scientific reports

Check for updates

OPEN

# Neural fields for rapid aircraft aerodynamics simulations

Giovanni Catalani[1,2,3]✉, Siddhant Agarwal[4], Xavier Bertrand[1], Frédéric Tost[1], Michael Bauerheim[2] & Joseph Morlier[2]

**This paper presents a methodology to learn surrogate models of steady state fluid dynamics simulations on meshed domains, based on Implicit Neural Representations (INRs). The proposed models can be applied directly to unstructured domains for different flow conditions, handle non-parametric 3D geometric variations, and generalize to unseen shapes at test time. The coordinate-based formulation naturally leads to robustness with respect to discretization, allowing an excellent trade-off between computational cost (memory footprint and training time) and accuracy. The method is demonstrated on two industrially relevant applications: a RANS dataset of the two-dimensional compressible flow over a transonic airfoil and a dataset of the surface pressure distribution over 3D wings, including shape, inflow condition, and control surface deflection variations. On the considered test cases, our approach achieves a more than three times lower test error and significantly improves generalization error on unseen geometries compared to state-of-the-art Graph Neural Network architectures. Remarkably, the method can drastically accelerate high fidelity numerical solver, achieving a five order of magnitude speedup on the RANS transonic airfoil dataset.**

Computational Fluid Dynamics (CFD) has become an indispensable tool in modern aircraft design, offering the possibility to perform high-fidelity simulations of the complex physics of an aircraft. Numerical solutions can be leveraged to evaluate aerodynamic performance, structural loads, and handling qualities at a much lower cost than experimental and full-scale flight test campaigns. These simulations primarily solve the Reynolds-averaged Navier-Stokes (RANS)[1] equations to produce accurate flow field predictions, over the full flight envelope[2]. However, the high computational cost associated with each simulation restricts their use in the early design stages and time-sensitive environments, such as the aerodynamic load estimation required over thousands of flight conditions and design choices. Given the prohibitive computational demands of traditional CFD simulations, there is a critical need for efficient surrogate models. These models are designed to approximate complex simulations quickly and accurately, facilitating rapid iterations during the design process.

Traditionally, methods like Proper Orthogonal Decomposition (POD)[3] combined with interpolation techniques (e.g., radial basis functions or Gaussian Processes[4]) have been prevalent[5,6]. Despite their efficiency and physical interpretability, these models often fall short in scenarios involving strongly non-linear phenomena, particularly at transonic flow conditions, due to their inherent linear nature[7]. Additionally, the fixed geometry and resolution constraints of modal decomposition methods limit their applicability to scenarios involving geometric variations and multi-fidelity data. Attempts to circumvent these drawbacks have focused on operating snapshot clustering[8,9], modifying the POD minimization metric[10], and performing mesh morphing to a common reference mesh[11]. Despite the potential of these proposed methods, their applicability can be limited to specific cases, due to the underlying assumptions.

The recent advances in data-driven modeling, especially deep learning (DL), offer new avenues for constructing surrogate models. Deep learning has demonstrated exceptional capability in extracting and representing complex hierarchical data features across various domains[12–14]. In the context of fluid dynamics, DL methods have been used to perform a variety of tasks: constructing reduced order models[15,16], accelerating numerical solvers[17,18], identifying turbulence closure models[19] or flow control strategies[20,21], are only some of the most relevant examples. Specifically for airfoil and aircraft aerodynamic predictions, Deep Learning applications include modeling aerodynamic coefficients[22], surface pressure distributions[23] pressure calibration[24] and shape optimization[25–27]. Building a surrogate model of the pressure fields has a central importance in all of the above tasks, replacing the need for a computationally expensive numerical simulator. Convolutional Neural Networks (CNNs) have been widely used to build surrogates, thanks to the strong capability of these networks to capture local interactions and the possibility to straightforwardly define multiscale operations. UNET architectures have been applied to predict scalar and vector fields over airfoils and 3D configurations[28–30]. Despite these successful

[1]Airbus, Aircraft Aerodynamics, Toulouse, France. [2]ISAE Supaero (DAEP), Toulouse, France. [3]Institut Clément Ader (ICA), CNRS, Toulouse, France. [4]Airbus, Aircraft Aerodynamics, Bremen, Germany. ✉email: giovanni.catalani@airbus.com

attempts, CNNs expect pixel-like input and output data at fixed resolution: CFD solutions are typically defined on unstructured meshes, and interpolation routines have been employed to map the two different types of data representations, with an inevitable loss in performance[30]. Additionally, the different level of refinement characteristic of numerical simulations, hinders the applicability of CNN to 3D configurations, due to computational constraints.

Geometric deep learning, specifically Graph Neural Networks (GNNs), provide a powerful paradigm to extend CNNs to unstructured domains, exploiting the inductive biases inherent to graph-like data[31]. This flexibility allows GNNs to model complex geometries typical of fluid dynamics simulations more naturally than traditional neural network architectures. Mesh Graph Networks[32] introduce convolution operations on meshes, involving relative node distances as edge features, and demonstrate good performance for fluid mechanics time-dependent problems defined on a common mesh, with a relatively limited number of nodes. Extension of message passing GNNs to larger graphs, inevitably requires pooling operations, in order to model multiscale behavior of fluid flows[33–35] and long range interactions over the mesh. In a monoscale architecture, a larger number of message passing layers would be needed to cover an equivalent graph region, leading to oversmoothing[36] and excessive computational overhead. The definition of pooling operations is, however, dependent on the mesh topology and the kind of application, as no general downsampling rule fits all cases. Despite their general formulation and success across a wide range of applications, the effectiveness of GNN for surrogate modeling of aerodynamic simulations is still limited. Firstly, GNNs architectures encounter challenges in generalizing across different mesh topologies and levels of discretization: this can be explained by the fact the node connectivity and neighborhood definitions are based on the graph's structural metrics rather than the physical metrics of the domain[37]. Refining a mesh over a certain threshold often degrades the performance of the GNN[33], as the neighborhood of each node shrinks, converging to a single point in the limit of infinite refinement, failing the discretization convergence criterion. Typical industrial aerodynamic applications in 3D involve largely refined meshes (even surfacic-only meshes can exceed 1 million nodes per geometry) in order to account for the complex interactions between lifting surfaces and the fluid flow at higher Reynolds numbers. Training a surrogate model at full resolution is often unfeasible with limited hardware resources: scalability requires the possibility of accurately generating predictions at higher levels of discretization while training at significantly coarser levels. It must be noted that a substantial effort to scale GNN to handle large meshes, in the content of aircraft aerodynamics, has been done in the work of Hines and Berkemeyer[23], as dynamic subsampling has been applied to predict the surface pressure distribution on the single NASA CRM aircraft geometry[38] over a range of flight parameters variations.

Recent advances in the area of Operator Learning have allowed the introduction of discretization invariant architecture, bypassing the limitations of GNN which have been mentioned earlier. Neural Operator, learns the mapping between the infinite-dimensional input and output function spaces and can, by construction produce the value of the output function at any point in the spatial domain. Fourier Neural Operator (FNO)[39] has been introduced to solve parametric PDEs, by learning the integral kernel directly in the Fourier Space, leveraging the computational efficiency of Fast Fourier Transforms (FFT) on uniform grids. The Geometry Informed Neural Operator (GINO)[37] employs a Graph Neural Network to map unstructured 3D domains to a structured reference latent grid, where Fourier Layers can be efficiently applied. Similarly, DeepOnet[40] learns operators from a sparse set of observations using two subnetworks, but it is constrained to input functions evaluated always in the same location, being poorly suited for problems involving shape variations. Recently, Geom-DeepOnet was introduced[41], extending the capabilities of the parent DeepOnet methodology to handle parametric geometry variations in 3D. It is important to mention that Physics-Informed Neural Networks (PINNs) have been used to incorporate known physical laws into neural network optimization metrics, potentially enhancing the generalization capabilities of the model even when data is scarce or absent[42,43]. Despite their promise and utility in many scenarios, PINNs face key limitations in complex aerodynamic applications. A major challenge is their difficulty in generalizing to varying geometries and boundary conditions: purely physics-driven PINNs require explicit knowledge of the boundary conditions for each case, limiting their flexibility for complex surfaces like wings with shape variations or control surfaces (e.g., ailerons). Additionally, hybrid PINNs often benefit from observations of all physical quantities (e.g., velocity, pressure, boundary forces) across the entire domain. At the same time, many real-world applications only provide surface measurements, such as pressure distributions. This restricts their use in industrial settings where volumetric flow data is unavailable, or it might be too costly to store. Enforcing physical constraints throughout the domain can also lead to high computational costs, particularly for large, high-resolution 3D geometries.

