22nd International Conference on
Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes 10-14 June 2024,
Pärnu, Estonia

# NEURAL SURROGATES FOR ATMOSPHERIC DISPERSION IN BUILT-UP AREAS

*Armand de Villeroché* [1], *Vincent Le Guen* [1], *Rem-Sophia Mouradi* [1], *Patrick Massin* [1],
*Marc Bocquet* [1], *Alban Farchi* [1], *Sibo Cheng* [1], *Patrick Armand* [2]
[1] CEREA, École des Ponts and EDF R&D, Île-de-France, France
[2] CEA, DAM, DIF, F-91297 Arpajon, France

**Abstract:** Studies of atmospheric dispersion of pollutants on a local scale are increasingly performed with Computational Fluid Dynamics (CFD) simulations. However, CFD computations can be numerically expensive, and are often only performed on a limited number of cases. Machine Learning (ML) approaches offer a promising potential to build meta-models, i.e. approximations to the CFD solver, for faster results, allowing to simulate cases not present in the initial CFD-generated database. Here, a Multi-Layer Perceptron (MLP) is trained on CFD simulation results with varying wind directions. The MLP model is able to predict accurately the wakes near the building, but has trouble predicting the correct wakes further downstream. Furthermore, knowledge of the continuity equation is embedded into the network via an additional loss term. This allows to slightly improve the surrogate model's accuracy.

***Key words:*** *Computational fluid dynamics, deep learning, physics-informed neural networks, atmospheric dispersion*

## INTRODUCTION

Industrial sites and urban areas present a risk of accidental release of pollutants. Such particles or gases can then be carried away by the wind and spread over large areas, with possible environmental and sanitary impacts. When a hazardous release occurs, it is crucial to precisely and rapidly know where the pollutant plume is likely to disperse in order to discriminate the impacted from the safe areas. Such information can help local authorities take preventive and reactive actions, for example when an emergency evacuation is needed.

At local scale, dispersion modeling is a complex problem that is highly dependent on topography, buildings geometry and meteorological conditions. In practice, this problem can be solved using Computational Fluid Dynamics (CFD) simulations of the atmospheric flow and of the dispersion of the pollutant. In case the pollutant has no impact on the fluid density, these two systems can be resolved independently, which is here the case. However, such simulations are computationally intensive, and the resulting data is costly to store.

The objective of our investigation is to establish a meta-model of the problem. Such a meta-model could be learned from existing or new simulations, and could possibly allow fast interpolation or extrapolation to new, non-simulated cases. Since, in resolving the dispersion problem, the most costly step is the resolution of the 3D wind flow around the buildings, the focus of this study is to learn a meta-model able to correctly predict the fluid flow and not the pollutant concentration.

## METHODOLOGY

### Case study

To assess the performance of the proposed methods, we study a toy case consisting of a few buildings and 3 hills, representing a simplified power production plant. The domain is cylindrical, with a radius of

1 km and a height of $600$ m. The buildings and hills are clustered around the domain center. The mesh resolution close to the buildings is 1 m, and the resolution far from the buildings is 5 meters. The mesh is displayed in Figure 1.

