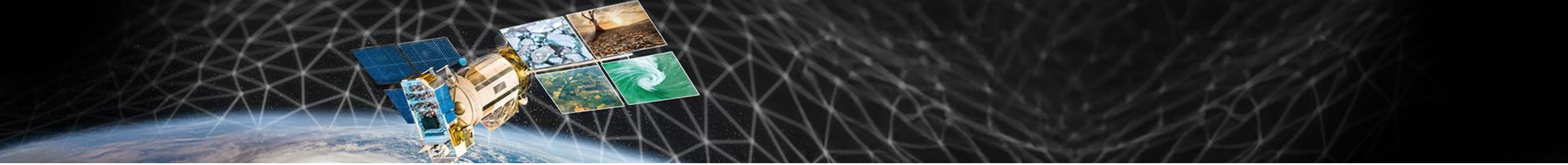


Latent Twins Framework for FORUM: Real-Time All-Sky Retrieval, Scene Recognition, and Cloud Characterization

Sgattoni C.[1], Chung M.[2], Sgheri L.[3]

[1] *Institute of BioEconomy (IBE), National Research Council (CNR), Florence, Italy*

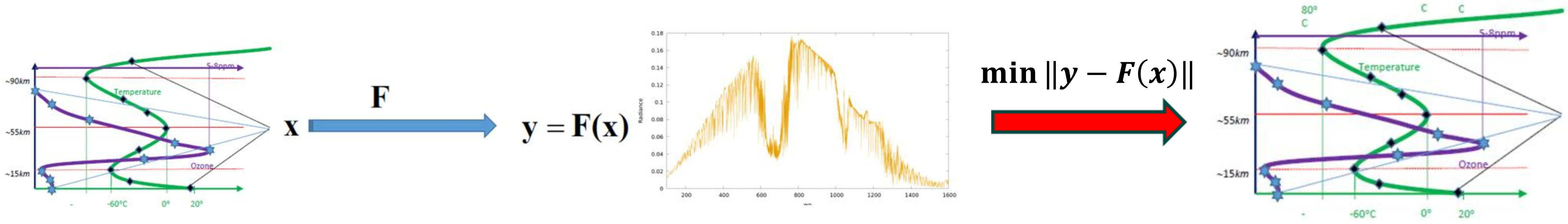
[2] *Department of Mathematics, Emory University, Atlanta, GA, USA*

[3] *Institute of Applied Mathematics (IAC), National Research Council (CNR), Florence, Italy*

Background

Direct problem: from the atmospheric status vector **x** find the simulated spectrum $\mathbf{y} = \mathbf{F}(\mathbf{x})$, with **F** known as **forward model.**

Inverse problem: from the measured spectrum **y** find the parameter vector **x** (**retrieval vector**) which minimizes $\left|\left|\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{x})\right|\right|$.



Retrieval – classical approach

- Find the atmospheric parameters $\mathbf{x}$ (**surface temperature, temperature, water vapor, ozone, surface spectral emissivity, clouds parameters**) that best reconstruct the measured spectrum $\mathbf{y}$.

! VERY ILL-CONDITIONED PROBLEM

- The problem is formulated as a Bayesian inference problem, solved using the **OPTIMAL ESTIMATION METHOD**[1]:

$$\mathbf{x}_{\mathrm{OE}} = \arg\min_{\boldsymbol{\xi}} \frac{1}{2}\left\|\mathbf{L}_{\mathbf{y}}\big(\mathbf{y} - \mathbf{F}(\xi)\big)\right\|_2^2 + \frac{1}{2}\|\mathbf{L}_{\mathrm{a}}(\boldsymbol{\xi} - \mathbf{x}_{\mathrm{a}})\|_2^2,$$

where $\mathbf{S}_{\mathbf{y}}^{-1} = \mathbf{L}_{\mathbf{y}}^{\mathrm{T}}\mathbf{L}_{\mathbf{y}}$ and $\mathbf{S}_{\mathrm{a}}^{-1} = \mathbf{L}_{\mathrm{a}}^{\mathrm{T}}\mathbf{L}_{\mathrm{a}}$ are the inverses of the covariance matrices of the measurements $\mathbf{y}$ and the a priori information $\boldsymbol{x}_a$, respectively.

