



# Operator learning of RANS equations: a Graph Neural Network closure model

Michele Quattromini, Michele Alessandro Bucci, Stefania Cherubini, Onofrio Semeraro

## ► To cite this version:

Michele Quattromini, Michele Alessandro Bucci, Stefania Cherubini, Onofrio Semeraro. Operator learning of RANS equations: a Graph Neural Network closure model. The mathematical and statistical foundation of future data-driven engineering – Computational Challenges and Emerging Tools, Apr 2023, Cambridge, United Kingdom. hal-04407775

**HAL Id: hal-04407775**

**<https://hal.science/hal-04407775v1>**

Submitted on 21 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Operator learning of RANS equations: a Graph Neural Network closure model

Michele Quattromini<sup>1,3</sup>  
michele.quattromini@poliba.it

Michele Alessandro Bucci<sup>2</sup>  
Stefania Cherubini<sup>1</sup>  
Onofrio Semeraro<sup>3</sup>

LISN  
LABORATOIRE INTERDISCIPLINAIRE  
DES SCIENCES DU NUMÉRIQUE



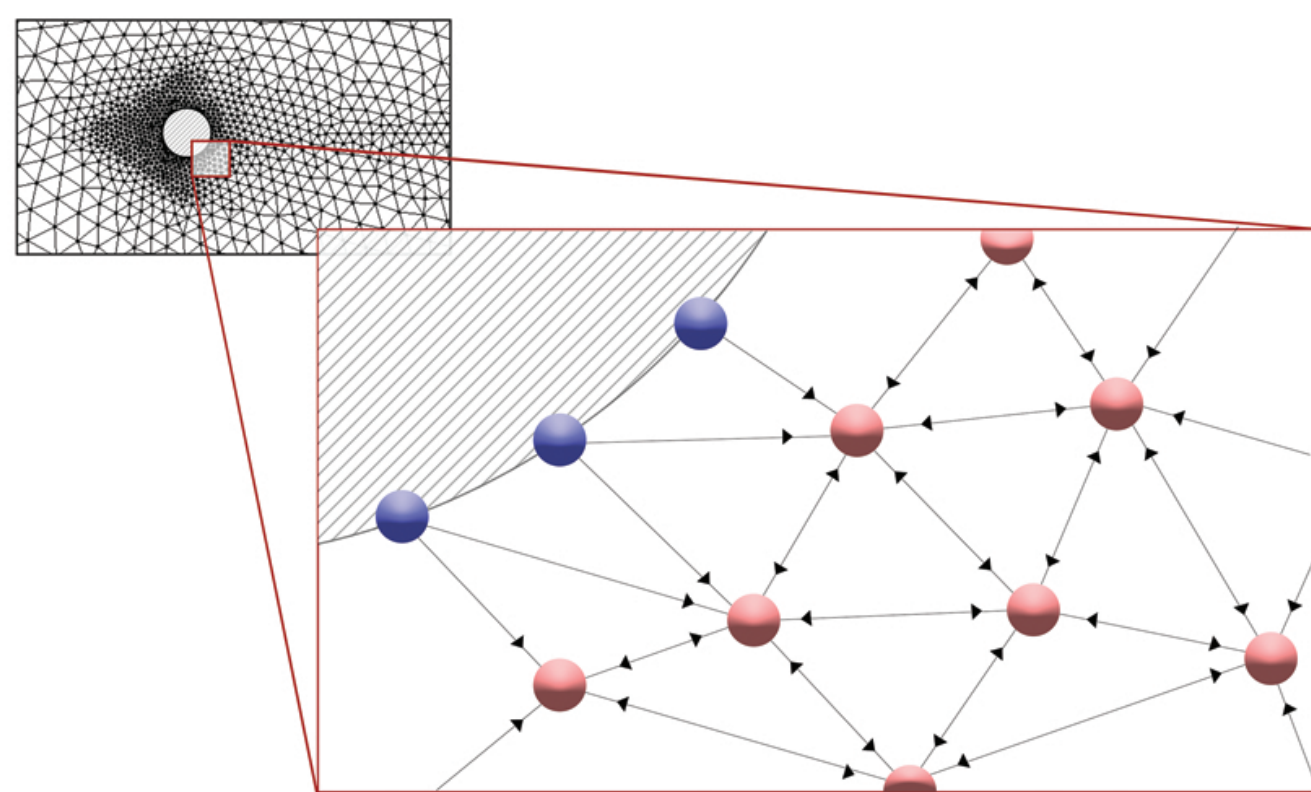
université  
PARIS-SACLAY

## Introduction

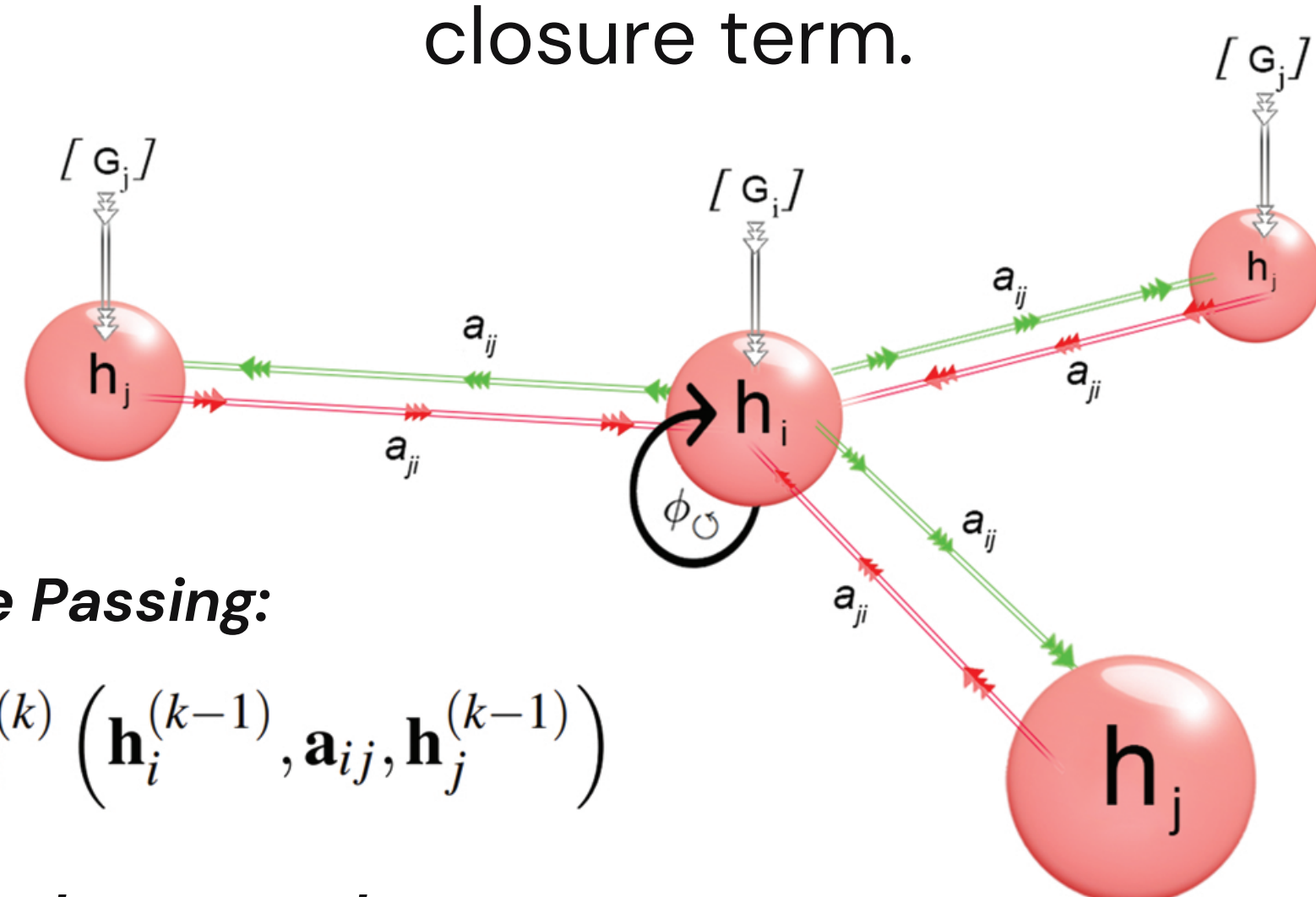
Closure modeling for Reynolds-averaged Navier-Stokes (RANS) equations is an active area of research. Finding an effective closure term may require significant modeling effort; sometimes a suitable closure model may not even exist. Here, we combine the use of **Graph Neural Networks (GNN)** [2] and finite-elements (FEM) solvers, in order to approximate a general, mesh-free **closure model** for RANS equations by solely leveraging the available data.