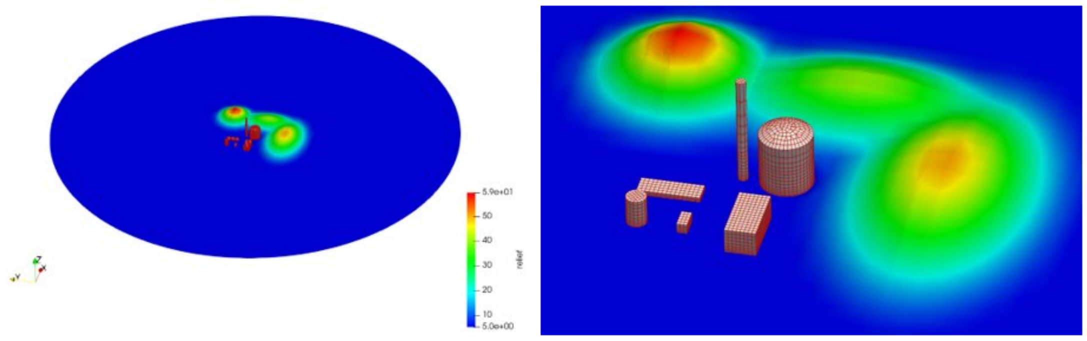

**Figure 1.** Left: domain of the studied toy case. Right: zoom on the buildings and hills.

As a first investigation, a total of 72 simulations were run with code_saturne 8.0 (Archambeau et al., 2004), focusing only on the air flow with no pollutants, in unstable stratification, and with varying wind directions. The cells were initialized with the values of Monin-Obukhov profiles (Foken, 2006) with the Högström universal functions suited for unstable cases (Högström, 1988). Parameters for the profiles are as follows: Monin-Obukhov length $L_{\mathrm{mo}} = -50$ m, ground roughness $z_0 = 0.13$ m, reference wind velocity $u_{\mathrm{ref}} = 6$ m/s at height $z_{\mathrm{ref}} = 60$ m, and ground temperature $T_0 = 20$ °C. The wind direction $\vartheta$ varies between each simulation, with one run every $5°$. Those profiles are prescribed as boundary conditions in the upstream half of the domain. Then, the Navier-Stokes equations with the Boussinesq approximation and the energy conservation equation are solved for 25000 time-steps of $1$ s each. This allows to reach a near stationary state. The last $1500$ time-steps are averaged to remove the remaining time variations (such as Von Karman vortex streets). The final simulation outputs are time-averaged 3D fields of velocity $\mathbf{v}$, potential temperature $\theta$, pressure $p$, turbulent kinetic energy $k$, and turbulent kinetic energy dissipation rate $\epsilon$ for every simulated case.

The dataset generated is then split into training, validation and test sets. The training dataset consists of regularly spaced wind direction occurrences, every $10°$, starting from $0°$ (North wind). This represents a velocity field in the domain for 36 different wind directions, for a total of 48 million spatio-directional points. The validation and test datasets also consist of regularly spaced wind direction occurrences, every $10°$, starting from $5°$, i.e. intermediate values from the train dataset. The validation dataset is constituted with case directions $\{5°, 45°, 95°, 135°, 185°, 225°, 275°\}$, that is, ordinal directions and points close to cardinal directions, as the cardinal points themselves are part of the training set. Finally, the remainder is used for the post-training test set. Since the CFD resolution is higher than necessary for training the meta-model far from the buildings, only $10\%$ of the spatial points are kept for training, with a focus around the building and close to the ground.

## End-to-end metamodel structure

In order to compare different models easily, and to facilitate the calculation of physical quantities and losses, normalisation and denormalisation is implemented directly around the model, as shown in Figure 2. Additionally, models predict only a perturbation relative to the computed Monin-Obukhov (MO) meteorological profiles, and profiles values are computed separately and added to the model output. This means that losses can be computed on physical quantities, but that the actual model trains on output normalised quantities. Gradients can then be automatically computed and backpropagated, including gradients of MO profiles relative to the inputs.

## Evaluation metrics for metamodels

In order to evaluate the accuracy of a given deep learning model, several statistical metrics may be used. Here, we use the Geometric Mean bias (MG), the Geometric Variance (VG), the correlation coefficient denoted R, and the fraction of prediction within a factor 2 of the observation denoted FAC2 (Chang and

