Learning physical simulations with Neural Fields

Ellis Machine Learning Insights Seminar

PhD Student: Giovanni Catalani
PhD Advisors: Joseph Morlier (ISAE-Supaero, ICA), Michael Bauerheim (ISAE-Supaero), Xavier Bertrand (Airbus)
TU Delft 26/11/2024

Agenda

- **Introduction.**
  - Physical Simulations & Computational Fluid Dynamics
  - Limitations and need for data-driven physical simulators.
- **Neural Operators**
  - Working principles and Fourier Neural Operator.
  - Implicit Neural Representations.
- **Applications for aircraft aerodynamics.**

Introduction: Computational Fluid Dynamics

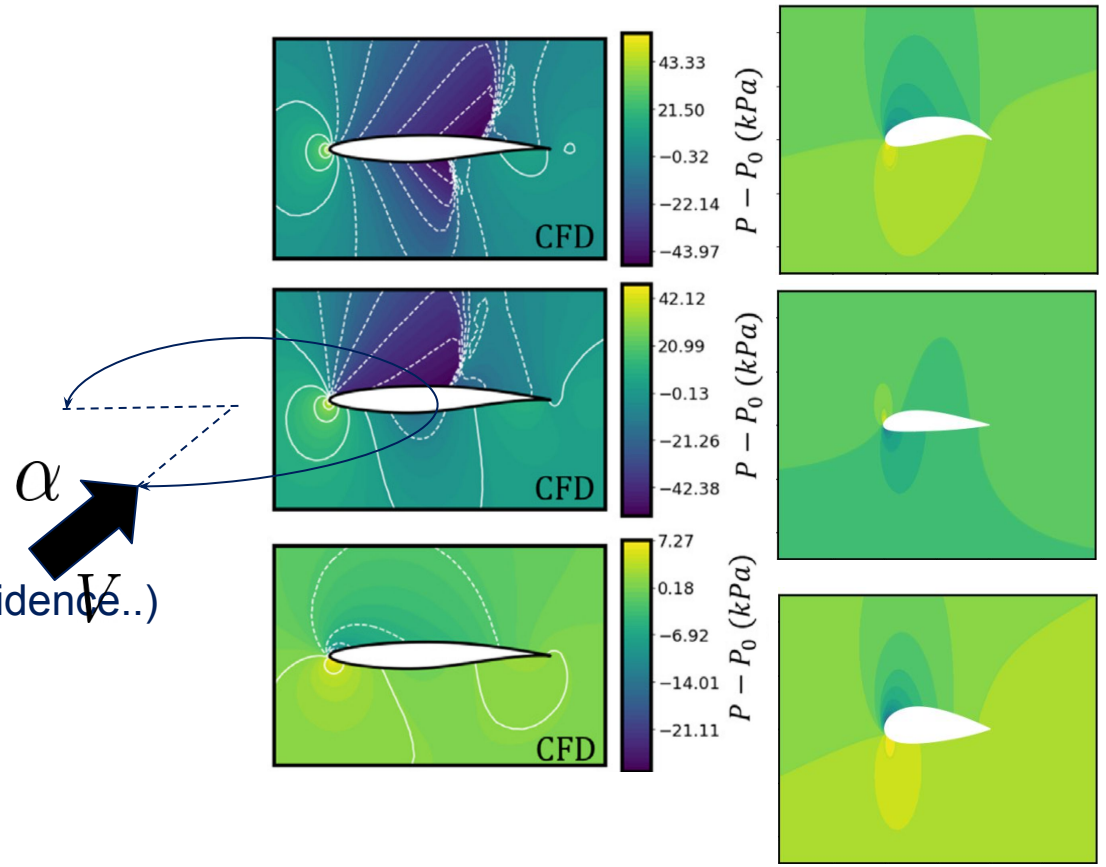
Computational Fluid Dynamic is used for:

- Aircraft aerodynamic design.
- Automotive and race car design.
- Weather modeling and forecast.
- Heart and biological systems simulation.
- ……

The resulting flowfield typically depends on:

- The governing physics (the PDE model used).
- The parametric and inflow conditions (velocity, angle of incidence..)
- The boundary conditions (the shape of the domain).

Parametric Variations (speed, inflow angle.) **Geometry Variations (Boundary)**



Bonnet, Florent, et al. "Airfrans: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier–stokes solutions." *Advances in Neural Information Processing Systems* 35 (2022): 23463-23478.

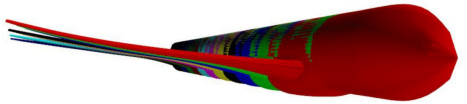
Catalani, Giovanni, et al. "A comparative study of learning techniques for the compressible aerodynamics over a transonic RAE2822 airfoil." *Computers & Fluids* 251 (2023): 105759.

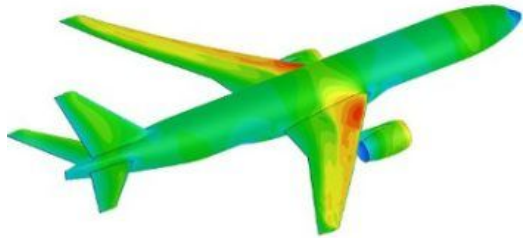
Introduction: Computational Fluid Dynamics

In order to solve the flowfield we need high mesh resolutions:

- For large domains in 3D this translates to **millions of mesh nodes**.
- A single CFD computation can take days.
- Multiple CFD computations are needed to capture all possible flight conditions.
- Multiple geometries and design need to be simulated.

Real Industrial applications: 3D, shape variations…





Introduction: Computational Fluid Dynamics

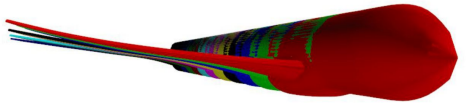
In order to solve the flowfield we need high mesh resolutions:

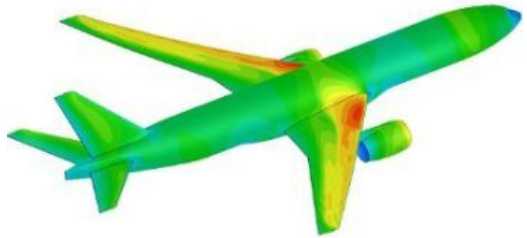
- For large domains in 3D this translates to millions of mesh nodes.
- A single CFD computation can take days.
- Multiple CFD computations are needed to capture all possible flight conditions.
- Multiple geometries and design need to be simulated.