- The minimization is carried out using Gauss Newton + Levenberg-Marquardt technique.

Motivation

- Classical retrieval methods rely on **explicit forward modeling** and **iterative minimization** (e.g., Optimal Estimation).

- These approaches are **computationally expensive** especially under cloudy-sky conditions.

- **Machine learning** methods can learn the non-linear relationships between radiances and atmospheric parameters directly from data.

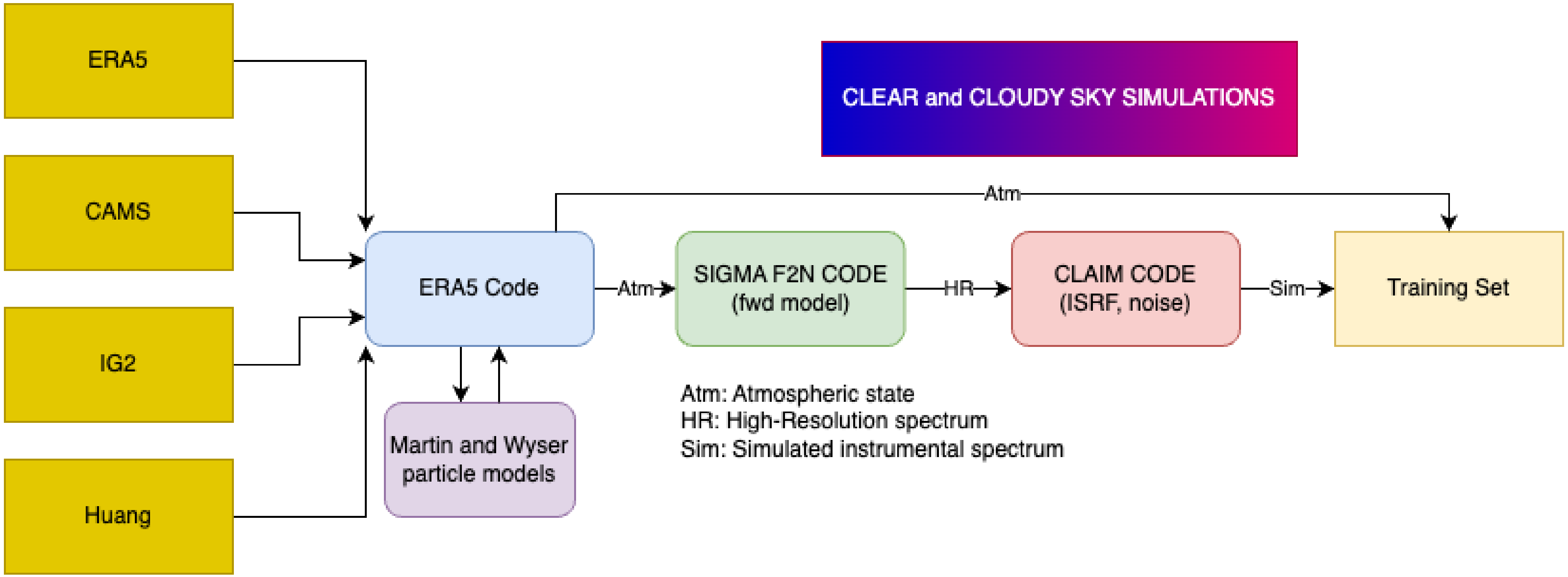
- Goal: develop a **fast, data-driven framework** capable of **real-time all-sky retrieval**, with **accurate scene recognition and cloud characterization → <u>Latent Twins Framework</u>**

- Such methods depend on large, representative **training datasets**, which must capture the variability of atmospheric and cloud conditions.

Database

- Data collected at **12:00 UTC**, covering **entire globe** on a 2°×2° lat-lon grid, **January** and **July 2021**.

- Total cases: $m = 31{,}862$ → **Training set:** $m_1 = 27{,}000$ (~85%), uniformly sampled from $\{1 \ldots m\}$.
  → **Test set:** $m_2 = 4{,}862$ (~15%), complementary subset.