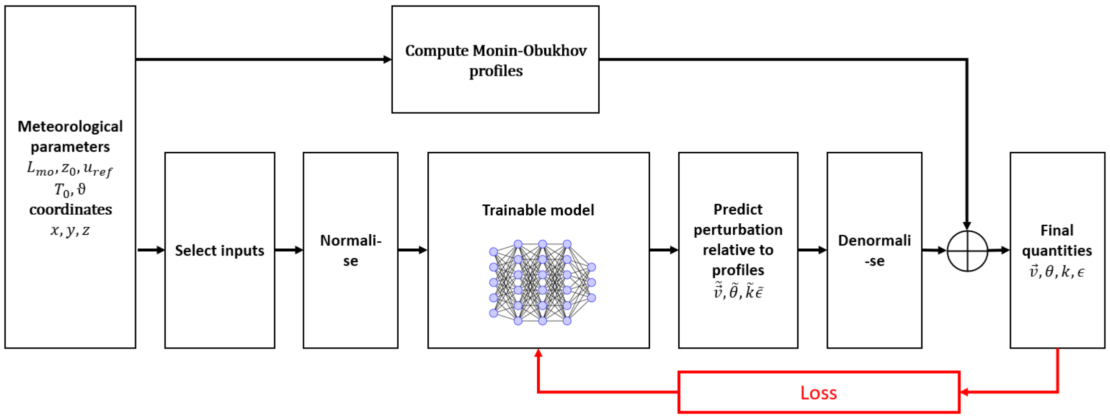
Meteorological parameters
$L_{mo}, z_0, u_{ref}$
$T_0, \vartheta$
coordinates
$x, y, z$
Compute Monin-Obukhov profiles
Select inputs
Normali-se
Trainable model
Predict perturbation relative to profiles
$\tilde{\vec{v}}, \tilde{\theta}, \tilde{k}\tilde{\epsilon}$
Denormali -se
Final quantities
$\vec{v}, \theta, k, \epsilon$
Loss

**Figure 2.** General schematic of the studied models.

**Table 1.** Metrics used for evaluating deep learning model's performances.

| Name | formula | meaning |
|---|---|---|
| MG | $\exp(\overline{\ln V_o} - \overline{\ln V_p})$ | bias indicator |
| VG | $\exp(\overline{(\ln V_o - \ln V_p)^2})$ | error estimator |
| R | $\frac{\overline{(V_o - \overline{V_o})(V_p - \overline{V_p})}}{\sigma_{V_o}\sigma_{V_p}}$ | correlation estimator |
| FAC2 | Fraction of data that satisfy $0.5 \leq \frac{V_p}{V_o} \leq 2$ | accuracy estimator |

Hanna, 2004). All those metrics have an ideal target value of 1. With $V_p$ the model prediction and $V_o$ the ground truth data, and with $\overline{V}$ and $\sigma_V$ representing the average and standard deviation over the dataset respectively, we have:

Metrics values are computed on denormalised perturbations, relative to the Monin-Obukhov profiles. This includes scalar and vector fields, such as the velocity, with potentially negative values. Since the given metrics only make sense on positive values, they are computed on the absolute values. For the case of the velocity, separate metrics are computed on the component parallel to the wind direction, the component perpendicular to the wind direction, and the vertical component. For each category, a final metric value can be computed as the mean of computed metrics on all predicted fields.

## SURROGATE MODELS

### Data-driven multi-layer perceptron

The first explored model is purely data driven. It is a fully connected multi-layer perceptron moderately deep and wide, consisting of 20 layers, each containing 50 neurons, with tanh activation function, implemented with Tensorflow. The model takes as input the spatial coordinates of a point and the physical parameters (cosine and sine of the wind direction angle $\vartheta$), and outputs the velocity $\mathbf{v}$, potential temperature $\theta$, turbulent kinetic energy $k$ and turbulent kinetic energy dissipation rate $\epsilon$. The data loss function is defined in Equation 1, with $||\cdot||$ the Euclidean norm:

$$\mathscr{L}_{\text{data}} = \sum_{\text{data points}} ||\mathbf{v}_{\text{pred}} - \mathbf{v}_{\text{saturne}}||^2 + |\theta_{\text{pred}} - \theta_{\text{saturne}}|^2 + |k_{\text{pred}} - k_{\text{saturne}}|^2 + |\epsilon_{\text{pred}} - \epsilon_{\text{saturne}}|^2. \tag{1}$$

The model is trained with the ADAM optimizer with amsgrad, with a learning rate of $\alpha = 10^{-3}$ and a batch size of 4096. The training is done until the loss function does not improve during more than 12 consecutive epochs, and the weights of the epoch with the lowest validation loss are saved at the end of the training. An example of prediction of the velocity field for the test direction $\vartheta = 135°$ is shown in Figure 3.