## Methodology

We exploit geometrical information from the mesh as a relevant part of the process of operator learning, using GNN, in order to reproduce the convective/diffusive dynamics of the underlying system. Points and connections of the mesh are one-to-one coupled with nodes and edges of the GNN architecture, creating a direct interface between FEM and GNN algorithm.



Then, the **message passing** technique allows to collect the information stored in each node of the net and reconstruct, via a node-wise **encoding-decoding** process, a meaningful closure term.



**Message Passing:**

$$\phi_{i,j}^{(k)} = \zeta_i^{(k)} \left( \mathbf{h}_i^{(k-1)}, \mathbf{a}_{ij}, \mathbf{h}_j^{(k-1)} \right)$$

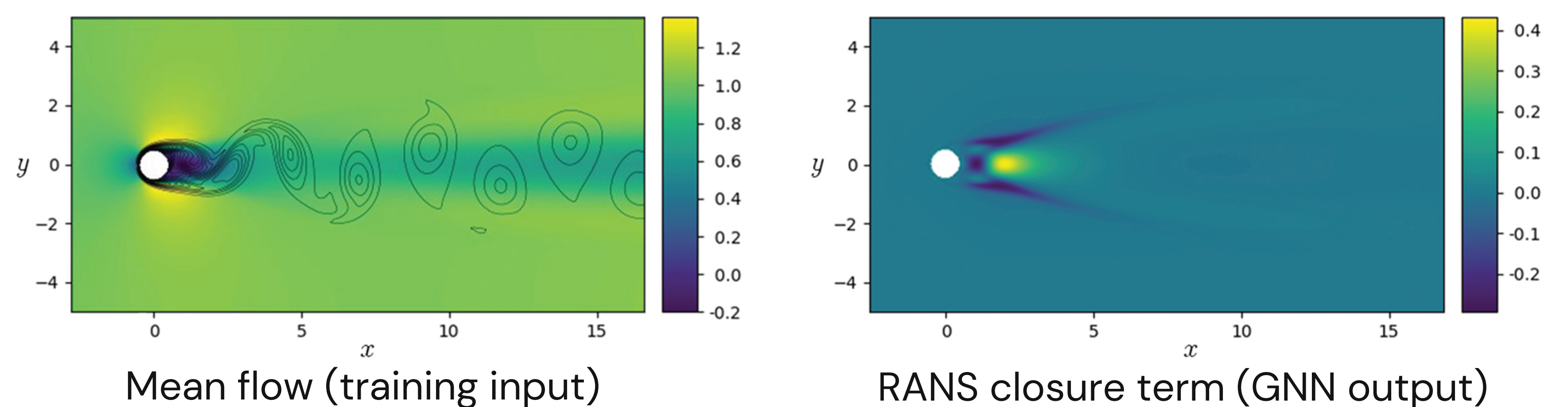
**Embedded state update:**

$$\mathbf{h}_i^{(k)} = \mathbf{h}_i^{(k-1)} + \alpha \Psi^{(k)} \left( \mathbf{h}_i^{(k-1)}, \{\bar{\mathbf{u}}, Re\}, \phi_{\rightarrow,i}^{(k)}, \phi_{\leftarrow,i}^{(k)}, \phi_{\odot,i}^{(k)} \right)$$

In the previous,  $\mathbf{h}$  is the embedded state, defined on each node;  $k$  indicates the progression of updates over  $\mathbf{h}$ ;  $\mathbf{a}$  stands for the directed connection;  $\phi$  indicates the messages shared across the net,  $\Psi$  and  $\zeta$  are two functions approximated by Multi-Layer Perceptrons (MLP), and  $\alpha$  is a hyper-parameter of the net.  $G$  indicates the inputs: the velocity  $\bar{\mathbf{u}}$ , and the Reynolds numbers ( $Re$ ).