Implicit Neural Representations (INRs), also known as Neural Fields, have emerged as a powerful alternative to classical methods for learning spatial representations of objects such as images, shapes, and 3D scenes through radiance fields[44–46]. INRs achieve an impressive level of detail with limited memory requirements by approximating functions at any point in the domain. They take spatial coordinates as input and produce the corresponding function value at those coordinates. Recent breakthroughs in the field, particularly in input encoding techniques like SIREN[47] and Fourier Feature Encoding (FFN)[48], have significantly enhanced the ability of INRs to approximate high-frequency functions with unmatched efficiency. SIREN employs periodic activation functions, enabling the representation of complex signals, while FFN leverages a set of predefined frequency components typically sampled from a Gaussian distribution. However, tuning the hyperparameters of these architectures, especially the frequency components in FFN, is nontrivial. The sampled frequencies can significantly impact the model's underfitting or overfitting behavior[48]. This aspect is particularly challenging in fluid dynamics datasets, which often exhibit a wide spectrum of frequencies. While initially limited to representing single samples, advancements such as latent modulation and meta-learning have extended INRs' capabilities to approximate entire classes of objects and datasets[45,47,49,50]. Serrano et al. introduced CORAL[51], a flexible framework for learning initial value problems, modeling PDE dynamics, and building surrogate models on 2D meshes using INRs. This framework demonstrated computational efficiency and the ability to

handle geometric variations accurately. The INFINITY model[52] further specialized this approach for the RANS equations over 2D airfoils in the incompressible regime, achieving state-of-the-art accuracy.

In this study, we present a comprehensive methodology to build surrogate models of steady aerodynamic simulations on meshed domains using Implicit Neural Representations.

Our contributions are twofold. Firstly, we introduce the first application of INR-based surrogate models to 3D wing surface meshes, incorporating shape variations and aileron deflections over a wide range of flight conditions. This showcases the potential of our model for real industrial applications. Our methodology is specialized into two main frameworks: an encode-process-decode design suitable for general problems with non-parametric shape variations, and an end-to-end design optimized for scenarios with fixed geometry or parametric shape variations. Secondly, we propose a Multiscale-INR backbone architecture to optimize spectral convergence, addressing the challenges associated with tuning the frequency hyperparameters of standard Fourier Feature Encoding architectures. This architecture ensures that our model captures the necessary frequency components to accurately represent complex aerodynamic phenomena without extensive hyperparameter tuning. Our extensive experimental study includes two key datasets: a transonic airfoil dataset with fixed mesh configurations and a 3D wing dataset with shape variations and aileron deflections. We chose the 2D transonic airfoil dataset due to its industrial relevance, as accurately predicting shocks is crucial for aircraft aerodynamics, with cruise speeds typically in the transonic range. Traditional methods like POD often struggle to capture these shock phenomena accurately. The 3D wing dataset presents extensive shape variations, making it particularly challenging for most deep learning architectures. We also test the model's generalization capabilities on unseen shapes. These experiments demonstrate the accuracy, efficiency, and scalability of our proposed methodology. We compare our approach with state-of-the-art Graph Neural Network (GNN) baselines, highlighting several aspects crucial to the method's scalability, such as discretization dependency. Additionally, we conduct a discretization dependence study to quantify our method's ability to generalize across different levels of mesh refinement. This study underscores the scalability potential of our framework, enabling accurate predictions at higher levels of discretization while training at significantly coarser levels. This capability is particularly important for industrial applications, where high-resolution simulations are computationally prohibitive.

## Methodology

**Problem Statement** In this work, we address the problem of finding data-driven solutions to steady-state fluid dynamics simulations discretized on meshes. Let $\Omega \subset \mathbb{R}^d$ denote the physical domain of the geometry under consideration, where $d$ indicates the dimensionality of the spatial domain. Let $\mathcal{A} = \mathcal{A}(\Omega; \mathbb{R}^{d_a})$ and $\mathcal{U} = \mathcal{U}(\Omega; \mathbb{R}^{d_u})$ be separable Banach spaces of functions taking values in $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_u}$ respectively. Each element of $\mathcal{A}$ is a function describing the input geometry (for instance through the Signed Distance Function), the boundary conditions and the inflow conditions. Let $\mathcal{G} : \mathcal{A} \to \mathcal{U}$ be a non-linear map, serving as the solution operator of the associated partial differential equation (PDE) that describes the physics of the problem. Numerical methods seek for an approximate solution $u \in \mathcal{U}$ of the governing PDE on a discretized version of the input domain $\Omega$, namely a mesh $\mathcal{M} = (X, \mathcal{E})$, defined by the set of nodes coordinates $X \in \mathbb{R}^{Nd}$ and their connectivities $\mathcal{E} \in \{0, 1\}^{N \times N}$ for specific instances of the input parameter vector $\mu \in \mathbb{R}^{d_p}$. Surrogate models are built to approximate the numerical solution, for different values of the input parameters and different geometrical configurations, typically at a much lower cost than the associated numerical methods. Data driven methods learn this mapping from a dataset of observed input-output pairs:

$$\mathcal{D} = \{(\mathcal{M}_i, \mu_i), U_i\}_{i=1}^{M} \tag{1}$$

where each tuple consists of a mesh ($M_i$), input parameters ($\mu_i$) typically describing the boundary conditions and inflow conditions, and the corresponding field solution ($U_i \in \mathbb{R}^{N_i d_u}$), evaluated on the mesh, usually with a CFD solver. Additional scalars, namely lift and drag coefficients, can be considered as the output of the surrogate model, but in the following, we will focus on building parametric models for full fields predictions. Specifically, we aim to define a methodology to build surrogate models of aerodynamic simulation on meshes, by directly approximating the solution operator of the underlying PDE with a data-driven architecture based on Implicit Neural Representations. It must be highlighted that although the primary objective of a surrogate model is to obtain discretized solutions on a given mesh, reformulating the problem in terms of approximating the continuous solutions operator allows to obtain a flexible and scalable algorithm with respect to discretization, as it will be shown in the following paragraphs.

**Implicit Neural Representations** Implicit Neural Representations (INRs) are coordinate based neural networks parameterized by weights $\theta \in \mathbb{R}^p$ mapping points $\mathbf{x}$ in Euclidean space (or equivalently on a manifold) to vector or scalar quantities:

$$f_\theta(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^{d_u}. \tag{2}$$

INRs can be thought of as a continuous approximation of an underlying signal, for which a discretely sampled version, is available for learning. Leveraging the continuous nature of INRs, numerous data modalities can be modeled, such as images, shapes, and physical fields without constraints on the type of discretization. In this work, we employ INRs to learn concise representations of physical solutions and shapes descriptors, that are invariant across levels and types of discretization. Field predictions can be obtained by querying an INR across multiple coordinate points:

$$f_\theta(X) = [f_\theta(\mathbf{x_1}), ..., f_\theta(\mathbf{x_N})]. \tag{3}$$

The Implicit Neural Architecture is implemented through a Multi-Layer-Perceptron (MLP) which in its most basic form takes as input the spatial coordinates and outputs the value of the physical field at the specific coordinate. This formulation is naturally suited to fit single signals defined over low-dimensional domains. For machine learning tasks, that require learning classes of objects (eg. a dataset of various PDE solutions, instead of a single element), the INR architecture can be conditioned through a latent representation that encodes sample-specific features. Hence, in the following we proceed to describe: (i) how to train INRs to handle variations in geometry and input parameters, and (ii) how the proposed architectural components make Neural Fields a powerful paradigm for learning mesh-based simulations.

**Input Encoding** Standard Neural Networks fail to represent high frequency oscillations on low dimensional domains[48], being biased towards lower frequency signals. This phenomenon, denominated Spectral Bias, hinders the convergence of vanilla MLP architectures to higher harmonics, leading to blurry reconstructions. The solutions proposed in the literature employ positional encoding[46] or Fourier Feature encoding[48] to the inputs.[48] shows that incorporating these encodings, leads to a stationary Neural Tangent Kernel[53] stationary, thus improving the network convergence to the high frequency component of the signal being approximated. The Fourier Encoding is implemented as follows:

$$\gamma(\mathbf{x}) = [a_1 cos(2\pi b_1^T \mathbf{x}), a_1 sin(2\pi b_1^T \mathbf{x}), ..., a_n cos(2\pi b_n^T \mathbf{x}), a_n sin(2\pi b_n^T \mathbf{x})] \tag{4}$$

