Neural network-based emulation of ISM models

P. Palud, L. Einig, F. Le Petit, E. Bron, P. Chainais, J. Chanussot
and the ORION-B consortium

Interstellar medium and numerical simulations

Numerical simulations are widely used in order to model the ISM and compare theory with observations.

Interstellar medium and numerical simulations

Numerical simulations are widely used in order to model the ISM and compare theory with observations.

Computation time is often prohibitive for inference procedures.

Interstellar medium and numerical simulations

Numerical simulations are widely used in order to model the ISM and compare theory with observations.

Computation time is often prohibitive for inference procedures.

Usual solutions

Interstellar medium and numerical simulations

Numerical simulations are widely used in order to model the ISM and compare theory with observations.

Computation time is often prohibitive for inference procedures.

Usual solutions

- Interpolation methods:
  1. Nearest point in grid (Sheffer+2011; Sheffer+2013; Joblin+2018)
  2. SciPy interpolation (Wu+2018; Ramambason+2022)

Interstellar medium and numerical simulations

Numerical simulations are widely used in order to model the ISM and compare theory with observations.

Computation time is often prohibitive for inference procedures.

Usual solutions

- Interpolation methods:
  1. Nearest point in grid (Sheffer+2011; Sheffer+2013; Joblin+2018)
  2. SciPy interpolation (Wu+2018; Ramambason+2022)
- Regression-based approximations:
  1. *k*-nearest neighbors (Smirnov-Pinchukov+2022)
  2. Random forests (Bron+2021)
  3. Neural networks (de Mijolla+2019; Holdship+2021; Grassi+2022)

Interstellar medium and numerical simulations

Numerical simulations are widely used in order to model the ISM and compare theory with observations.

Computation time is often prohibitive for inference procedures.

Usual solutions

- Interpolation methods:
  1. Nearest point in grid (Sheffer+2011; Sheffer+2013; Joblin+2018)
  2. SciPy interpolation (Wu+2018; Ramambason+2022)
- Regression-based approximations:
  1. k-nearest neighbors (Smirnov-Pinchukov+2022)
  2. Random forests (Bron+2021)
  3. Neural networks (de Mijolla+2019; Holdship+2021; Grassi+2022)

     → Less complex, so faster and allow more training data

A Meudon PDR approximation as a template

The Meudon PDR code (Le Petit+2006)

- Emulates **photo-dissociation regions** (PDRs) at equilibrium.
- This version: **4** inputs $\longmapsto \sim$ **5 000** spectral lines intensities
- Execution time $\sim$ **6 hours** and may yield **anomalies**.
- Predictions **directly comparable with observations**.

A Meudon PDR approximation as a template

The Meudon PDR code (Le Petit+2006)

- Emulates **photo-dissociation regions** (PDRs) at equilibrium.
- This version: **4** inputs $\longmapsto \sim$ **5 000** spectral lines intensities
- Execution time $\sim$ **6 hours** and may yield **anomalies**.
- Predictions **directly comparable with observations**.

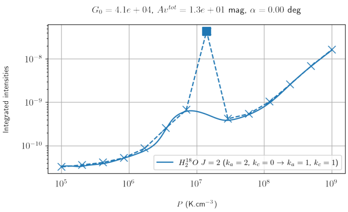
Overall: representative example of ISM numerical simulations.

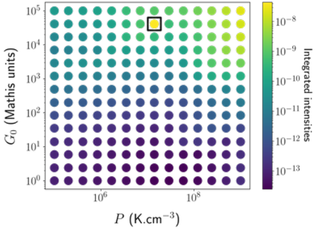
A Meudon PDR approximation as a template

The Meudon PDR code (Le Petit+2006)

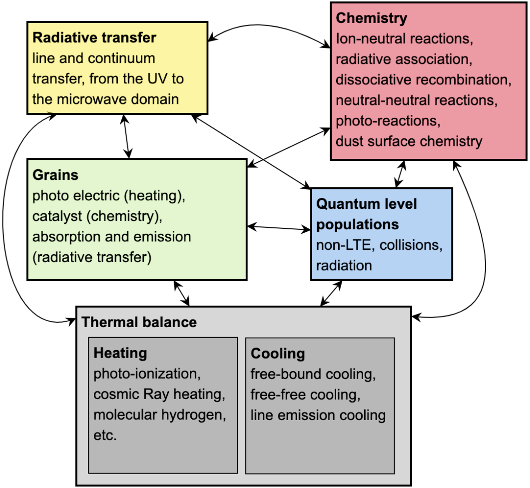
- Emulates **photo-dissociation regions** (PDRs) at equilibrium.
- This version: **4** inputs $\longmapsto \sim$ **5 000** spectral lines intensities
- Execution time $\sim$ **6 hours** and may yield **anomalies**.
- Predictions **directly comparable with observations**.