## Dataset & training

**Training dataset:** Centered cylinder at different Reynolds numbers  
[ $Re=50$  to  $Re=150$ ,  $\Delta Re=10$ ] [ $Re=110$  is excluded for validation]



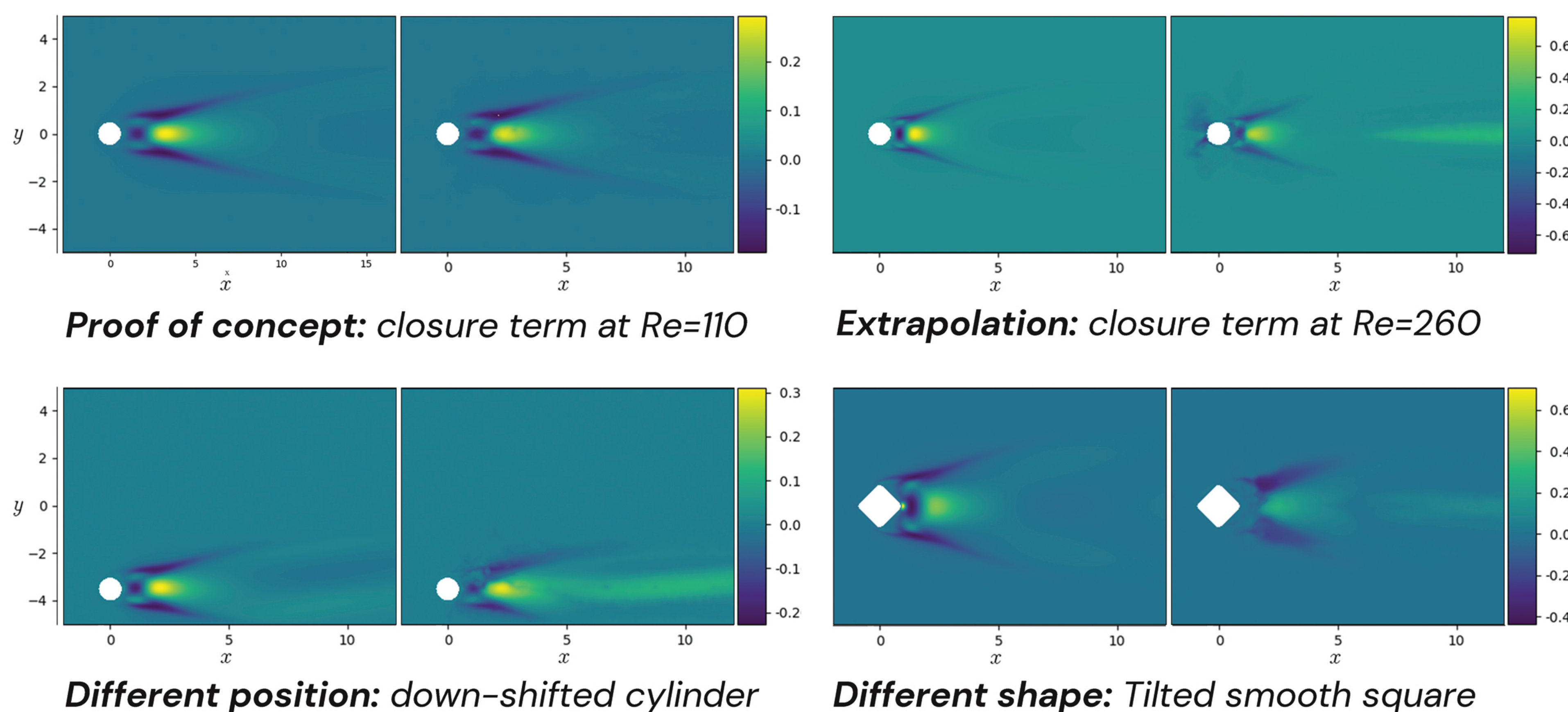
The training process has been performed on 3 differently refined meshes, depending on the Reynolds numbers, to ensure the mesh independence of the NN model.

The loss function to be minimized during the training process is as follow:

$$\varepsilon = \sum_{k=1}^{\bar{k}} \gamma^{\bar{k}-k} \left[ \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \right]$$

in which  $N$  is the number of nodes,  $\mathbf{x}$  is a vector containing the GNN's prediction for each node,  $\mathbf{y}$  is a vector of the same dimensions representing the ground truth,  $\bar{k}$  is the number of GNN layers and  $\gamma$  a process parameter

## Results



## Conclusions

In this work we show the use of GNN in combination with FEM simulations for RANS modeling. Despite the relatively small amount of data required for the training (10 snapshots), this architecture can show remarkable results in prediction terms. Future and ongoing work focuses on enforcing physical constraints deriving from the mathematical RANS equations to provide physical information to the GNN structure [1].

## References

- [1] Donon, Balthazar, et al. "Deep statistical solvers." Advances in Neural Information Processing Systems 33 (2020): 7910–7921.
- [2] Zhou, Jie, et al. "Graph neural networks: A review of methods and applications." AI open 1 (2020): 57–81.