where the frequency components are sampled from a zero-centered Gaussian Distribution $b_i \sim \mathcal{N}(0, \sigma)$. The standard deviation ($\sigma$) of the sampling distribution is a hyper-parameter that can have a significant effect on the underfitting-overfitting behavior of the network[48,54]. In fact, its magnitude determines the range of frequency that the network can represent (this is typically referred to as kernel bandwidth) in the reconstructed signals. Low $\sigma$ values lead to the network acting as a low pass filter on the reconstructed signal, causing underfitting behavior as the network fails to capture the details of the target signal. Larger $\sigma$ values, on the other hand, can lead the network to incorporate high frequency patterns, and produce noisy reconstruction characteristics of overfitting behavior. Attempts to learn the optimal value or infer it from the frequency spectrum of the target signal are limited to simple cases in 1D or single signal scenarios. It can be much more complex to tune this hyperparameter to fit a dataset of CFD simulation, exhibiting multiscale phenomena. To circumvent this issue we propose a multiscale architecture where multiple input encoding are performed using distinct values of $\sigma$, passed through the network leading to intermediate outputs that are concatenated through the final layer. This approach, inspired by the work of[54] in the context of PINNs, appears to simplify other proposed methods in the field of computer vision[55,56], based on a progressive increment of $\sigma$ during training. We refer the reader to the corresponding literature for a deeper dive into this topic. The encoding function for each scale $i$ is defined as:

$$\gamma_{\sigma_i}(\mathbf{x}) = \left[\sin(2\pi \mathbf{B}^{(i)}\mathbf{x}), \cos(2\pi \mathbf{B}^{(i)}\mathbf{x})\right]^T, \quad for\ i = 1, 2, \ldots, M \tag{5}$$

where each element of $\mathbf{B} \in \mathbb{R}^{n \times d}$ is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma_i)$. The output of the network, $f_\theta(\mathbf{x})$, is then obtained by passing the concatenated intermediate output at each scale through a final linear layer :

$$f_\theta(\mathbf{x}) = W_{L+1}[H_L^{(1)}, H_L^{(2)}, \ldots, H_L^{(M)}] + b_{L+1}. \tag{6}$$

Note that the hidden layers of the network are shared for all the encoded inputs, and therefore this approach does not considerably increase the parameter count. In Figure 1 a schematic illustration of the multiscale architecture is presented. A simple pedagogical experiment is performed by fitting a single 1D signal with large frequency bandwidth: choosing *a priori* one single value of $\sigma$ is not trivial. Larger values of this hyperparameter (eg. $\sigma = 5$ ) can lead to overfitting behavior as high frequency oscillations are present in the reconstructed signal, while smaller sigma values (eg. $\sigma = 0$) can result in underfitting. The Multiscale architecture ($\sigma = [1, 5]$)leads to better reconstructions, by balancing the two behaviors: this advantage becomes even more relevant when fitting multiple signals in higher dimensions. Conditioning the network to a specific sample is performed through shift modulation of the intermediate layer outputs. The hidden layer outputs are adjusted by a modulation vector $\phi$, which is added to the output of each layer before applying the ReLU activation function. The first layer receives the encoded input $\gamma_{\sigma_i}(\mathbf{x})$. The composition of layers in the network is described as follows:

$$\begin{aligned} H(\mathbf{x}) &= W_L(\eta_{L-1} \circ \eta_{L-2} \circ .... \circ \eta_1 \circ \gamma_\sigma(\mathbf{x})) + b_L \\ \eta_l(\cdot) &= ReLU(W_l(\cdot) + b_l + \phi_l) \end{aligned}$$

where $W_l, b_l \in \theta$ are the weights and biases of the $l$-th layer, and $\phi_l$ is the shift modulation vector for the $l$-th layer. Here we omit the index of the scale encoding. The modulation vectors carry the specific information of each sample and allow to use a unique set of global network parameters to account for all parametric or geometry variations.

**End-to-end framework** In the case where the desired output is a physical field on a fixed geometry, the learning task can be handled end-to-end by minimizing the pointwise distance between the model predictions and the target values across the sampled coordinates and training samples :

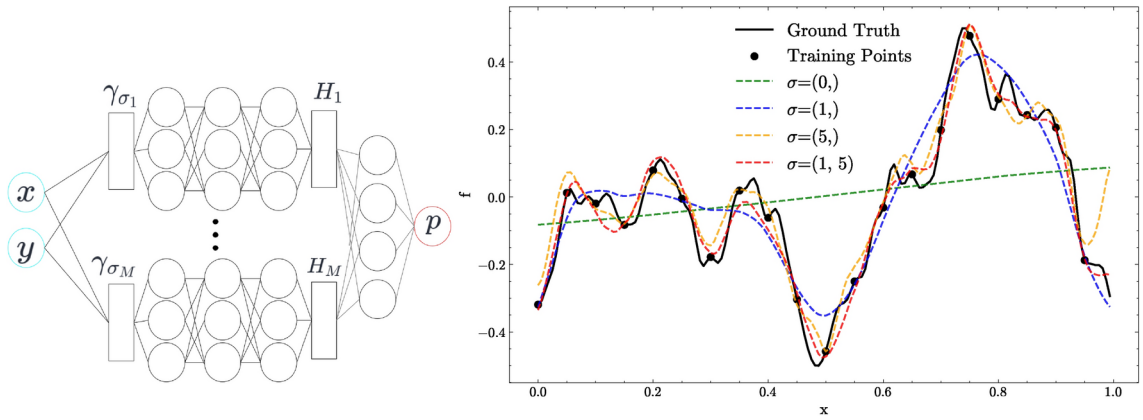


**Fig. 1**. Left: Illustration of the multiscale architecture. Right: Experimental results for fitting a single signal using a 3-layer INR with different Fourier encoding. This experiment can be easily reproduced using the notebook provided in the accompanying Git repository (https://gitlab.isae-supaero.fr/gi.catalani/aero-nepf).

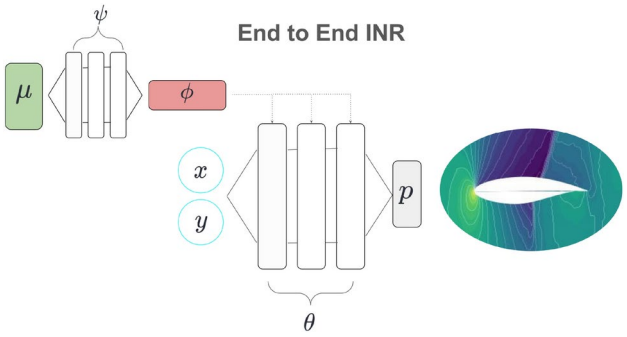


**Fig. 2**. In this setup, the input parameters $\mu$ are directly fed into a hypernetwork $h_\psi$ to produce the modulations $\phi$. This modulation, along with the spatial coordinates, is then processed by a neural network parameterized by $\theta_{out}$ to output the physical fields $p$.

$$\min_{\theta,\psi} \mathcal{L} = \sum_{i}^{N} \sum_{\mathbf{x}\in\mathcal{M}} \ell\left(f_\theta(\mathbf{x}; h_\psi(\mu_i)), u_i(\mathbf{x})\right). \tag{7}$$

where a hypernetwork $h$ of parameters $\psi$ can be used to map the input parameters $\mu$ to the latent modulation vectors $\phi$. Figure 2 illustrates the components of the Implicit end-to-end framework. Potentially, this framework can be equipped with a physics-based loss, in the style of PINNs: however this would require the prediction of all physical quantities (including turbulent viscosity) and knowledge of the explicit formulation used to generate the training CFD data. Additionally, the parametric changes in the inflow conditions should be considered by the physics-based loss[57], with its associated modeling challenges.

The test case for this methodology is a dataset Reynolds Averaged Navier Stokes (RANS) simulations over the two-dimensional transonic RAE2822 airfoil. The dataset, derived from simulations in Fluent using the Spalart-Allmaras turbulence model on a hybrid mesh of 27,499 cells, includes 1200 samples. These simulations cover a Reynolds number typical of cruise flight (6.5 million), with angles of attack ranging from 0° to 9° and Mach numbers from 0.3 to 0.9, encapsulating both weakly compressible and transonic conditions. Each RANS computation requires about 30 minutes on an Intel Core i7 processor. The machine learning task involves developing a surrogate model to predict the output pressure field over the mesh for any combination of Mach number and angle of attack, based on this dataset. We compare the proposed INR end-to-end model with a simple MLP baseline and a state of the art Graph UNET Architecture. A standard Multi-Layer-Perceptron architecture serves as a simple baseline, consisting of a 5-layer MLP with residual connections. For this baseline, the input parameters $\mu$ are directly fed into the network along with the spatial coordinates, making it essentially a basic INR without latent conditioning and input encoding. This baseline helps to isolate and evaluate the specific impact of input encoding and latent modulation on model performance. The Graph UNET (GUNET), is a multiscale message passing graph convolutional neural network introduced in 2019[34] that extends the popular UNET architecture[58] to handle non uniform grids. Graph UNets are designed to handle graph-structured data

by performing hierarchical pooling operations, which reduce the size of the graph at each level, and increase the receptive field of the convolutions, thereby capturing multi-scale features. A multiscale GNN is particularly suited for this dataset, due to the moderately large mesh size, which requires more than local message-passing as in mono-scale architecture. However, we observe that it is possible (and beneficial) to train the model on a downsampled version of the original mesh and then to perform direct inference on the full mesh.