Ideally, we want real time physical simulations :

- Leverage past simulations to train data-driven simulators of the physics.
- Trade-off: **accuracy** vs **speed.**

Real Industrial applications: 3D, shape variations…





Introduction: Computational Fluid Dynamics

In order to solve the flowfield we need high mesh resolutions:

- For large domains in 3D this translates to millions of mesh nodes.
- A single CFD computation can take days.
- Multiple CFD computations are needed to capture all possible flight conditions.
- Multiple geometries and design need to be simulated.

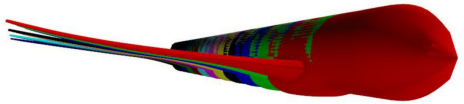
Ideally, we want real time physical simulations :

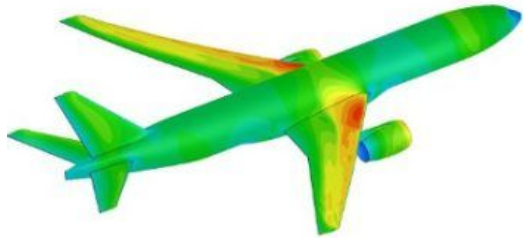
- Leverage past simulations to train data-driven simulators of the physics.
- Trade-off: **accuracy** vs **speed.**

Challenges: physical simulations data is

- High Dimensional.
- Unstructured.
- Strongly non-linear.
- Changing underlying domain and geometries.

Real Industrial applications: 3D, shape variations…





Introduction: Computational Fluid Dynamics

In order to solve the flowfield we need high mesh resolutions:

- For large domains in 3D this translates to millions of mesh nodes.
- A single CFD computation can take days.
- Multiple CFD computations are needed to capture all possible flight conditions.
- Multiple geometries and design need to be simulated.

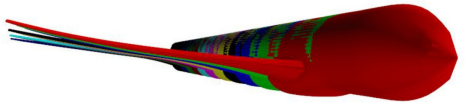
Ideally, we want real time physical simulations :

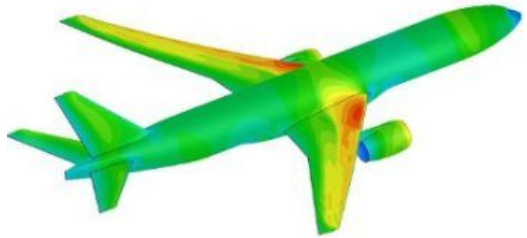
- Leverage past simulations to train data-driven simulators of the physics.
- Trade-off: **accuracy** vs **speed.**

Challenges: physical simulations data is

- High Dimensional.
- Unstructured.
- Strongly non-linear.
- Changing underlying domain and geometries.

Real Industrial applications: 3D, shape variations…





⇒ Need new class of ML architectures to learn physical simulations

Operator Learning

Given input/output infinite dimensional function spaces $\mathcal{A}, \mathcal{U} \subset \mathcal{F}(\Omega, \mathbb{R})$ defined on a physical domain $\Omega \subset \mathbb{R}^d$

We aim to learn an approximation for the true PDE operator mapping functions defined on those function spaces:

$$G^\dagger : \mathcal{A} \to \mathcal{U}$$

The goal is to find a parameterized approximation of the underlying operator $G_\theta : \mathcal{A} \to \mathcal{U} \;\; \theta \in \Theta$ such that :

$$G_\theta \approx G^\dagger$$

Given a probability measure on the data $a \sim \mu$ we want to minimize the approximation error:

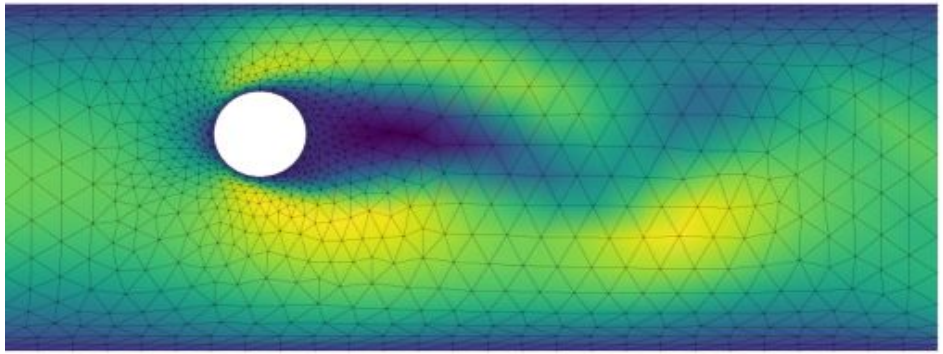
$$\|G^\dagger - G_\theta\|^2_{L^2_\mu(\mathcal{A},\mathcal{U})} := \int_{\mathcal{A}} \|G^\dagger(a) - G_\theta(a)\|^2_{\mathcal{U}} d\mu(a)$$

Practically, in a supervised setting given a set of input and output samples $a_j, u_j = G^\dagger(a_j)$

$$\theta^\dagger = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{j=1}^{N} \|u_j - G_\theta(a_j)\|^2_U$$

Operator Learning: learning from discrete data.

Typically we know the values of the input and output functions in a **finite set of points** on the spatial domain $D = \{x_1, ..., x_n\} \subset \Omega$



However we want to approximate the underlying continuous operator **independently of the training discretization:**

- The learned operator converges to the continuum operator as discretization is increased (**discretization convergence**)
- The learned representations are the same across different discretizations of the same input/output functions.

