time spent as ENT expert. Computing time has been provided by the Italian CINECA HPC Center in Bologna under the *IscrB_ StocLung* grant. This publication is part of the project PNRR-NGEU which has received funding from MUR-DM 351/2022.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Steven Brunton, Bernd Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.
2. Ettore Saetta and Renato Tognaccini. Identification of flow field regions by machine learning. In *AIAA SCITECH 2022 Forum*, pages 1503–1518. American Institute of Aeronautics and Astronautics, San Diego, CA, 2022.
3. Jared L. Callaham, James V. Koch, Bingni W. Brunton, J. Nathan Kutz, and Steven L. Brunton. Learning dominant physical processes with data-driven balance models. *Nature Communications*, 12(1):1016, 2021.
4. Andrea Schillaci, Maurizio Quadrio, Carlotta Pipolo, Marcello Restelli, and Giacomo Boracchi. Inferring functional properties from fluid dynamics features. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4091–4098, 2021.
5. Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point Transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16239–16248, 2021. ISSN: 2380-7504.
6. Andrea Schillaci, Maurizio Quadrio, and Giacomo Boracchi. A database of cfd-computed flow fields around airfoils for machine-learning applications, 2021.
7. M. P. Brenner, J. D. Eldredge, and J. B. Freund. Perspective on machine learning for advancing fluid mechanics. *Physical Review Fluids*, 4(10):100501, 2019.
8. Neil Ashton, Jordan B. Angel, Aditya S. Ghate, Gaetan K. Kenway, Man L. Wong, Cetin Kiris, Astrid Walle, Danielle C. Maddix, and Gary Page. WindsorML: High-Fidelity Computational Fluid Dynamics Dataset For Automotive Aerodynamics. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:37823–37835, 2024.
9. Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.
10. Pin Wu, Kaikai Pan, Lulu Ji, Siquan Gong, Weibing Feng, Wenyan Yuan, and Christopher Pain. Navier–stokes generative adversarial network: a physics-informed deep learning model for fluid flow generation. *Neural Computing and Applications*, 34(14):11539–11552, 2022.
11. Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.
12. P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, pages 439–445. American Institute of Aeronautics and Astronautics, Reno, NV, 1992.
13. Kai Fukami, Koji Fukagata, and Kunihiko Taira. Assessment of supervised machine learning methods for fluid flows. *Theoretical and Computational Fluid Dynamics*, 34(4):497–519, 2020.
14. Hermann Sohr. *The Navier-Stokes Equations*. Birkhäuser, 2001.
15. Michael D. Escobar and Mike West. Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].
16. Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.