**Encode-Process-Decode framework** This framework, proposed by Dupont et al.[50] and specialized by Serrano et al.[51] for mesh based simulations, splits the learning process into two main steps: firstly, learning compact representation of input and output field quantities ($z_{in}$ and $z_{out}$), and secondly, learning the mapping between the two latent spaces ($z_{out} = MLP(z_{in}, \mu)$ conditioned by the input parameters. The encoding step is analogous for both input and output variables (e.g. for the geometry and the pressure fields). We aim to model the geometry through the Signed Distance Function. During the encoding step, for both input and output variables, latent codes $z$, global networks $\theta$, and hyper-network parameters $\psi$ are jointly optimized using a second-order Meta-Learning approach based on the CAVIA algorithm[59]. The learning task can in fact be split into a sample-specific regression problem: to find the optimal $z$ specific to each distinct sample, and a global regression to learn the optimal values of the global shared networks parameters $(\theta, \psi)$ across the whole dataset samples. A detailed description of the training algorithm can be found in Algorithm 1 and in the provided references. As observed by Serrano[51], this strategy offers several advantages over classical approaches for training conditional Neural Fields: training history is stabilized and the risk of overfitting is reduced, as the latent parameters are re-initialized to zero at each epoch and learned in few gradient steps inside the inner loop, thus acting as a regularization term on the optimized latent vectors. Moreover, this speeds up remarkably the inference time, as only few optimization steps are required to learn representations of the input quantities compared to a classical approach as a larger amount of optimization steps would be required to obtain the latent input representations. Encoding the input and output variables allows to obtain input-output latent vector pairs:

$$\mathcal{D}^{enc} = \{(z_i^{in}, \mu_i), z_i^{out}\}_{i=1}^{M} \quad z_{in} \in \mathbb{R}^{d_{in}}, z_{out} \in \mathbb{R}^{d_{out}}. \tag{8}$$

The regression step entails learning a mapping in the compressed space of latent input vector ($z_{in}$), input parameters ($\mu$), and output latent vectors ($z_{out}$). For this task, we employ train a simple residual MLP $p_\delta$, with SiLu activations, by minimizing the latent loss:

$$\min_{\delta} \sum_{i}^{N} [p_\delta(z_i^{in}, \mu_i) - z_i^{out}]^2 \tag{9}$$

At **inference**, predictions for an unseen shape and flight parameters can be performed using the pretrained Neural Fields and processor network. In particular, the input latent code for an unseen shape can be inferred by minimizing the loss function ($\ell$) between the reconstructed input function (Signed Distance) and the target values:

$$\hat{z}_{test}^{in} = argmin_z \sum_{\mathbf{x} \in \mathcal{M}_{test}} \ell \left( f_{\theta^{in}}(\mathbf{x}; h_{\psi^{in}}(z)), u_{test}(\mathbf{x}) \right) \tag{10}$$

practically, this corresponds to the just performing the inner loop described in Algorithm 1, while keeping the global network parameters fixed. Once the geometry descriptor is obtained, it can be used as an input to the processor network, together with the input parameters, in order to obtain the latent representation of the output variable:

$$\hat{z}_{test}^{out} = p_\delta(z_{test}^{in}, \mu_{test}). \tag{11}$$

Finally, the output fields can be decoded on the spatial domain, by querying the pretrained output Neural Field:

$$\hat{u}_{test} = f_{\theta^{out}}(\mathbf{x}; h_{\psi^{out}}(\hat{z}_{test}^{out})) \tag{12}$$

The inference steps are also sketched in Fig. 3 for a 3D Wing shape.

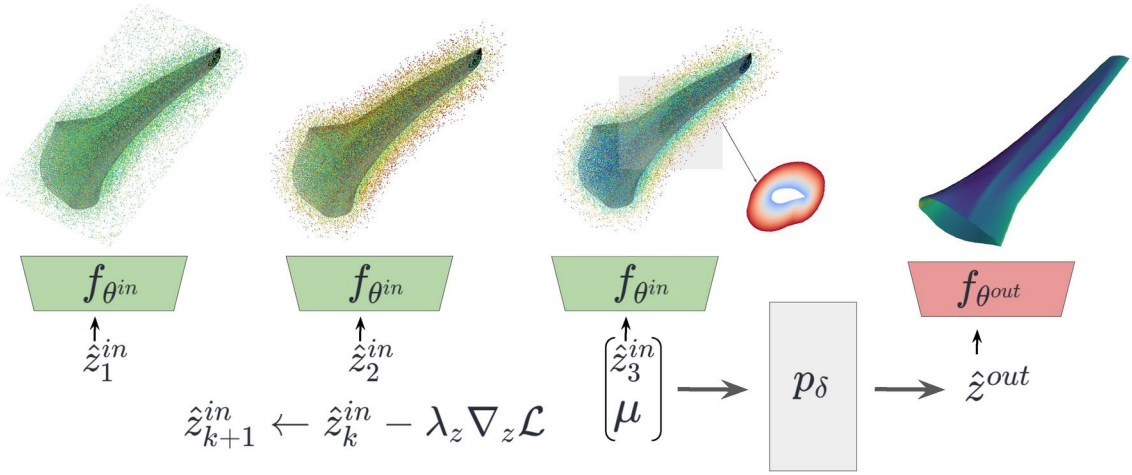


**Fig. 3**. At inference time, few gradient steps are needed to optimize the input shape descriptors $\hat{z}^{in}$ to produce good reconstructions of the Signed Distance Function. The processor network $p_\delta$ maps the input latent and the parameters $\mu$ to the output $\hat{z}_{out}$, which allows decoding the output fields everywhere in the spatial domain.

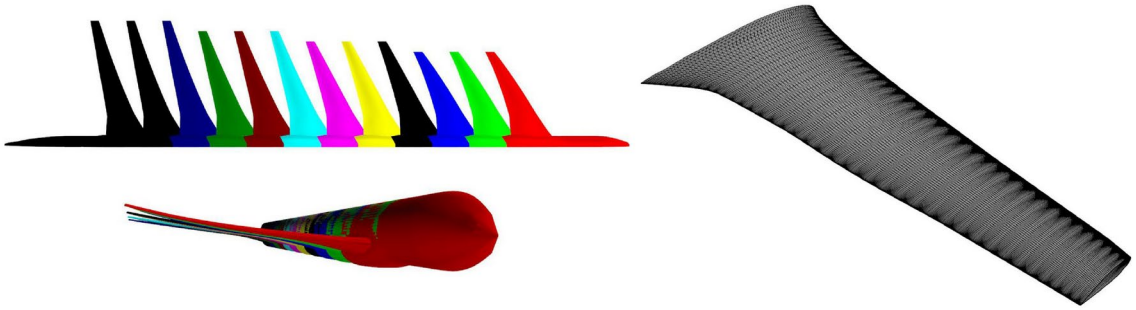

**Fig. 4**. XRF Wing Dataset geometries.Left: Top and front view of 10 out of 120 different wing shapes. Span, shape and thickness are varied.Right: Sample computational mesh for a given shape.

**Input:** Dataset $\mathcal{D}$, epochs $E$, inner loops $K$, batch size $B$, learning rates $\lambda_\theta$, $\lambda_\psi$,$\lambda_z$
**Output:** Optimized parameters $\theta$, $\psi$
**for** epoch $e = 1, \ldots, E$ **do**
  Sample $\mathcal{B}$ from $\mathcal{D}$
  Initialize context parameters $z_i = 0$
  **for** $i \in i_\mathcal{B}$ and $k$ in $1, .., K$ **do** ▷ Inner loop: Optimize local parameters
    Compute sample loss $\mathcal{L}_i = \sum_{\mathbf{x} \in \mathcal{M}_i} \ell\left(f_\theta(\mathbf{x}; h_\psi(z_i)), u_i\right)$
    Update local parameters: $z_i \leftarrow z_i - \lambda_z \nabla_{z_i} \mathcal{L}_i$
  **end for**
  Compute total loss for the batch: $\mathcal{L}_\mathcal{B} = \sum_{i \in i_\mathcal{B}} \mathcal{L}_i$
  Update global parameters: $\theta \leftarrow \theta - \lambda_\theta \nabla_\theta \mathcal{L}_\mathcal{B}$, and $\psi \leftarrow \psi - \lambda_\psi \nabla_\psi \mathcal{L}_\mathcal{B}$
**end for**

**Algorithm 1**. Training Encoder Networks with CAVIA.

We showcase this methodology on the XRF1 Wing Dataset. The dataset comprises 8,640 surface pressure simulations over 120 wing shapes, and different operating conditions. The numerical solver is based on the BLWF method[60] developed at the Central Aerohydrodynamic Institute (TsAGI). This code assumes external inviscid flow coupled with a viscous boundary layer. For each computation, we consider the pressure field on the wing, computed on a triangular mesh of 6,600 nodes. An example of the mesh for a specific shape is displayed in Figure 4. The dataset's diversity and complexity make it highly relevant for industrial aerospace applications. Testing generalization to unseen geometries and configurations in 3D is in fact crucial for a surrogate model to be effectively used within preliminary aerodynamic design and analysis of an aircraft. The dataset spans 120 distinct wing configurations characterized by systematic variations of a reference wing, namely through the Span

*S*,Thickness $T$ and Dihedral Sweep $D_s$. In Figure 4 a subset of wing shapes used in this study is depicted. For each configuration, operational conditions are defined with a simple Design of Experiment (DoE) approach, by changing the Mach number within typical cruise values $M \in [0.50, 0.86]$, the Angle of attack $\alpha \in [-4°, 8°]$ and the Aileron Deflection Angle $\delta_{ail} \in [0°, -15°]$. The Reynolds number is kept fixed at $Re = 3 \times 10^6$. This provides a wide range of aerodynamic conditions, in the compressible and transonic regime, including control surface deflection effects.