Operator Learning: Desirable Features

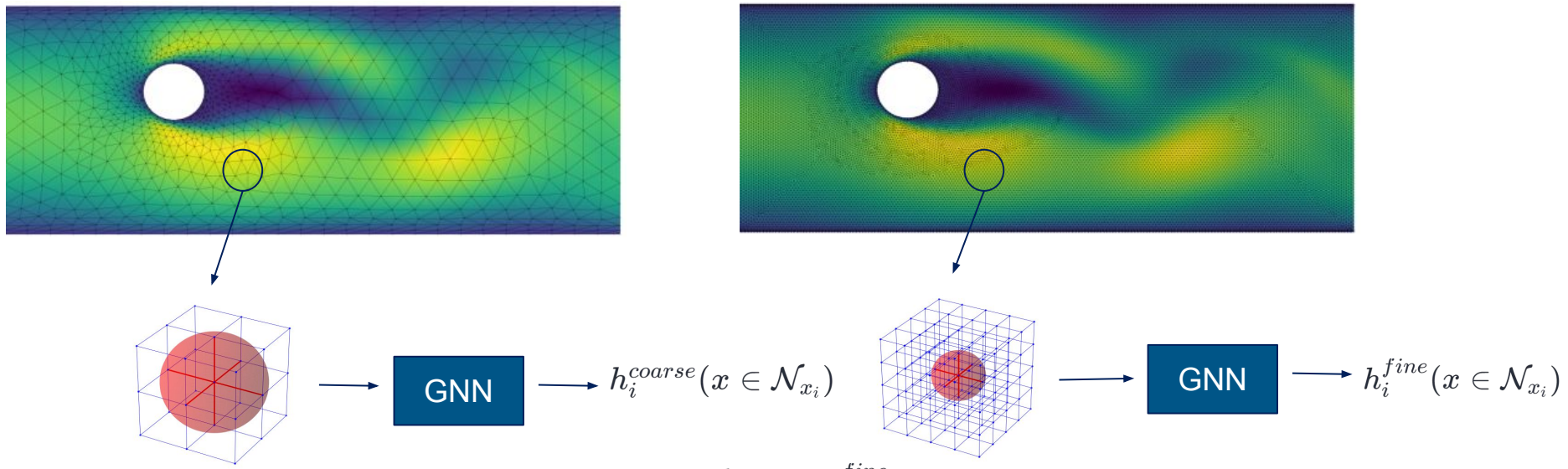
The approximated operator model is a function: it can be queried anywhere in the input domain.

The approximated model can be trained at different discretizations.

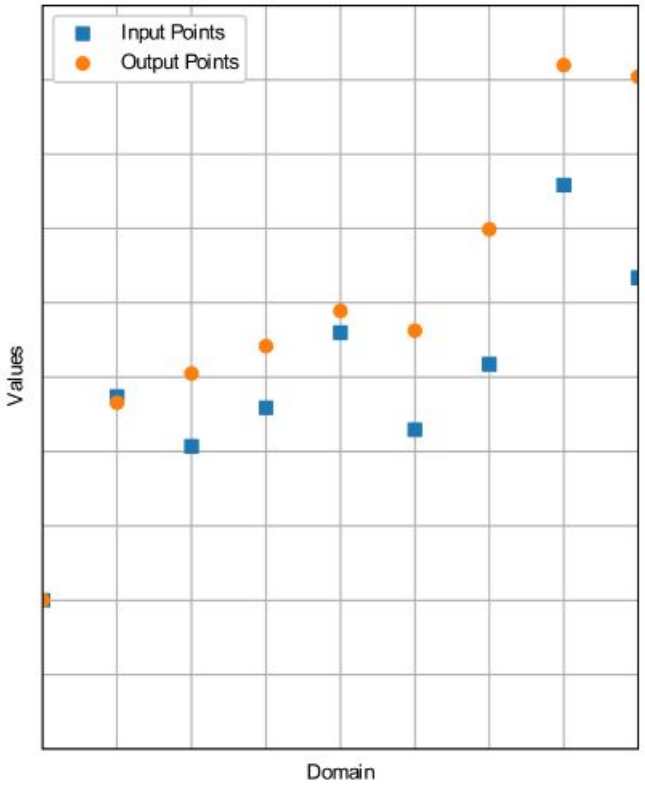
The learned operator converges to the continuum operator as discretization is increased (discretization convergence).

The learned representations are the same across different distcetizations of the same input/output functions.

Example: Message Passing Graph Neural Networks are not discretization invariant:



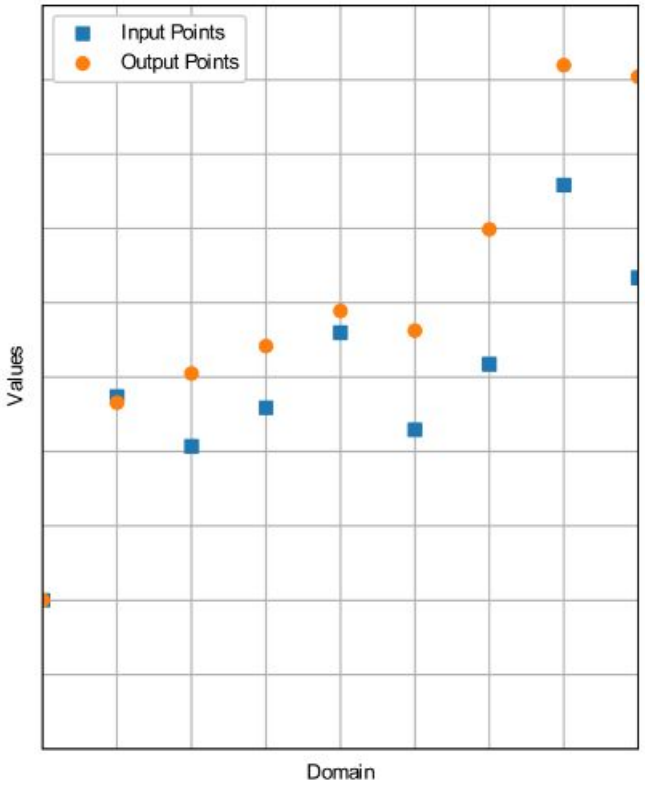
Neural Operators - Learning continuous representations