The network predicts very well the wakes close to the buildings, including the different recirculation

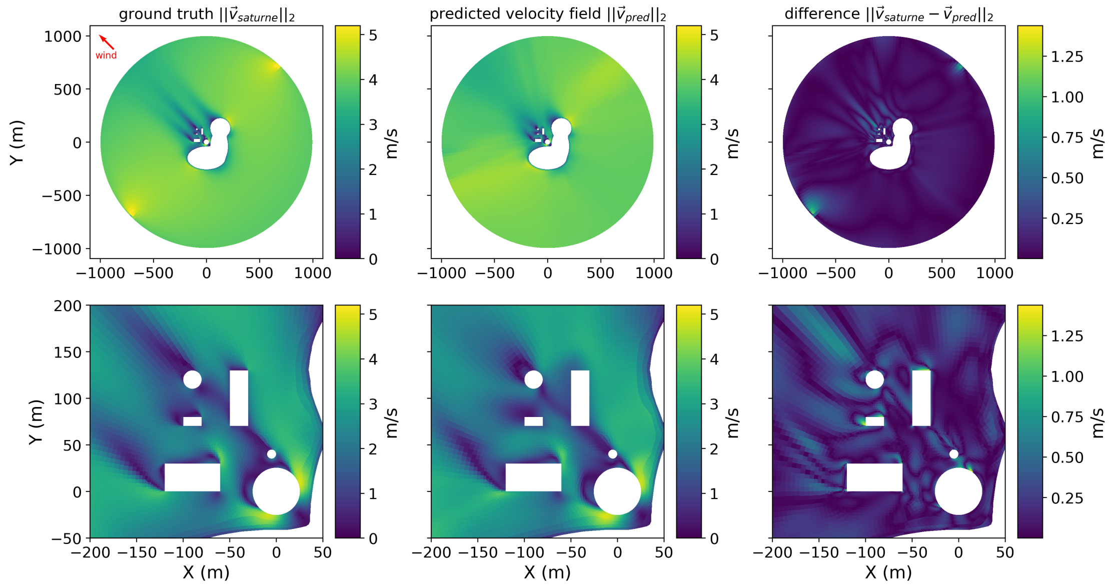

**Figure 3.** Left: Ground truth CFD data for the test case $\vartheta = 135°$. Middle: MLP prediction for this case. Right: difference between the two fields.

zones, except for the lowest building. The highest errors close to the buildings are located around sharp gradients, such as around corners of rectangular buildings. Overall, adding boundary conditions around the buildings as an additional loss term may help solve these issues. This may also be due to the spectral bias in neural networks (Rahaman et al., 2019).

The network however struggles to predict wakes farther away from the buildings, and tends to overspread them. This may in part be due to the double-penalty effect of the Mean Square Error (Farchi et al., 2016). A possible solution would be to add the momentum conservation and the mass conservation as additional loss terms, as they are key to preserve the wakes.

## Physics-informed loss

Under the Boussinesq approximation, with $\rho_a$ the adiabatic density, the continuity equation becomes Equation 2.

$$\nabla(\rho_a \mathbf{v}) = 0 \tag{2}$$

In order to improve the model results, the continuity equation is incorporated in the model. The model corresponds to a PINN (Raissi et al., 2019) with both data and physics-based terms in the loss. The collocation points are the same as the data points. An additional parameter $\lambda_{\text{continuity}}$ is introduced to balance the two loss terms. The resulting loss function is defined in Equation 3.

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{data}} + \lambda_{\text{continuity}} \sum_{\text{collocation points}} \nabla(\rho_a \mathbf{v}) \tag{3}$$

Metrics on trained models with different values of $\lambda_{\text{continuity}}$ are shown in Figure 4. We focus on the metrics of the velocity field components. Metrics improve slightly for intermediate values of $\lambda_{\text{continuity}}$. This shows that adding the continuity equation does improve the accuracy of the model. However, for $\lambda_{\text{continuity}} > 1$, the metrics worsen. This shows that putting too much emphasis on the continuity equation hampers the training of the network, leading to an overall less accurate surrogate model. However, improvements on the metrics are always small, meaning that the added value of the continuity equation is relatively limited. This is in part due to the MLP network having a low base error on the continuity equation.