The machine learning task consists of predicting the pressure fields on the surface mesh from the operational parameters $\mu = [M, \alpha, \delta_{ail}]$ and the input mesh. It is important to note that here we assume non-parametric geometry definition: although the shape variations are defined by modification of the shape parameters, we decide to learn a representation of the geometry only from the mesh node position, connectivity and surface normals. For this test case, we compare the performance of the INR model, with a Graph Neural Network, specifically the MeshGraphNet, which is widely used for this type of task[23,32]. Training the input INR model requires a preprocessing step of the surface meshes to obtain the volumetric Signed Distance Function fields. We sample a point-cloud around the meshes, following the same process to the one applied to train the DeepSDF model of Park et al.[44]. Finally, the SDF is computed in each of the sampled points, using the point-cloud-utils[61] library. A set of input coordinates and ouptut SDF values is obtained for each shape:

$$\mathcal{D}_i^{SDF} = \{(\mathbf{x}_j, sdf_j)\}_{j=1}^{N^{vol}} \quad i = 1, ..., M \tag{13}$$

The input INR is trained to learn the mapping between coordinates and sdf values, conditioned by latent vectors specific for each geometry $\{z_i^{in}\}_{i=1}^{M}$. A processor network is used to learn the mapping from input latent codes and the flight parameters to output latent codes. The flight parameters are stored in a 3-dimensional vector, containing respectively the Angle of Attack, the Mach Number, the Aileron Deflection. This regressor model is implemented as a 4-Block MLP with skip connections and SiLu activation. Training the processor is performed for a maximum 1000 epochs and stopped when validation loss has not improved for more than 200 epochs. It must be observed that when only surfacic physical fields are available (as in this case) it is not possible to define a Physics-informed methodology on the surface, as the governing equations are formulated to take into account the derivatives of the physical quantities outside the boundary. Therefore, the neural network optimization is totally data-driven. It is worth exploring the idea of enforcing a physics loss in the volume and a purely data-driven loss on the surface.

## Results and discussion

### Transonic RAE2822 Airfoil

The performance of the models on the transonic airfoil dataset is summarized in Table 1. In Figure 5, we observe the performance of the models in capturing the pressure distribution in the spatial domain and on the airfoil surface. The simple MLP approach shows high accuracy away from the shock region, effectively capturing the pressure distribution in the weakly compressible regime. However, due to the spectral bias, and the absence of input encoding, the shock region is oversmooth, failing to capture the sharp gradients accurately. The GUNET model, designed to handle graph-structured data with multi-scale features, performs well in the shock region prediction. The hierarchical pooling operations in GUNET enable it to capture multi-scale features, which helps in modeling the sharp gradients near the shock. However, its overall error is still higher than the MLP-based approaches, notably in the trailing edge region. The INR model outperforms all other approaches, achieving the lowest MSE. The multi-layer MLP with multiscale feature encoding and shift modulation enables the INR to effectively model the complex aerodynamic phenomena present in the dataset. The speedup factor of all tested surrogate models, compared to the high-fidelity RANS computations is in the order of $10^4 - 10^5$, demonstrating the computational gain of using a surrogate model.

Although in this study the focus is on the pressure distribution prediction, the presented models can be used for multi-output predictions: additional experiments are included in the supplementary material section. Moreover, for the estimation of the viscous drag component, the prediction of the skin friction coefficient is required: this quantity is in general hard to predict even for high-fidelity numerical methods as it is a function of the velocity gradient in the boundary layer and of the associated flow regime. In the supplementary material section, we complement this study by presenting the skin friction coefficient predictions using the proposed INR-based model, demonstrating the capability of the methodology to capture this strongly non-linear quantity across the airfoil surface, and consequently the potential to estimate accurately the drag coefficient.

**Discretization invariance analysis** A large class of Deep Learning based reduced order models can only predict test signals at the same resolution of the training signals, and are often coupled with interpolation routines to extract the value of the underlying signals at finer resolutions: this process can perform poorly, as subgrid

| Model | MSE | Training Time (s) | Inference Time (ms) |
|---|---|---|---|
| INR | 0.0020 | 9497 | 4 |
| Vanilla MLP | 0.0057 | 5749 | 2 |
| GUNET | 0.0074 | 17100 | 17 |

**Table 1**. Performance comparison of different models in terms of error (MSE), training time (s), and inference time (ms)

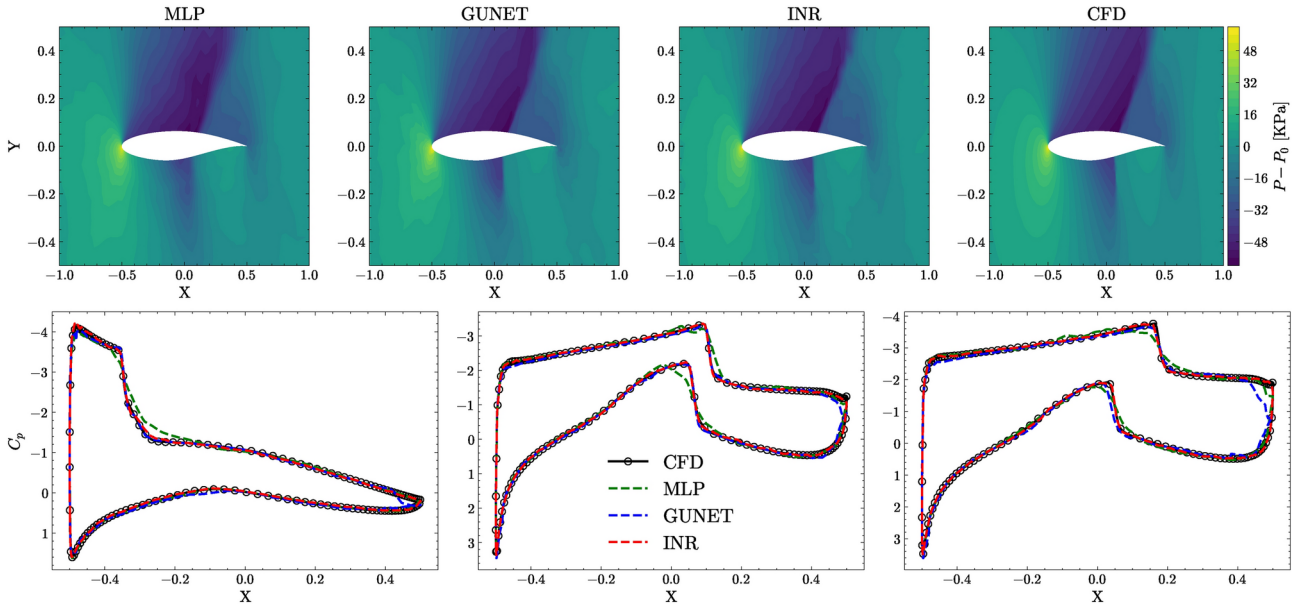


**Fig. 5**. Top: Pressure field distribution prediction at Angle of Attack 4.94 [deg], Mach Number 0.84, comparing different surrogate models with CFD results. Bottom: Surface Pressure Coefficient distribution predictions for different conditions: Left: Angle of Attack [6.2 deg], Mach Number [0.60]. Center: Angle of Attack [4.94 deg], Mach Number [0.84]. Right: Angle of Attack [7.2 deg], Mach Number [0.85], across various surrogate models and compared with CFD.