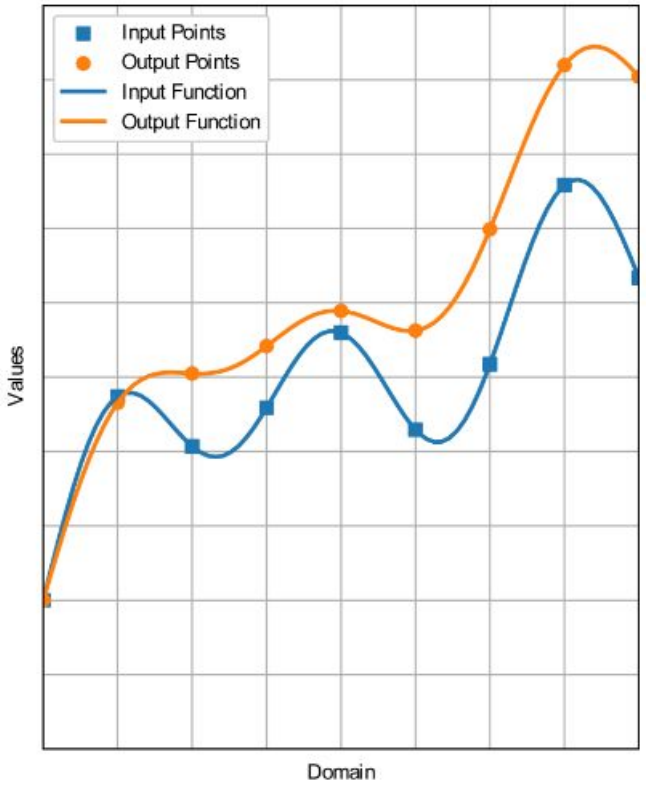
Traditional NN learns a mapping between input and output points on a fixed, discrete grid.

Azizzadenesheli, Kamyar, et al. "Neural operators for accelerating scientific simulations and design." *Nature Reviews Physics* (2024): 1-9.

Neural Operators - Learning continuous representations



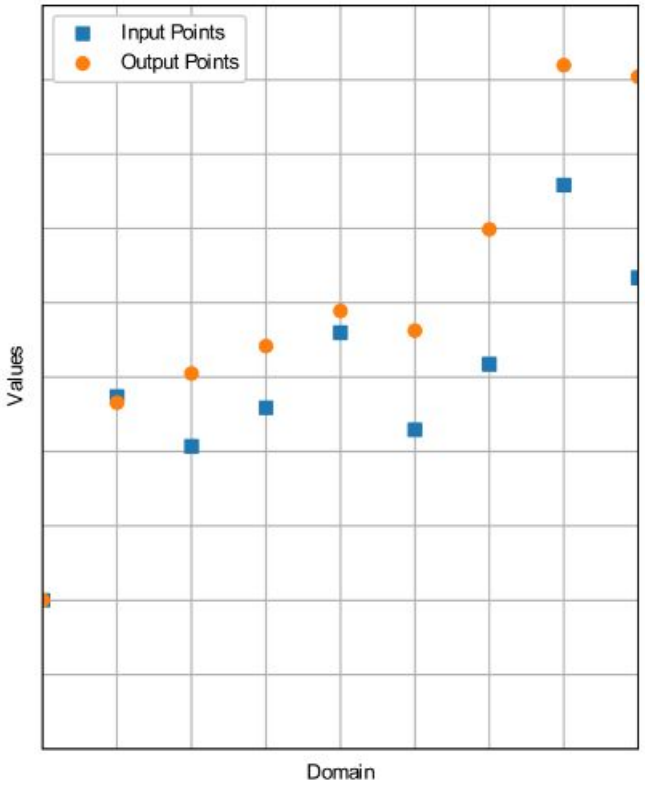
Traditional NN learns a mapping between input and output points on a fixed, discrete grid.



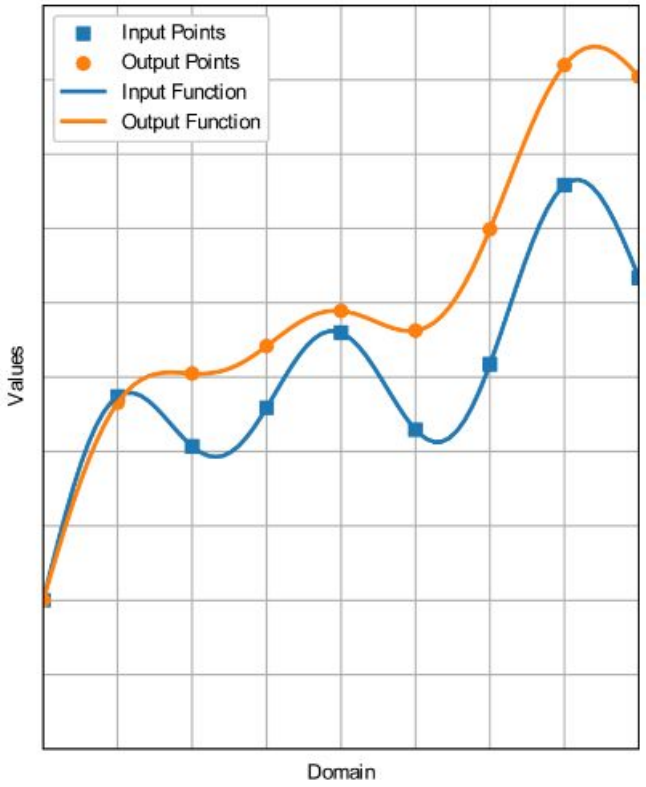
A Neural Operator maps between functions on continuous domain even if the data is defined on a fixed grid

Azizzadenesheli, Kamyar, et al. "Neural operators for accelerating scientific simulations and design." *Nature Reviews Physics* (2024): 1-9.

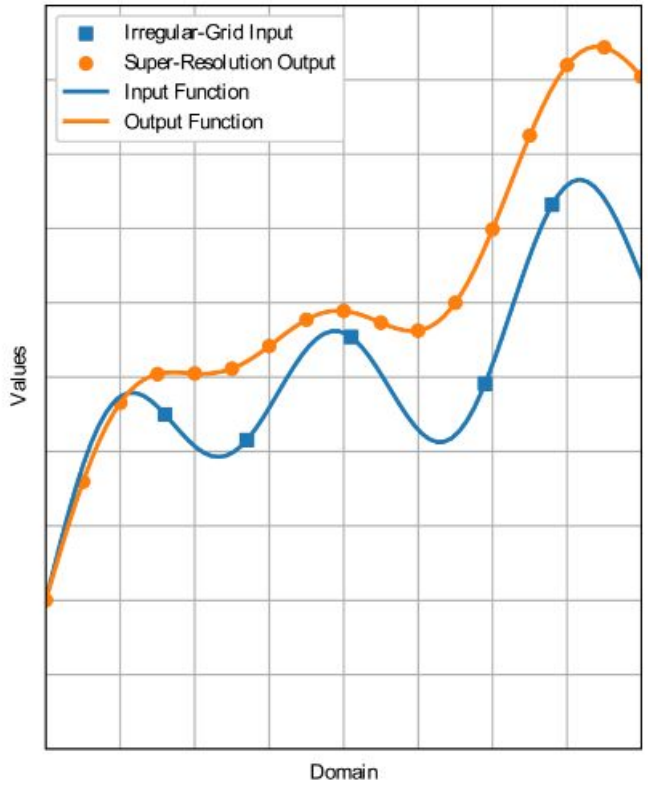
Neural Operators - Learning continuous representations



Traditional NN learns a mapping between input and output points on a fixed, discrete grid.