## CONCLUSION

We used an MLP to interpolate between known directions for the velocity field. The MLP shows very good performance for predicting the wind behavior close to buildings but has trouble predicting the correct

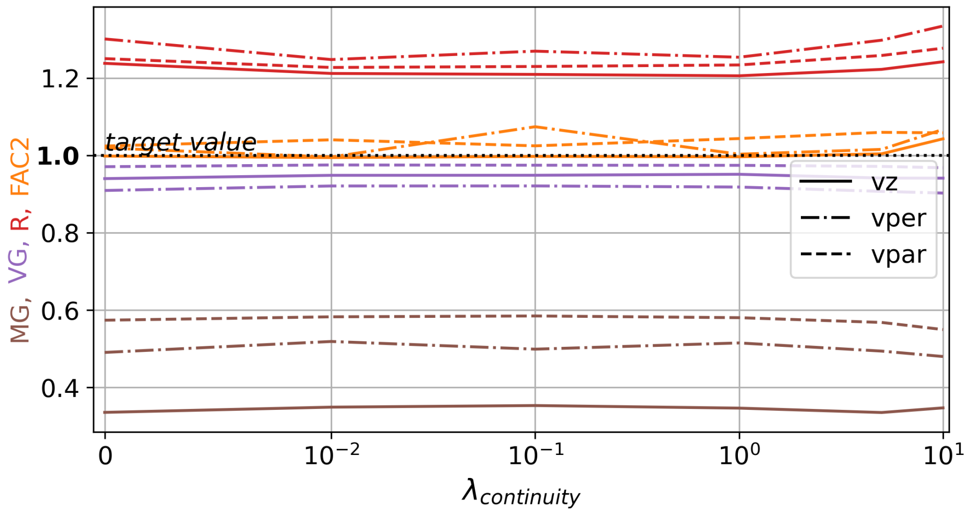

**Figure 4.** Metrics on trained models for different values of $\lambda_{\text{continuity}}$, computed on the vertical component of the velocity field $V_z$ (continuous), the component perpendicular to the wind direction $V_{\text{per}}$ (dash-dotted), and the component parallel to the wind direction $V_{\text{par}}$ (dashed).

wakes further downstream.

We then use the PINN formulation to embed knowledge of the continuity equation into the network. This allows to slightly improve the accuracy of the network. However, since the data-driven network has a relatively low base error on the continuity equation, the added value is relatively small.

An interesting perspective would be to add equations which are not as well respected, such as the momentum equation, and evaluate its impact on the network's accuracy and in particular the wakes distant from the built-up areas.

## REFERENCES

F. Archambeau, N. Méchitoua, and M. Sakiz. Code saturne: A finite volume code for the computation of turbulent incompressible flows-industrial applications. *International Journal on Finite Volumes*, 1 (1), 2004.

J. C. Chang and S. R. Hanna. Air quality model performance evaluation. *Meteorology and Atmospheric Physics*, 87(1-3):167–196, 2004.

A. Farchi, M. Bocquet, Y. Roustan, A. Mathieu, and A. Quérel. Using the wasserstein distance to compare fields of pollutants: application to the radionuclide atmospheric dispersion of the fukushima-daiichi accident. *Tellus B: Chemical and Physical Meteorology*, 68(1):31682, 2016.

T. Foken. 50 years of the monin–obukhov similarity theory. *Boundary-Layer Meteorology*, 119:431–447, 2006.

U. Högström. Non-dimensional wind and temperature profiles in the atmospheric surface layer: A re-evaluation. *Topics in Micrometeorology. A Festschrift for Arch Dyer*, pages 55–78, 1988.

N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. 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. doi: https://doi.org/10.1016/j.jcp.2018.10.045.