dynamics is filtered out[62]. Crucially, fluid flows are modeled by PDEs on continuous domains, and numerical methods seeks for the best approximation of the continuous operator mapping the input to the output function spaces on a discretized domain, without the limitation of mapping input and outputs of fixed resolution. An even more compelling consideration regarding surrogate models for CFD must take into account the scalability of the method to handle large mesh simulation, as the ones often used for real aeronautical applications. In these scenarios, surface meshes can be constituted by a number of nodes in the order of 1 Million. This resolution is required in order to accurately model strongly non-linear phenomenon in the boundary layer regions, or pressure discontinuities in the transonic regime. The amount of computational resources, namely memory footprint and training time, can represent the main limitation in training ML-based surrogates for this class of applications. Ideally, surrogate models that can generalize across different resolutions present the advantage of allowing training at much lower resolutions, while being able to predict accurately solutions at full resolution. Implicit Neural Representations are naturally formulated as coordinate based networks, without constraints on the signal discretization, and can perform zero shot superesolution at finer scales at test time. We quantify the trade-off between accuracy and efficiency, by performing an analysis on the approximate discretization invariance of the model, obtained on the RAE2822 dataset. The study is devised as follows: multiple identical INR architecture (we use the end-to-end best performing architecture) are trained on different version of the datasets, obtained by randomly downsampling the mesh nodes at different resolution levels. The models are then used to predict the CFD solutions at increasing resolutions and the global metrics in terms of mean square error are summarized. In Table 2 the quantitative results are presented: the INR can be trained at much lower resolutions (on less than 2 % of the full mesh), and perform super-resolution at test time with a contained error increase. The test error at full resolution converges to a minimum as the training resolution is increased, demonstrating the approximate discretization convergence property of Neural Operators. Optimal trade-off between test error and training time can be achieved by training the model with a 6 times lower resolution than the full mesh.

## XRF1 wing dataset

The performance of the models on the XRF1 wing dataset is summarized in Table 3. The INR framework demonstrates far superior accuracy with the lowest MSE for predictions on unseen shapes. In Figure 6, the surrogate models' predictions are displayed against the ground truth CFD data. The MGN method can capture the general trend of the pressure distribution, while failing to model the more complex aerodynamic features. The shock region prediction is the most challenging for both surrogate models. Nevertheless, the INR method is capable to predict the sharp gradient with good accuracy. Towards the leading edge, the accuracy of MGN degrades, as the pressure distribution shows strongly non-linear pattern due to the presence of strong vorticity. The INR method shows remarkable generalization capabilities to unseen shapes, demonstrating the efficacy of the geometry latent representations. This is visible in Figure 7, as the surrogate models' predictions on three different test shapes, at the same operating conditions, are presented against the CFD ground truth. The inference time, for the proposed INR approach, is dominated by the geometry encoding step : as explained in the Section 2, thanks to the meta-learning approach, the latent codes for a test geometry can be inferred in few gradient steps

| Model | Training Res | Test Res | Average MSE | Training Time (s) |
|---|---|---|---|---|
| INR | 500 | 500 | 0.00294 | 2899 |
| | | Full | 0.00276 | |
| | 5000 | 5000 | 0.00200 | 9497 |
| | | Full | 0.00202 | |
| | Full | Full | 0.00197 | 46985 |
| GUNET (2 levels) | 500 | 500 | 0.0059 | 6080 |
| | | Full | 0.0078 | |
| | 5000 | 5000 | 0.0038 | 17100 |
| | | Full | 0.0074 | |
| | Full | Full | 0.0079 | 35580 |
| GUNET (3 levels) | Full | Full | 0.0113 | 43320 |

**Table 2**. Performance of models at different resolutions. The table shows the average mean squared error (MSE) and training time for INR and GUNET models at various training resolutions. GUNET levels refer to the levels of pooling used in the model

| Model | MSE | Training Time (s) | Inference Time (ms) |
|---|---|---|---|
| INR | 0.008 | 28320 (INR Input) | 108 (Geometry) |
| | | 73560 (INR Output) | |
| | | 3318 (Processor) | 9 (Output) |
| MGN | 0.035 | 90720 | 17 |

**Table 3**. Performance comparison of different models in terms of error (MSE), training time (s), and inference time (ms)

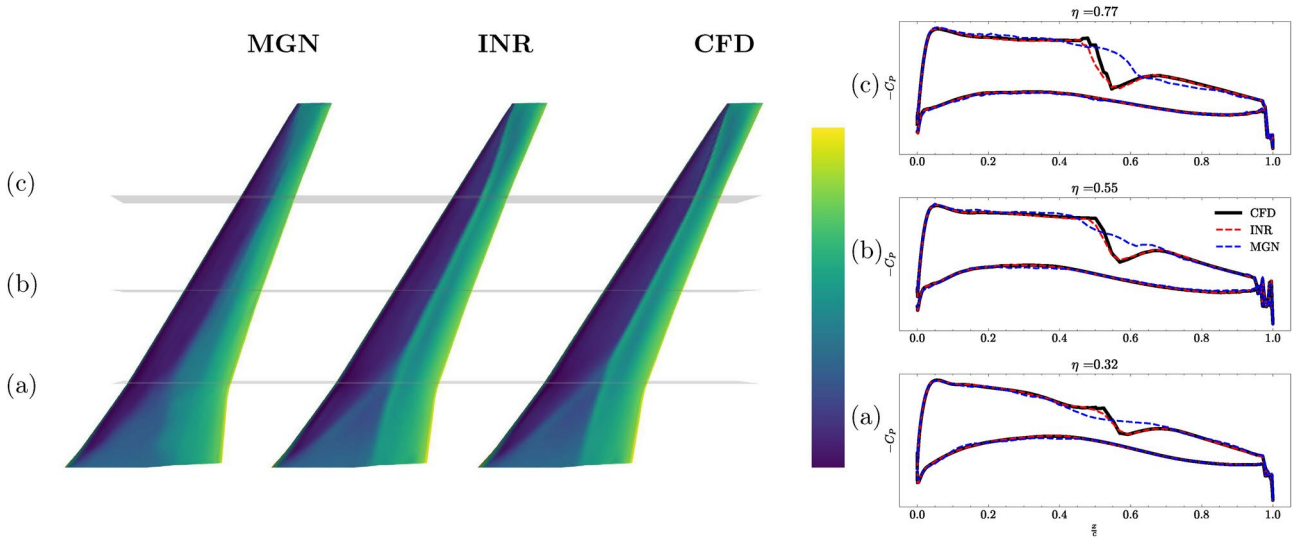


**Fig. 6**. Comparison between MeshGraphNet, INR, and CFD predictions of the surface pressure distribution at $M = 0.82$,$\alpha = 4[deg]$,$\delta_{ail} = -15[deg]$ Left: Surface pressure distribution on the upper surface of the wing (top view). Cutting planes indicate the location of pressure distribution on the right. Right: Pressure coefficient cuts at three different locations $\eta = 0.32, 0.55, 0.77$ along the wingspan.

while freezing the main network parameters. It must be noted that this step is only required once per geometry. Thus, INR model can be used to decode the output fields for different operating conditions, on the same shape, without incurring in this computation. In a typical design scenario, in fact, the same shape design is tested over a variety of flight conditions relevant for the aircraft mission.

**Geometry encoding and reconstruction** A key factor in the effectiveness of the INR methodology is the possibility to effectively encode the geometrical information inside the input latent codes, by fitting a modulated Neural Field on the Signed Distance Function scalar field. In order to capture the shape variations in the

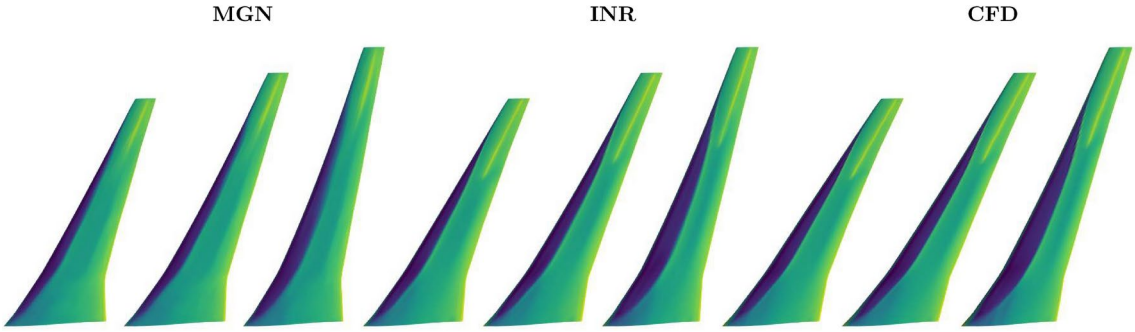


**Fig. 7**. Pressure Distribution prediction on top surface, across different shapes at $M = 0.70$,$\alpha = 8[deg]$ ,$\delta_{ail} = 0[deg]$.