Database pre-processing

- **Normalization:**
  - Min–max scaling of features to [0,1] based on **training set.**
  - Test set normalized using same min/max to avoid data leakage.

- **Vertical profile standardization:**
  - Earth's surface altitude varies → number of vertical points differs per case.
  - Normalize vertical pressures to [0, 1], interpolate each profile on **N = 60** equally spaced points → uniform input size while preserving case-specific pressure ranges.

Retrieval variables: $\mathbf{x} = \left[\mathrm{T}^0, \mathbf{T}, \mathbf{w}_{\mathrm{vap}}, \mathbf{o}, \mathbf{e}, \mathbf{c}_{\mathrm{liq}}, \mathbf{c}_{\mathrm{ice}}, \mathbf{r}_{\mathrm{liq}}, \mathbf{r}_{\mathrm{ice}} \right]^{\mathrm{T}} \in \mathrm{R}^{722}.$

Input variables: $\mathbf{y} = [\mathrm{y}_{\mathrm{lon}}, \mathrm{y}_{\mathrm{lat}}, \mathrm{d}_{\mathrm{loc}}, \mathrm{h}_{\mathrm{loc}}, \mathbf{p}, \mathbf{s}]^{\mathrm{T}} \in \mathrm{R}^{4233}.$

- $T^0, y_{lon}, y_{lat}, d_{loc}, h_{loc}, \tau_{liq}, \tau_{ice} \in R,$
- $\boldsymbol{T}, \boldsymbol{w}_{vap}, \mathbf{o}, \mathbf{c}_{\mathrm{liq}}, \mathbf{c}_{\mathrm{ice}}, \mathbf{r}_{\mathrm{liq}}, \mathbf{r}_{\mathrm{ice}}, \mathbf{p} \in R^{60}$,
- $\boldsymbol{e} \in R^{301}$ (from 100 cm^{-1} to 1600.5 cm^{-1} with step size 5 cm^{-1}),
- $\boldsymbol{s} \in R^{4169}$ (from 100 cm^{-1} to 1600.5 cm^{-1} with step size 0.36 cm^{-1}).

Autoencoders

- Neural network that learns to map its input to the output.

- Two parts:

    - **Encoder:** $\mathbf{z_x} = \mathbf{e_x}(\mathbf{x})$.
    - **Decoder:** $\mathbf{x}' = \mathbf{d_x}(\mathbf{z_x})$.

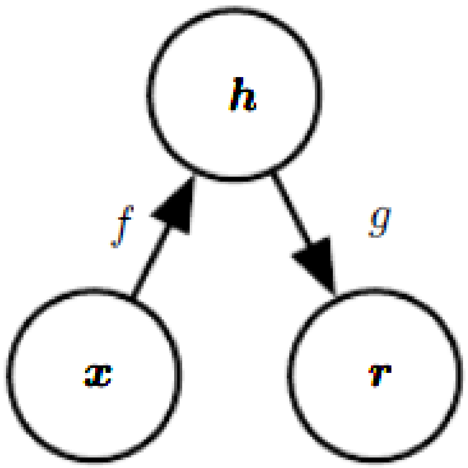
- Simply learning $\mathbf{d_x}\big(\mathbf{e_x}(\mathbf{x})\big) = \mathbf{x}$ is not effective.