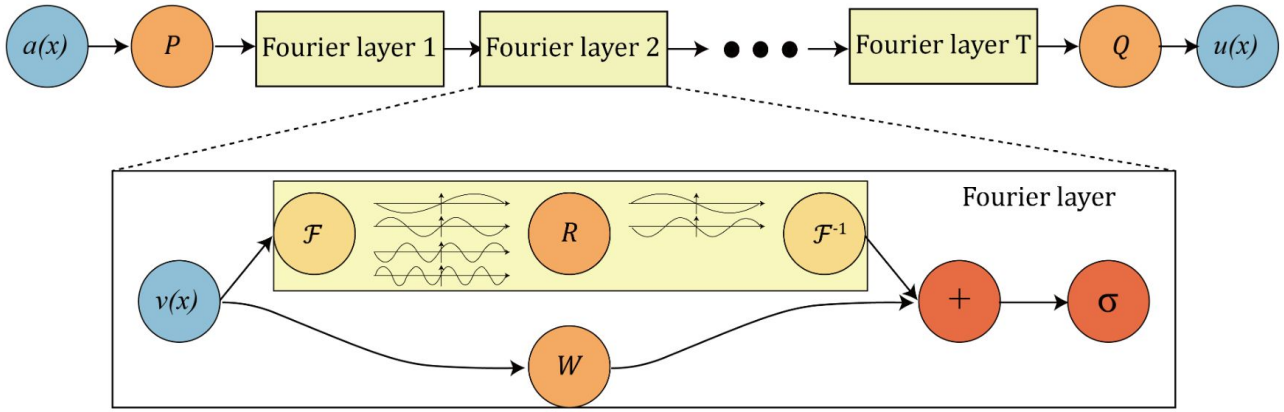
A Neural Operator maps between functions on continuous domain even if the data is defined on a fixed grid

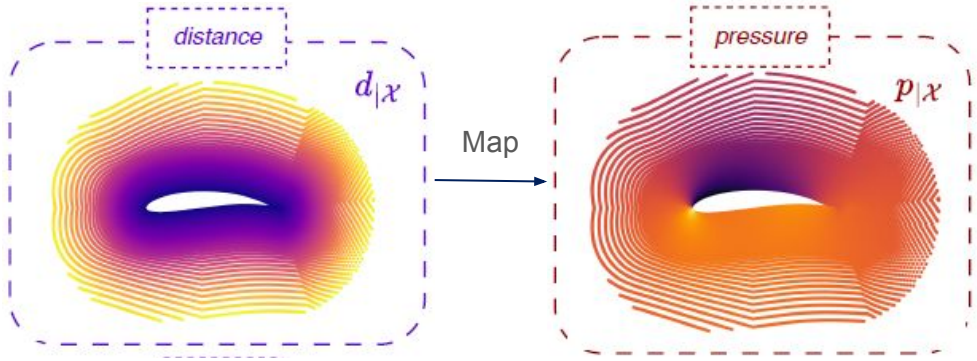


A Neural Operator can perform superresolution at test time.

Azizzadenesheli, Kamyar, et al. "Neural operators for accelerating scientific simulations and design." *Nature Reviews Physics* (2024): 1-9.

Fourier Neural Operator





Methodology:

- Continuous representation obtained with the Fourier Transform of the signals.
- A parametric map is learnt between the input and output truncated Fourier representations.

Results:

- Resolution flexibility: train on coarse meshes, test on fine mesh without loss of accuracy.
- Preserve physical meaning across resolutions.

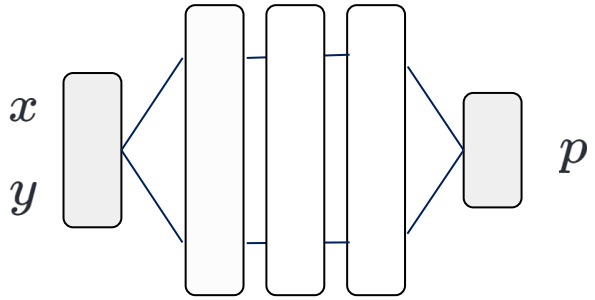
Limitations:

- Computing Fourier Transform on irregular grids is not efficient as FFT cannot be used.

Li, Zongyi, et al. "Fourier neural operator for parametric partial differential equations." *arXiv preprint arXiv:2010.08895* (2020).

Implicit Neural Representations: continuous representation of data.

Neural Networks can be used as a continuous approximation of signals on general domains: the value of the signal at any spatial input location can be obtained as the output of an **Implicit Neural Representation** (INRs).



Training a Neural Network: finding the optimal parameters $\theta = \{W_l, b_l\}_{l=1,..,n_l}$ that minimize the reconstruction error

$$\{(x_i, y_i), p_i)\}_{i=1,...N} \longrightarrow p = f_\theta(x, y)$$

Once we fit a signal with a Neural Network, we can query the Network at any spatial location: we have a **continuous representation.**

Implicit Neural Representations

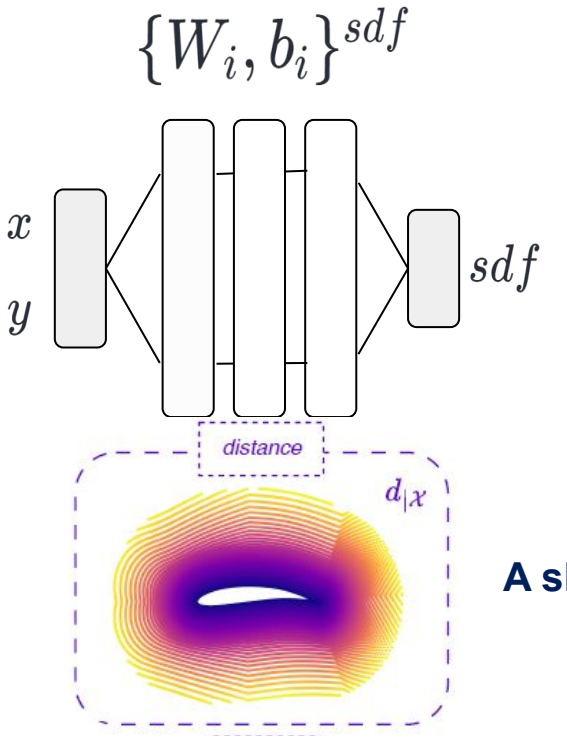
By learning **continuous representation of the data**, we can formulate the operator learning task in the Neural Field weight space.