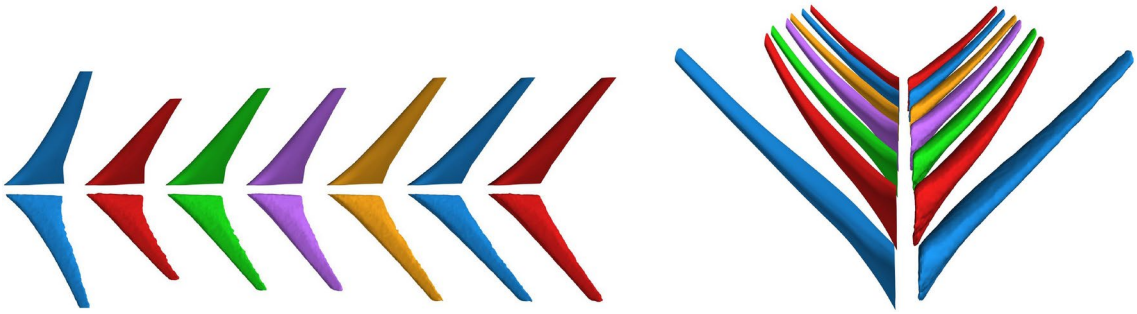

**Fig. 8**. Left: Ground Thruth (top) vs reconstructed (bottom) wing shapes, top view. Right: Ground Thruth (left) vs reconstructed (right) wing shapes, front view.

dataset, a good level of accuracy is required, so that the latent representations unambiguously carry the specific geometrical features of the encoded objects. However, at test time, inferring the latent code for an unseen shape should not be a computationally demanding task, especially for surrogate modeling applications, where the main focus is placed on speedup and accuracy on the predicted physical fields rather than extremely detailed shape reconstruction. For this reason, the employed training strategy detailed in Algorithm 1, which allows to infer the latent codes with a small number of gradient updated ($K = 3$ in all the experiments), fits particularly well the surrogate modeling task. In Figure 8 the original shapes are reconstructed from the inferred latent codes, obtained by fitting the geometry encoder (with frozen parameters) on the SDF fields (see Fig. 3). The marching cube algorithm[63] is applied on the decoded SDF fields to obtain the reconstructed surface meshes. Despite some noisy regions towards the wing trailing edge and wingtip regions, it can be observed that the main geometrical features, determining the dataset shape variations, are well represented.

## Conclusion

In this work, we introduced a robust methodology to build surrogate models for the prediction of aerodynamic fields in 2D and 3D on unstructured meshes that can generalize to unseen non-parametric geometric configurations. The methodology is based on Implicit Neural Representations, enabling a continuous approximation of the target function and approximate discretization invariance. The proposed MultiScale backbone architecture allows optimal signal reconstruction, on complex fluid dynamics dataset, without extensive tuning of frequency hyperparameters. Our experimental results on the Transonic Airfoil and XRF1 Wing datasets demonstrate the advantages of the INR-based models over state-of-the-art Graph Neural Networks (GNNs). The INR models achieve consistently the lowest mean squared error (MSE) across various aerodynamic conditions, showcasing their ability to accurately capture complex flow dynamics, including shock regions and other nonlinear phenomena. The approximate discretization invariance of the INR models allows for training on lower resolution meshes, while maintaining high accuracy at full resolution. This leads to substantial reductions in computational cost and memory footprint. Compared to high-fidelity CFD simulations, the surrogate model can achieve a 5 orders of magnitude speedup, allowing for real-time aerodynamic simulations. This paves the way for building surrogate models of complex 3D industrial simulations on large meshes, without substantial changes to the presented method.

**Limitations and future work:** Despite discussing and experimentally showing the scalability of the method, the experimental section does not report results on large mesh experiments. Due to the scarce availability of open-source datasets of aircraft aerodynamics CFD simulations involving shape variations, this investigation is saved for further work. Additionally, this work only deals with the analysis of single component aerodynamic configurations (i.e. wings and airfoils), while most aircraft aerodynamic applications involve the interaction of

multiple parts, such as fuselage, propellers and high-lift devices. Extending the methodology to these cases is not trivial, as it requires learning geometric representations of components with different length scales, as well as their relative position.

## Data Availability

The code and the data used in the paper can be found at the dedicated GitLab repository (https://gitlab.isae-supaero.fr/gi.catalani/aero-nepf). Due to company policy, the data used for the experiments on the XRF1 Wing cannot be made open source. Readers interested in accessing this data are encouraged to contact the corresponding author directly.

Received: 24 July 2024; Accepted: 18 October 2024
Published online: 26 October 2024

## References

1. Pope, Stephen B. Turbulent flows. *Measurement Science and Technology* **12**(11), 2020–2021 (2001).
2. Da Ronch, A., Ghoreyshi, M. & Badcock, K. J. On the generation of flight dynamics aerodynamic tables by computational fluid dynamics. *Progress in Aerospace Sciences* **47**(8), 597–620 (2011).
3. Berkooz, Gal, Holmes, Philip & Lumley, John L. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics* **25**(1), 539–575 (1993).
4. Chiplunkar, Ankit, Bosco, Elisa, & Morlier, Joseph Gaussian process for aerodynamic pressures prediction in fast fluid structure interaction simulations. In *Advances in Structural and Multidisciplinary Optimization: Proceedings of the 12th World Congress of Structural and Multidisciplinary Optimization (WCSMO12) 12*, 221–233. Springer, 2018.
5. Hijazi, Saddam, Stabile, Giovanni, Mola, Andrea & Rozza, Gianluigi. Data-driven pod-galerkin reduced order model for turbulent flows. *Journal of Computational Physics* **416**, 109513 (2020).
6. San, Omer, Maulik, Romit & Ahmed, Mansoor. An artificial neural network framework for reduced order modeling of transient flows. *Communications in Nonlinear Science and Numerical Simulation* **77**, 271–287 (2019).
7. Lucia, David J., King, Paul I. & Beran, Philip S. Domain decomposition for reduced-order modeling of a flow with moving shocks. *AIAA journal* **40**(11), 2360–2362 (2002).
8. Dupuis, Romain, Jouhaud, Jean-Christophe., Sagaut, Pierre & Aerodynamic data predictions for transonic flows via a machine-learning-based surrogate model. In,. AIAA/ASCE/AHS/ASC Structures. *Structural Dynamics, and Materials Conference, page* **1905**, 2018 (2018).
9. Catalani, Giovanni, Machine learning based local reduced order modeling for the prediction of unsteady aerodynamic loads. *TU Delft*, (2022).
10. Bui-Thanh, Tan, Willcox, Karen, Ghattas, Omar, van Bloemen, Bart & Waanders.,. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics* **224**(2), 880–896 (2007).
11. Casenave, Fabien, Staber, Brian, & Roynard, Xavier. Mmgp: a mesh morphing gaussian process-based machine learning method for regression of physical problems under nonparametrized geometrical variability. *Advances in Neural Information Processing Systems*, 36, (2024).
12. LeCun, Yann, Bengio, Yoshua & Hinton, Geoffrey. *Deep learning. nature* **521**(7553), 436–444 (2015).
13. Bengio, Yoshua, Courville, Aaron & Vincent, Pascal. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **35**(8), 1798–1828 (2013).
14. Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, & Llion, Gomez, Aidan N., Kaiser, Łukasz, Polosukhin, Illia. Attention is all you need. *Advances in neural information processing systems*, 30, (2017).
15. Fresca, Stefania, Dede', Luca & Manzoni, Andrea. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes. *Journal of Scientific Computing* **87**, 1–36 (2021).
16. Eivazi, Hamidreza, Le Clainche, Soledad, Hoyas, Sergio & Vinuesa, Ricardo. Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows. *Expert Systems with Applications* **202**, 117038 (2022).
17. Illarramendi, Ekhi Ajuria, Alguacil, Antonio, Bauerheim, Michaël, Misdariis, Antony, Cuenot, Benedicte, & Benazera, Emmanuel. Towards an hybrid computational strategy based on deep learning for incompressible flows. In *AIAA Aviation 2020 Forum*, 3058, (2020).
18. Ranade, Rishikesh, Hill, Chris, Ghule, Lalit & Pathak, Jay. A composable machine-learning approach for steady-state simulations on high-resolution grids. *Advances in Neural Information Processing Systems* **35**, 17386–17401 (2022).
19. Schmelzer, Martin, Dwight, Richard P. & Cinnella, Paola. Discovery of algebraic reynolds-stress models using sparse symbolic regression. *Flow, Turbulence and Combustion* **104**, 579–603 (2020).
20. Rabault, Jean, & Kuhnle, Alexander. Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Physics of Fluids* 31(9), (2019).
21. Corban, Baptiste, Bauerheim, Michael & Jardin, Thierry. Discovering optimal flapping wing kinematics using active deep learning. *Journal of Fluid Mechanics* **974**, A54 (2023).
22. Zahn, Rebecca, Winter, Maximilian, Zieher, Moritz & Breitsamter, Christian. Application of a long short-term memory neural network for modeling transonic buffet aerodynamics. *Aerospace Science and Technology* **113**, 106652 (2021).
23. Hines, Derrick & Bekemeyer, Philipp. Graph neural networks for the prediction of aircraft surface pressure distributions. *Aerospace Science and Technology* **137**, 108268 (2023).
24. Bertrand, Xavier, Tost, Frédéric, & Champagneux, Steeve. Wing airfoil pressure calibration with deep learning. In *AIAA Aviation 2019 Forum*, 3066, (2019).
25. Baque, Pierre, Remelli, Edoardo, Fleuret, Francois, & Fua, Pascal. Geodesic convolutional shape optimization. In *International Conference on Machine Learning*, 472–481. PMLR, (2018).
26. Wei, Zhen, Yang, Aobo , Li, Jichao, Bauerheim, Michaël, Liem, Rhea Patricia, & Fua, Pascal. Deepgeo: Deep geometric mapping for automated and effective parameterization in aerodynamic shape optimization. In *AIAA AVIATION Forum*, (2024).
27. Wei, Zhen, Dufour, Edouard, Pelletier, Colin, Fua, Pascal, & Bauerheim, Michaël. Diffairfoil: An efficient novel airfoil sampler based on latent space diffusion model for aerodynamic shape optimization. In *AIAA AVIATION Forum*, (2024).
28. Thuerey, Nils, Weißenow, Konstantin, Prantl, Lukas & Xiangyu, Hu. Deep learning methods for reynolds-averaged navier-stokes simulations of airfoil flows. *AIAA Journal* **58**(1), 25–36 (2020).
29. Guo, Xiaoxiao, Li, Wei, & Iorio, Francesco. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 481–490, (2016).
30. Catalani, Giovanni et al. A comparative study of learning techniques for the compressible aerodynamics over a transonic rae2822 airfoil. *Computers & Fluids* **251**, 105759 (2023).
31. Bronstein, Michael M., Bruna, Joan, LeCun, Yann, Szlam, Arthur & Vandergheynst, Pierre. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* **34**(4), 18–42 (2017).