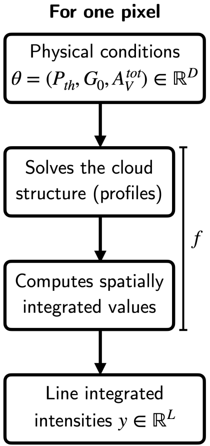
Overall: representative example of ISM numerical simulations.



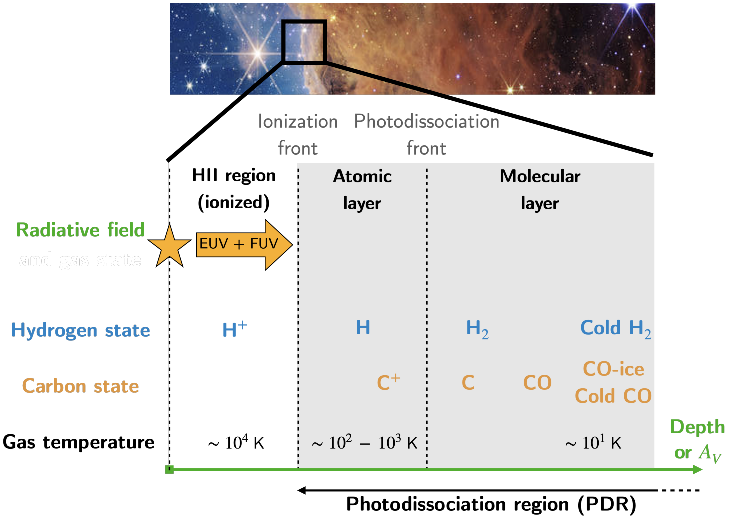


The Meudon PDR code

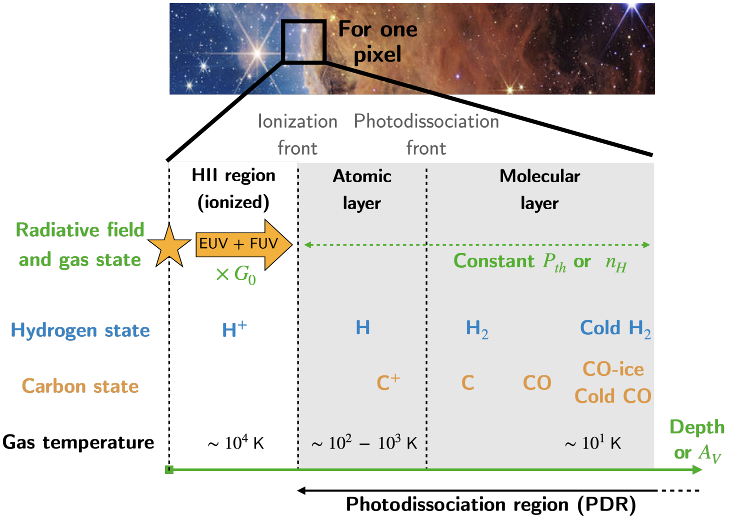




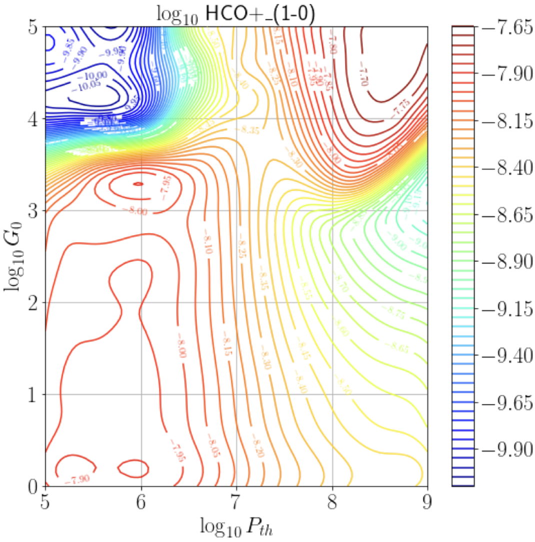
The Meudon PDR code



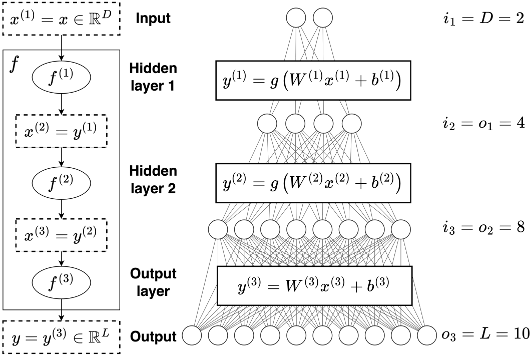
The Meudon PDR code



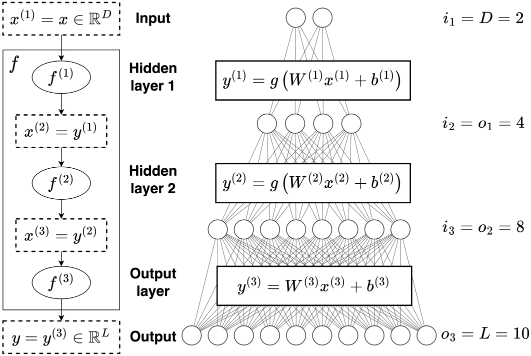
The Meudon PDR code



Evolutions of a standard multilayer perceptron



Evolutions of a standard multilayer perceptron



Proposition 1: polynomial expansion of inputs

Can help a network creating non-linearities. It has to be done **at execution** to ensure the network to be **fully differentiable**.