For physical simulation we are interested in solving PDEs on different geometries (shapes).

Implicit Neural Representations

By learning **continuous representation of the data**, we can formulate the operator learning task in the Neural Field weight space.

For physical simulation we are interested in solving PDEs on different geometries (shapes).

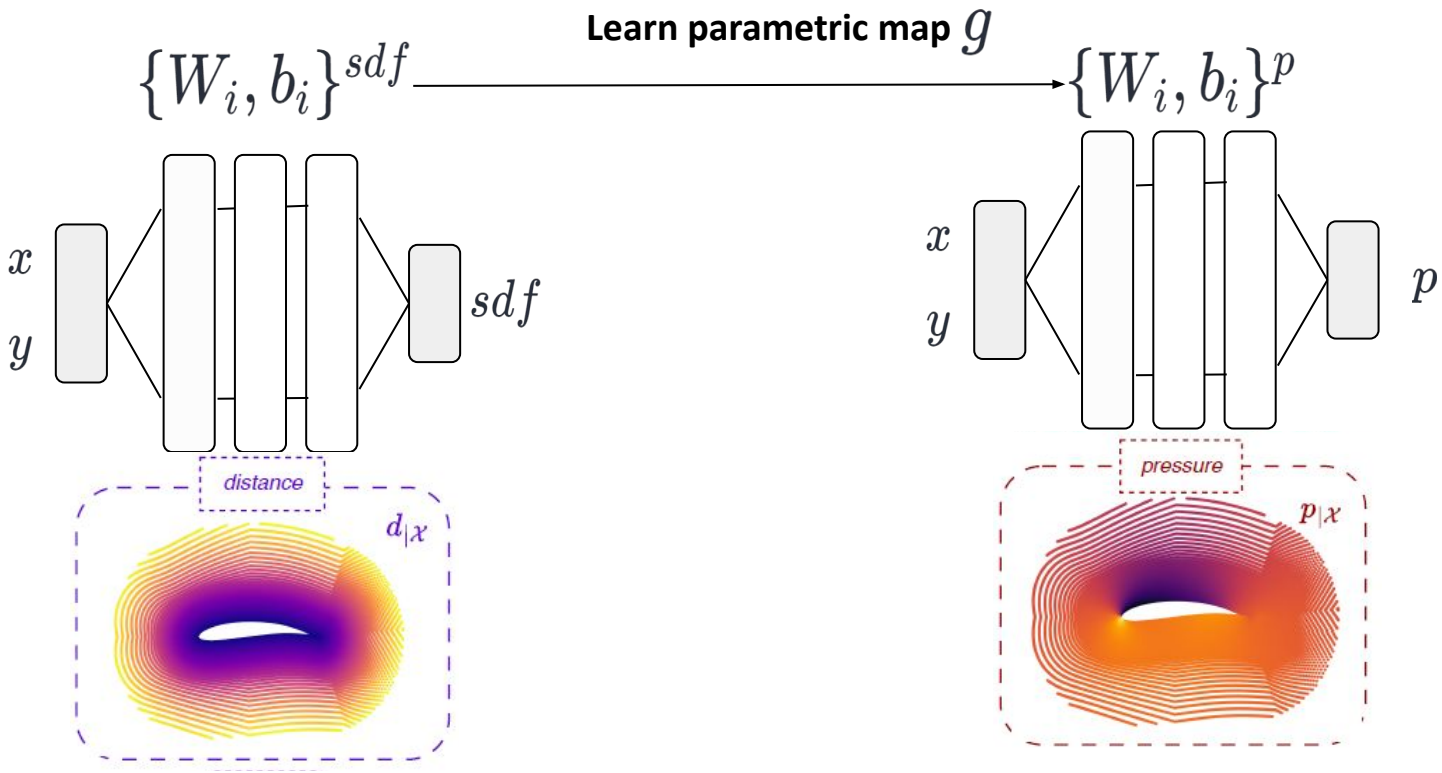


A shape is conveniently describe in space by a distance function field.

Implicit Neural Representations

By learning **continuous representation of the data**, we can formulate the operator learning task in the Neural Field weight space.

For physical simulation we are interested in solving PDEs on different geometries (shapes).