32. Pfaff, Tobias, Fortunato, Meire, Sanchez-Gonzalez, Alvaro, & Battaglia, Peter W. Learning mesh-based simulation with graph networks. arXiv preprint arXiv:2010.03409, (2020).
33. Fortunato, Meire, Pfaff, Tobias, Wirnsberger, Peter, Pritzel, Alexander, & Battaglia, Peter. Multiscale meshgraphnets. arXiv preprint arXiv:2210.00612, (2022).
34. Gao, Hongyang, & Ji, Shuiwang, Graph u-nets. In *international conference on machine learning*, 2083–2092. PMLR, (2019).
35. Lino, Mario, Fotiadis, Stathi, Bharath, Anil A., & Cantwell, Chris D. Multi-scale rotation-equivariant graph neural networks for unsteady eulerian fluid dynamics. *Physics of Fluids*, 34(8), (2022).
36. Rusch, T Konstantin, Bronstein, Michael M., Mishra, Siddhartha. A survey on oversmoothing in graph neural networks. arXiv preprint arXiv:2303.10993, (2023).
37. Li, Zongyi, Kovachki, Nikola, Choy, Chris, Li, Boyi, Kossaifi, Jean, Otta, Shourya, Nabian, Mohammad Amin, Stadler, Maximilian, Hundt, Christian, & Azizzadenesheli, Kamyar. et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36 (2024).
38. Vassberg, John, Dehaan, Mark, Rivers, Melissa, Wahls, Richard. Development of a common research model for applied cfd validation studies. In *26th AIAA applied aerodynamics conference*, 6919 (2008).
39. Li, Zongyi, Kovachki, Nikola, Azizzadenesheli, Kamyar, Liu, Burigede, Bhattacharya, Kaushik, Stuart, Andrew, & Anandkumar, Anima. Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895, (2020)
40. Lu, Lu, Jin, Pengzhan, & Karniadakis, George Em. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. arXiv preprint arXiv:1910.03193, (2019).
41. He, Junyan, Koric, Seid, Abueidda, Diab, Najafi, Ali, & Jasiuk, Iwona. Geom-deeponet: A point-cloud-based deep operator network for field predictions on 3d parameterized geometries. arXiv preprint arXiv:2403.14788, (2024).
42. Raissi, Maziar, Perdikaris, Paris & Karniadakis, George E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* **378**, 686–707 (2019).
43. George Em Karniadakis. Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. *Physics-informed machine learning. Nature Reviews Physics* **3**(6), 422–440 (2021).
44. Park, Jeong Joon, Florence, Peter, Straub, Julian, Newcombe, Richard, Lovegrove, Steven. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 165–174 (2019).
45. Chen, Zhiqin, Zhang, Hao. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5939–5948 (2019).
46. Mildenhall, Ben et al. Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021).
47. Sitzmann, Vincent, Martel, Julien, Bergman, Alexander, Lindell, David & Wetzstein, Gordon. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems* **33**, 7462–7473 (2020).
48. Tancik, Matthew et al. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems* **33**, 7537–7547 (2020).
49. Tancik, Matthew, Mildenhall, Ben, Wang, Terrance, Schmidt, Divi, Srinivasan, Pratul P., Barron, & Jonathan T. Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2846–2855 (2021).
50. Dupont, Emilien, Kim, Hyunjik, Eslami, SM, Rezende, Danilo, & Rosenbaum, Dan. From data to functa: Your data point is a function and you can treat it like one. arXiv preprint arXiv:2201.12204, (2022).
51. Serrano, Louis, Boudec, Lise Le, Koupaï, Armand Kassaï, Wang, Thomas X., Yin, Yuan, Vittaut, Jean-Noël , & Gallinari, Patrick. Operator learning with neural fields: Tackling pdes on general geometries. *Advances in Neural Information Processing Systems*, 36, (2024).
52. Serrano, Louis, Migus, Leon, Yin, Yuan, Mazari, Jocelyn Ahmed, & Gallinari, Patrick. Infinity: Neural field modeling for reynolds-averaged navier-stokes equations. In *Workshop on Synergy of Scientific and Machine Learning Modeling (ICML 2023)*, (2023).
53. Jacot, Arthur, Gabriel, Franck, & Hongler, Clément. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, (2018).
54. Wang, Sifan, Wang, Hanwen & Perdikaris, Paris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering* **384**, 113938 (2021).
55. Hertz, Amir, Perel, Or., Giryes, Raja, Sorkine-Hornung, Olga & Cohen-Or, Daniel. Sape: Spatially-adaptive progressive encoding for neural optimization. *Advances in Neural Information Processing Systems* **34**, 8820–8832 (2021).
56. Landgraf, Zoe, Hornung, Alexander Sorkine, & Cabral, Ricardo Silveira. Pins: progressive implicit networks for multi-scale neural representations. arXiv preprint arXiv:2202.04713, (2022).
57. Wassing, Simon, Langer, Stefan & Bekemeyer, Philipp. Physics-informed neural networks for parametric compressible euler equations. *Computers & Fluids* **270**, 106164 (2024).
58. Ronneberger, Olaf , Fischer, Philipp, & Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer, (2015).
59. Zintgraf, Luisa, Shiarli, Kyriacos, Kurin, Vitaly, Hofmann, Katja, & Whiteson, Shimon. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, 7693–7702. PMLR, (2019).
60. Zhang, Ke-shi, & Hepperle, Martin. Evaluation of the blwf code-a tool for the aerodynamic analysis of transonic transport aircraft configurations. (2010).
61. Williams, Francis. Point cloud utils, (2022). https://www.github.com/fwilliams/point-cloud-utils.
62. Azizzadenesheli, Kamyar, Kovachki, Nikola, Li, Zongyi, Liu-Schiaffini, Miguel, Kossaifi, Jean, & Anandkumar, Anima. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 1–9 (2024).
63. Lorensen, William E., & Cline, Harvey E. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, 347–353 (1998).


## Acknowledgements

This was was supported in part by the French Government represented by Agence Nationale de la Recherche (ANRT), through CIFRE PhD Fellowship sponsored by Airbus Operations SAS and ISAE-Supaero.

## Author contributions

Giovanni Catalani: conception of the study, performed experiments, data generation, manuscript redaction. Siddhant Agarwal: conception of the study, guidance, manuscript revision, industrial research support. Xavier Bertrand: guidance, manuscript revision, industrial research support. Frederic Tost: guidance, manuscript revision, industrial research support. Michael Bauerheim: guidance, manuscript revision, academic research

support. Joseph Morlier: guidance, manuscript revision, academic research support.

## Declarations

## Competing interests
The authors declare no competing interests.

## Additional information
**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-76983-w.

**Correspondence** and requests for materials should be addressed to G.C.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.