- **Key Idea:**

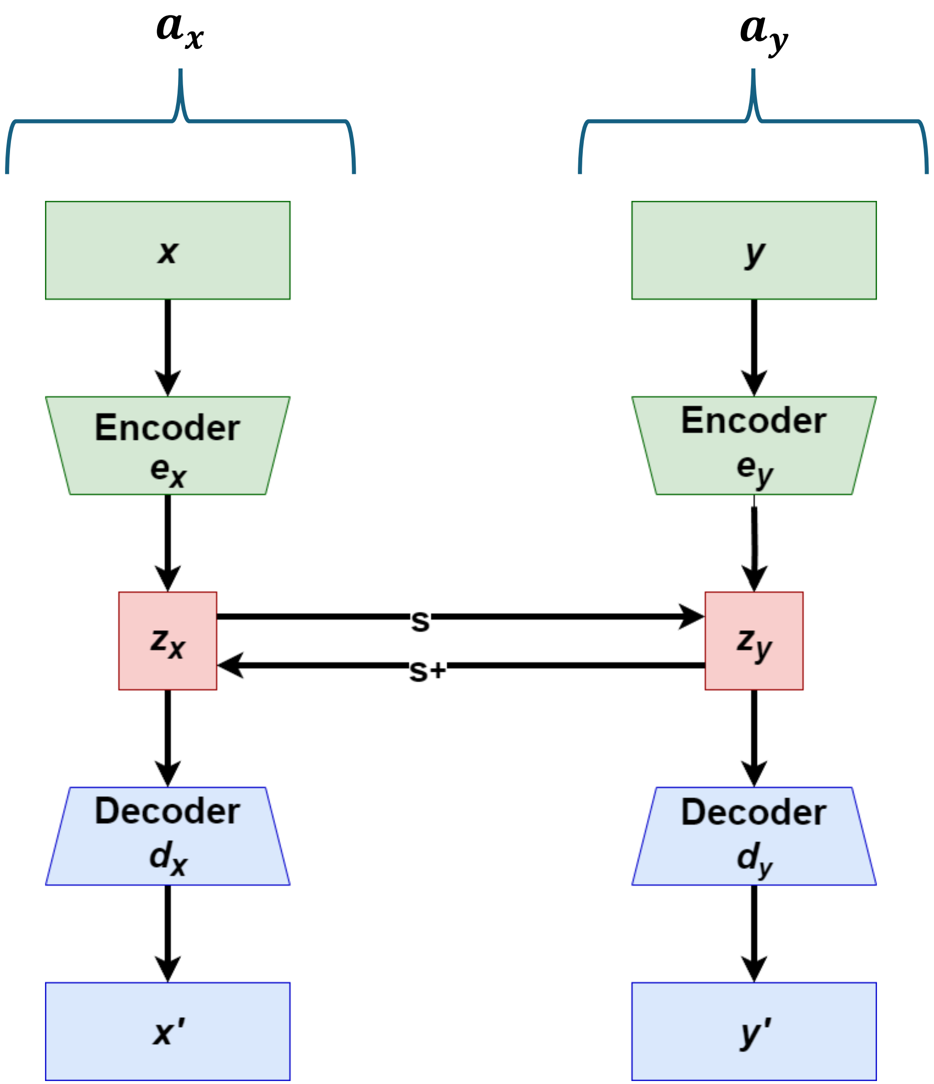
    - The model is **restricted** to approximate reconstruction.
    - Forces the model to **prioritize important features** of the data.

- Traditionally used for **dimensionality reduction** or feature learning.

- Special case of feedforward networks → same training techniques.



Latent Twins framework



(input dim + latent dim)/2

a_x: [722] → [617] → [512] → [512] → [617] → [722]

a_y: [4233] → [2372] → [512] → [512] → [2372] → [4233]