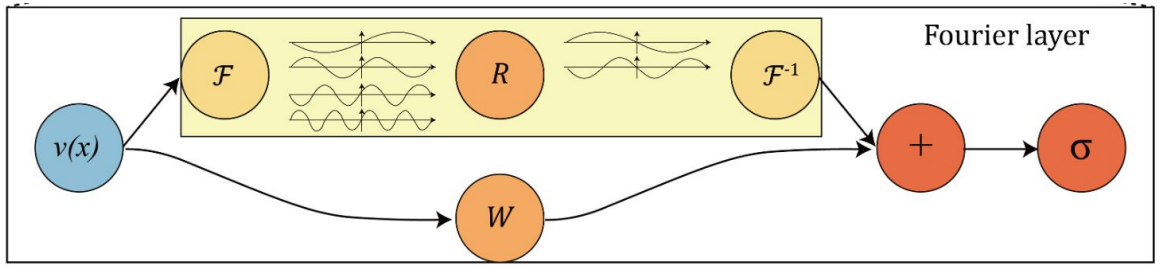


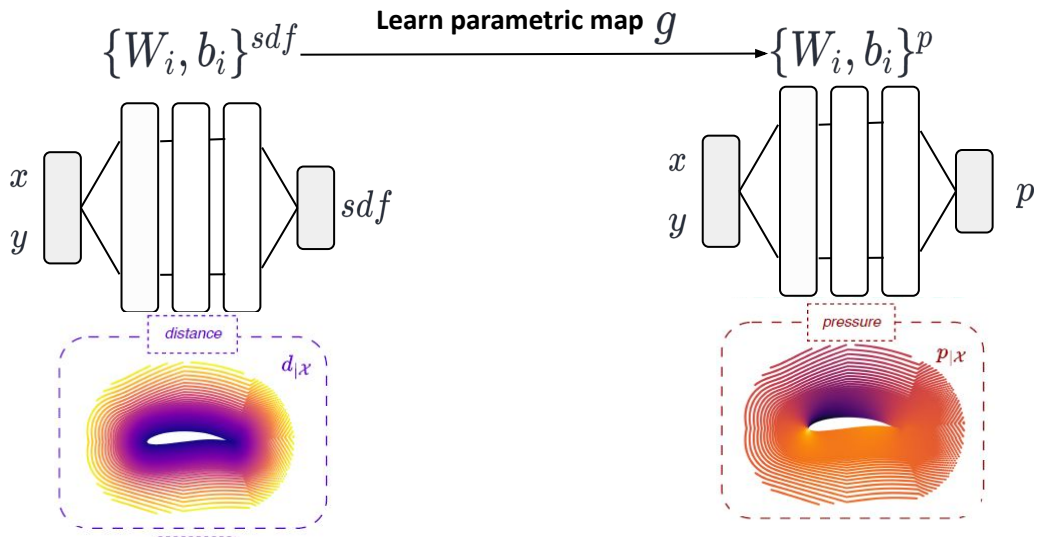
Implicit Neural Representations

By learning **continuous representation of the data**, we can formulate the operator learning task in the Neural Field weight space.

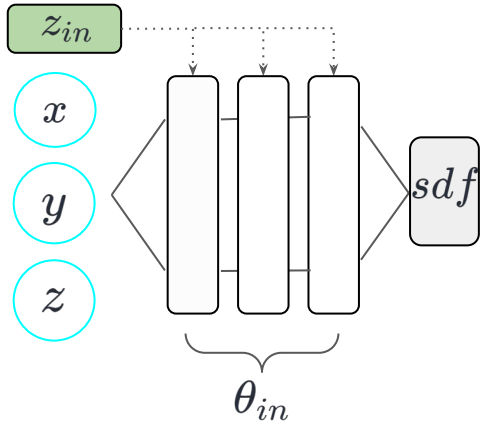
For physical simulation we are interested in solving PDEs on different geometries (shapes).

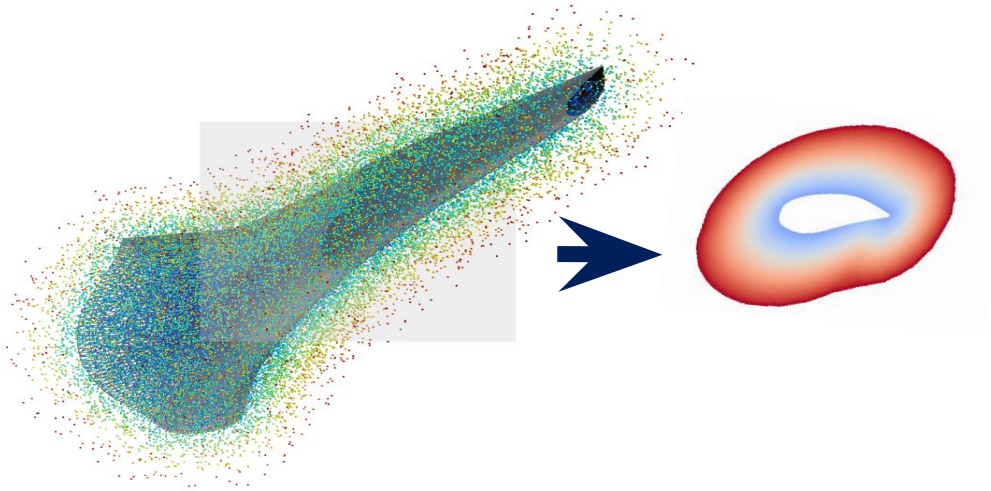
Comparison with FNO: Neural Field weights vs Fourier Modes



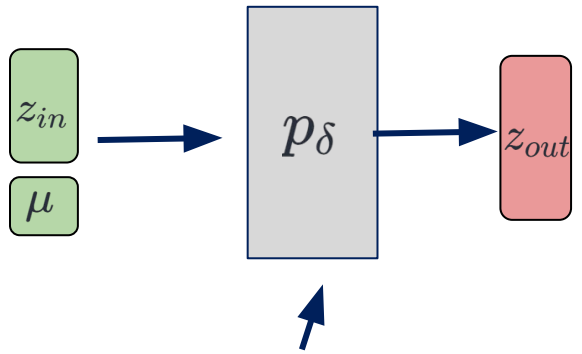


Neural Field Encoder for the Geometry: Learning the Signed Distance Function



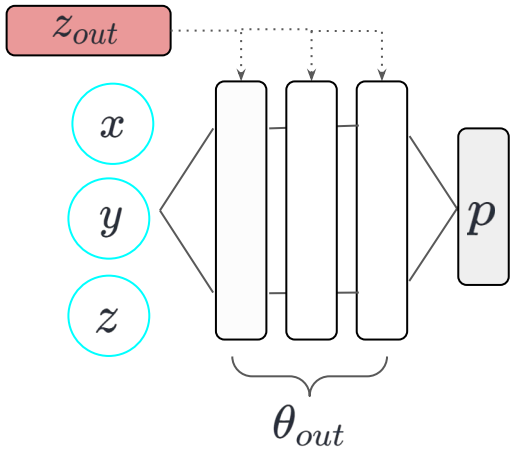


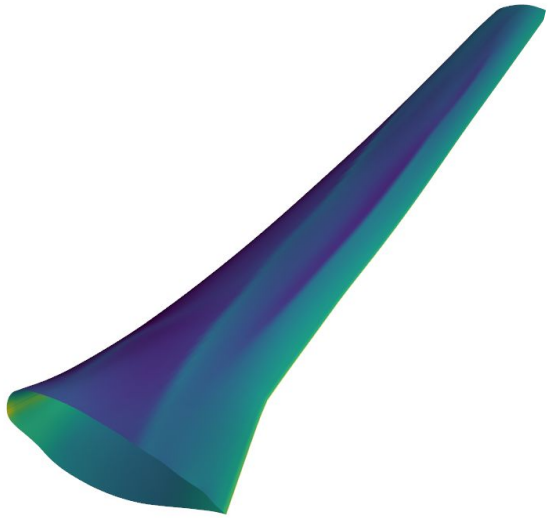
Mapping the latent spaces



Additional Parameters (Mach, Angle of Attack…)