Ex: $P_2(x_1, x_2, x_3) = (x_1, x_2, x_3, x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2)$

Evolutions of a standard multilayer perceptron

Proposition 2: ignoring anomalies

Anomalies $\neq$ well-modeled points with sensitive behavior!

1. Training with a **robust loss** (e.g., Cauchy) to detect badly reconstructed points.
2. Use physics knowledge to determine anomalies among them.
3. New training from scratch with a **masked non-robust loss function** (e.g., MSE), **ignoring the abnormal outputs**.

Evolutions of a standard multilayer perceptron

Proposition 2: ignoring anomalies

Anomalies $\neq$ well-modeled points with sensitive behavior!

1. Training with a **robust loss** (e.g., Cauchy) to detect badly reconstructed points.
2. Use physics knowledge to determine anomalies among them.
3. New training from scratch with a **masked non-robust loss function** (e.g., MSE), **ignoring the abnormal outputs**.

Proposition 3: outputs clustering

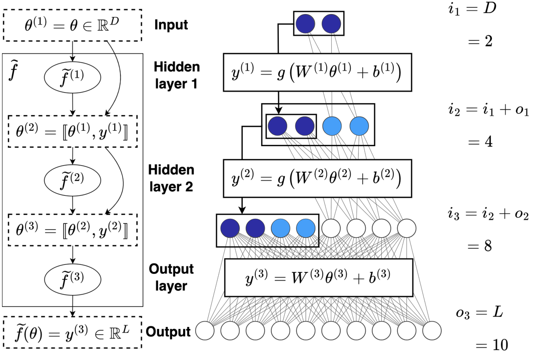
We split lines into clusters based on their **similarity**, and train **dedicated networks** for each cluster.

Method: Spectral clustering.

Evolutions of a standard multilayer perceptron

Proposition 4: reuse intermediate computations

As some outputs can be computed from other outputs, keeping track of **intermediate results** optimizes network capacities.



Dense architecture with a growing factor of 2

Results on the Meudon PDR code

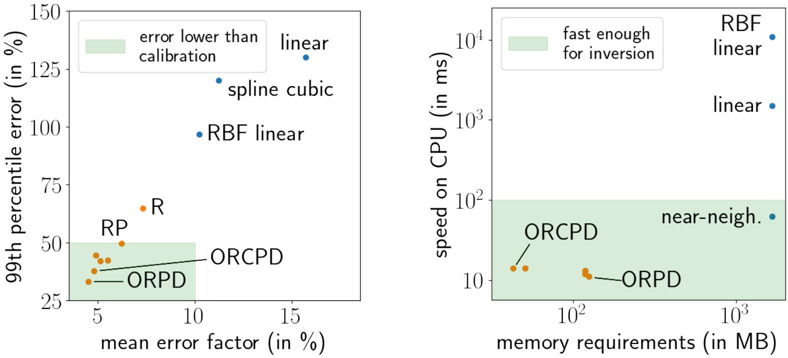
We compare the results obtained with the neural network with interpolation methods.

Metrics

- **Accuracy** – error factor, a kind of symmetrized relative error, computed on **unseen data** (testing set).
- **Memory** – size of the model (whole grid for interpolations)
- **Speed** – computed on a laptop for a batch of 1 000 entries.

To be usable in inference, emulation must satisfy some constraints on these metrics.

Results on the Meudon PDR code



Summary of results and comparison with interpolation methods.

R: regression by an ANN **O:** outliers removal **P:** polynomial expansion

C: lines clustering **D:** dense architecture

Conclusion

Take-home messages

- Deep learning is efficient to emulate complex simulations, especially with additional constraints.
- Emulators can be plugged in Bayesian inference.
- AI benefits from physical knowledge and rigorous data analysis.

- **Paper:** Palud, Einig et al., A&A, 677, A158 (2023)
- **GitHub:** `einigl/ism-model-nn-approximation`
- **PyPI:** `pip install nnbma`

References

J. Pety et al. (2017)
The anatomy of the Orion B giant molecular cloud: A local template for studies of nearby galaxies
Astronomy & Astrophysics, 599, p. A98

A. Licciardi A and J. Chanussot (2015)
Nonlinear PCA for visible and thermal hyperspectral images quality enhancement
IEEE Geoscience and Remote Sensing Letters, 12, pp. 1228-1231

A. Marchal et al. (2019)
ROHSA: Regularized Optimization for Hyper-Spectral Analysis-Application to phase separation of 21 cm data
Astronomy & Astrophysics, 626, A101