

## Graph Neural Network models as compliant RANS closure

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

<sup>1</sup> *Department of Mechanics, Mathematics and Management, Politecnico di Bari, Bari, Italy*

<sup>2</sup> *Dataflot–Meca Dept, LISN, CNRS, Université Paris-Saclay, Orsay, France*

<sup>3</sup> *TAU–Team, INRIA Saclay, LISN, Université Paris-Saclay, Orsay, France*

Reynolds-Averaged Navier-Stokes (RANS) equations allow to compute averaged quantities of turbulent flows at affordable costs compared to exact solutions from direct numerical simulations (DNS). However, they require modeling efforts for their closure, based on the Reynolds stress tensor. Nowadays, thanks to the spread of machine learning (ML) techniques, boosted by the development of deep learning, and the great availability of computational resources and datasets [5], it is possible to obtain these closure models directly from neural networks (NN). Within this new field of research, there are already numerous examples [3]; an early one is provided by [6], where the closure terms are predicted combining various techniques for the classification and model evaluation based on NN. However, a large variety of strategies and architectures is available, ranging from regression methods [8] to decision trees [3]. Here, we focus on the application of graph neural networks (GNN) and deploy them for the modeling closure of RANS equations. Within this context, GNNs have a strong potential due to their peculiarities. Beside being differentiable, compared to techniques such as convolutional NN, this architecture is characterized by complex multi-connected nets of nodes that can be provided as input and easily adapted to unstructured mesh for computational fluid mechanics and data-driven modeling (see Fig. 1a). It has been recently shown that this feature allows to perform operator learning for the solution of partial differential equations (PDE), while retaining a great flexibility with respect to the domain of computation [2].

Based on this rationale, we developed and tested a framework interfacing a Finite Elements (FEM) solver written in `Fenics` [1] and `Pytorch geometric`; as demonstrator, we consider DNS of the dynamics past a cylinder flow, at different Reynolds numbers  $40 < Re < 150$  resolved on different grids from coarser to finer ones. The final dataset is composed by the meanflows and the Reynolds stress tensors computed at different  $Re$ , with a stride of  $\Delta Re = 10$ . The training is performed by supervised learning and aims at predicting the Reynolds stress based on the meanflow as input, thus reproducing the action of the RANS operator. Preliminary results show that within a relatively small number of epochs, *i.e.* the training cycle, it is possible to obtain GNN models that are capable of approximating with an accuracy of  $\approx 80\%$  the Reynolds stress also at  $Re$  numbers for which the GNNs are not trained (see Fig. 1b – c) inside and outside the  $Re$  range. These preliminary results provide a first assessment on the extent to which it is possible to parametrize the GNN models and perform operator learning.

However, the application can represent a true advantage with respect to standard applications only if the independence from the geometry is demonstrated. To this end, further strategies are at our disposal and are currently under scrutiny. A first aspect concerns the boundary conditions that can be treated leveraging the directionality along the edges of the GNN net, such that the propagation of the information can be designed for fulfilling Dirichlet or Neumann boundary conditions. Moreover, the physics of the system dictates the depth the neural network and its parametrization. In that sense, an incompressible case requires aggregating information from the entire domain and thus necessitates a rather deep network that allows the update of the feature vector – which in our case consists of the state of the system including velocity and pressure – through the information collected also from points rather far. To tackle this problem, the second improvement under investigation is the application of U-Net architectures, that were recently extended to GNN [4] and applied to PDE simulations [7]. The U-net strategy allows to

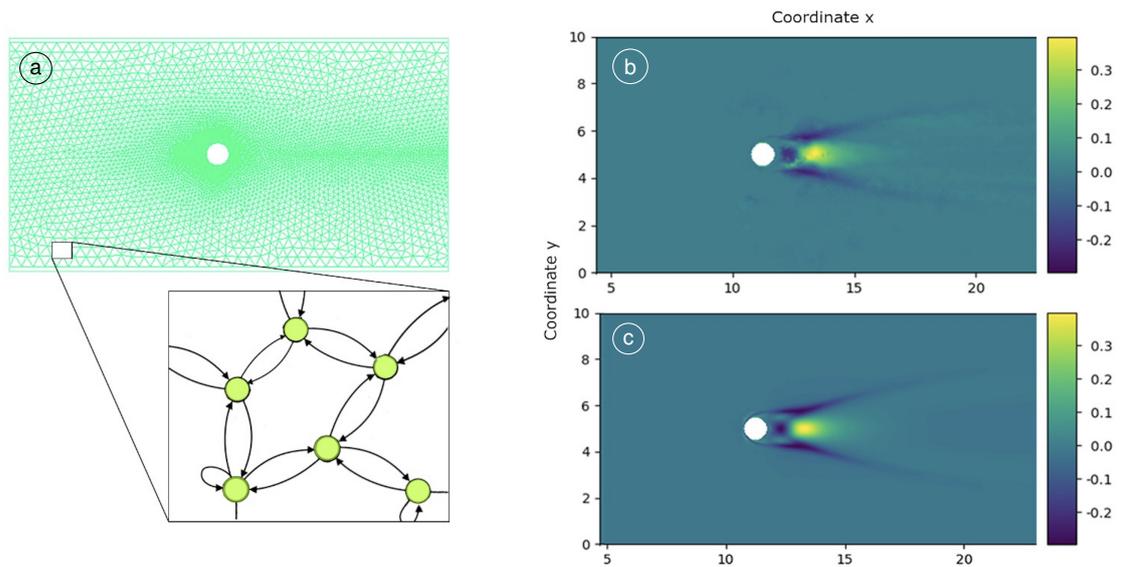


Figure 1: Example of a FEM mesh for simulations of the flow past a cylinder serving as a multi-connected net for a neural network architecture (a); Reynolds stresses obtained as an output of a graph neural network at  $Re = 140$ , (b), compared to the reference computed by DNS in (c).

downsample a fine grid and to collect the information from distant nodes within a limited number of connections, mimicking multigrids methods.

In the final contribution, we will discuss our studies with a special focus on the generalization properties of the GNN models, in presence of different geometries and boundary conditions.

## References

- [1] Alnæs, M., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M.E., Wells, G.N. *The FEniCS project version 1.5*
- [2] Donon, B., Liu, Z., Liu, W., Guyon, I., Marot, A. Schoenauer, M. (2020) *Deep Statistical Solvers* NeurIPS
- [3] Duraisamy, K., and Iaccarino, G., Xiao, H. (2019) *Turbulence modeling in the age of data* Ann. Rev. Fluid Mech., **51**357–377
- [4] Gao, H., Ji, S.(2019) *Graph U-nets* International conference on machine learning, 2083–2092
- [5] Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y. (2016) *Deep learning* MIT press
- [6] Ling, J., Templeton, J. (2015) *Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty* Phys. of Fluids, **27**(8)–085103
- [7] Lino, M., Cantwell, C., Bharath, A.A., Fotiadis, S. (2021) *Simulating Continuum Mechanics with Multi-Scale Graph Neural Networks* arXiv:2106.04900
- [8] Schmelzer, M., Dwight, R.P., Cinnella, P. (2019) *Machine learning of algebraic stress models using deterministic symbolic regression*, arXiv:1905.07510