Neural Field Encoder for the Surface Pressure





scientific reports

Explore content · About the journal · Publish with us

nature › scientific reports › articles › article

Article | Open access | Published: 26 October 2024

Neural fields for rapid aircraft aerodynamics simulations

Giovanni Catalani, Siddhant Agarwal, Xavier Bertrand, Frédéric Tost, Michael Bauerheim & Joseph Morlier

Scientific Reports **14**, Article number: 25496 (2024) | Cite this article

Catalani, Giovanni, et al. "Neural fields for rapid aircraft aerodynamics simulations." *Scientific Reports* 14.1 (2024): 25496.

ML4CFD Neurips Challenge

https://ml-for-physical-simulation-challenge.irt-systemx.fr/

Objectives:

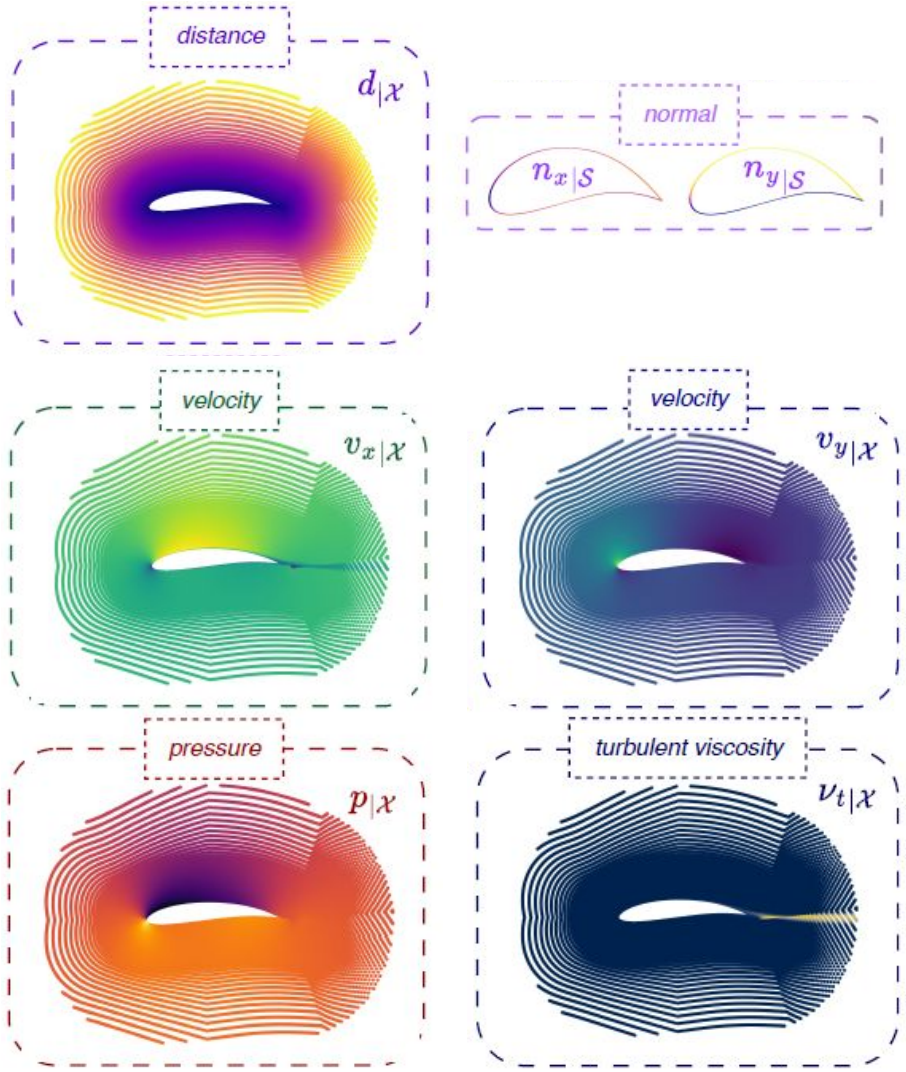
- Open international competition aimed at developing Data Driven physical simulators of Computational Fluid Dynamics.
- Task: **Predict surfacic & volumic fields** around unseen airfoils at test Reynolds numbers, Mach and Angles of Attack.
- Really small training dataset with only 100 CFD computations.

Evaluation metrics:

- Accuracy on prediction on and off the airfoil surface.
- Physical compliance: accuracy in prediction of lift and drag.
- Speed: acceleration compared to CFD solver.
- Out of Distribution performance: tested on ood geometries and ood flow regimes.

Results:

- Our approach based on Neural Fields positioned **3rd** among more than 200 teams.
- **Times 5000 speedup** at inference compared to the high fidelity simulator.



AIRBUS ISAE SUPAERO NEURAL INFORMATION PROCESSING SYSTEMS

References

Park, Jeong Joon, et al. "Deepsdf: Learning continuous signed distance functions for shape representation." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

Serrano, Louis, et al. "Operator Learning with Neural Fields: Tackling PDEs on General Geometries." *Advances in Neural Information Processing Systems* 36 (2024).

Catalani, G., Agarwal, S., Bertrand, X., Tost, F., Bauerheim, M., & Morlier, J. (2024). Aero-Nef: Neural Fields for Rapid Aircraft Aerodynamics Simulations. *arXiv preprint arXiv:2407.19916*.
Wang, Hanchen, et al. "Scientific discovery in the age of artificial intelligence." *Nature* 620.7972 (2023): 47-60.

Catalani, G., Costero, D., Bauerheim, M., Zampieri, L., Chapin, V., Gourdain, N., & Baqué, P. (2023). A comparative study of learning techniques for the compressible aerodynamics over a transonic RAE2822 airfoil. *Computers & Fluids*, *251*, 105759.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, *34*(4), 18-42.