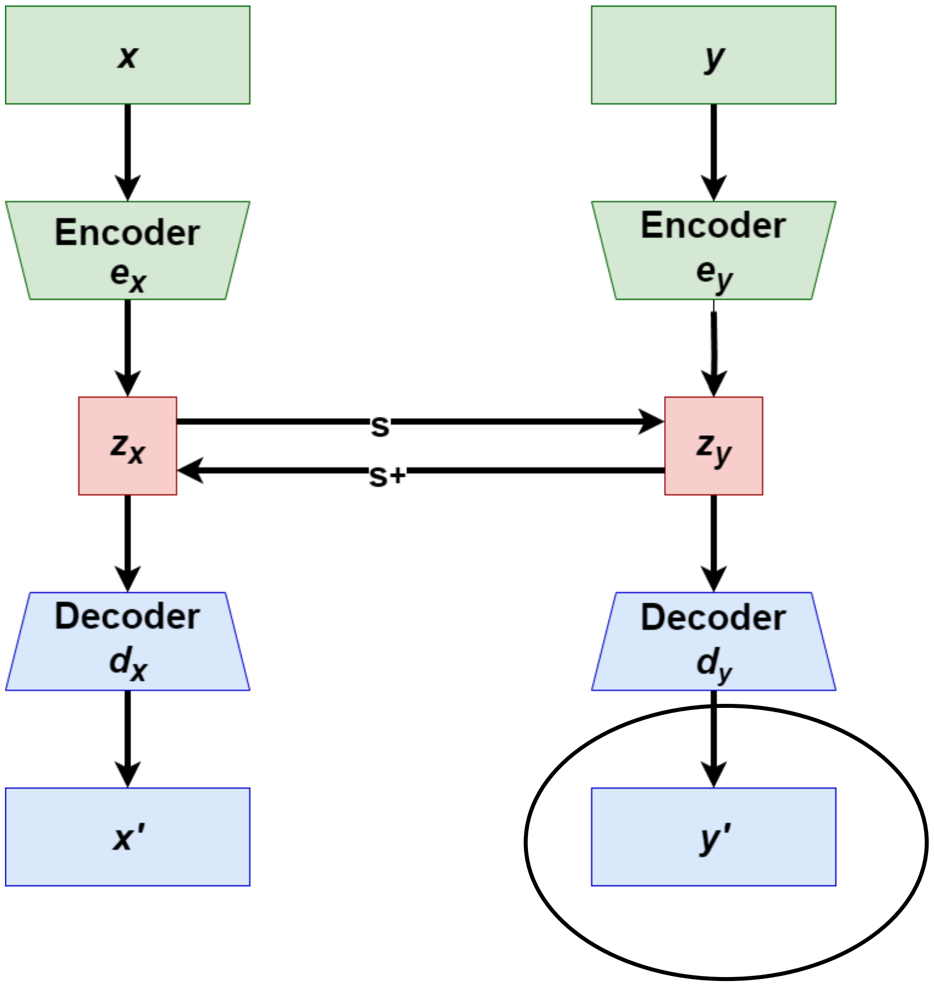
*Linear + ReLU**

s, s^+: [512] → [512]

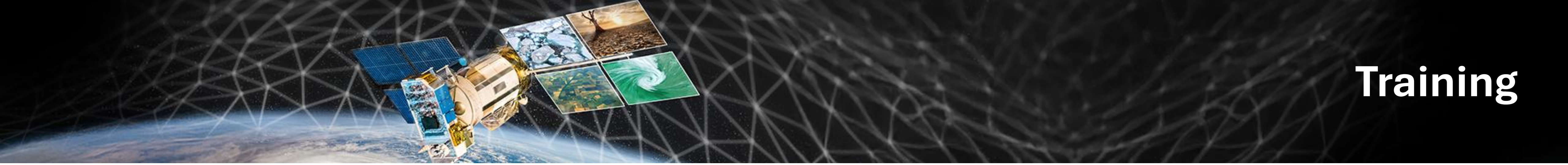
Linear

* **ReLU** (Rectified Linear Unit) **function:** $\mathbf{f(x) = max(0, x)}$ → **it introduces non linearity**

Surrogate inversion



$$f^{+}(\mathbf{y}) \approx (\mathbf{d}_{\mathrm{x}} \circ s^{+} \circ e_{y})(y)$$



Training

- **Joint learning on GPU**
- **Batch learning** (batch_size = 512)
- **Epochs** (10k)
- **ADAM (ADAptive Moment Estimation) Optimization**
- **Learnig rate: $\mathbf{10^{-3} + scheduler}$**

$$\textbf{Loss 1}: \mathrm{L}_{\mathbf{xx}} = \frac{1}{\mathrm{m}_1}\sum_{\mathrm{j}=1}^{\mathrm{m}_1}\left\|\mathbf{d}_{\mathbf{x}}\left(\mathbf{e}_{\mathbf{x}}(\mathbf{x})\right) - \mathbf{x}_{\mathrm{j}}\right\|_2^2$$

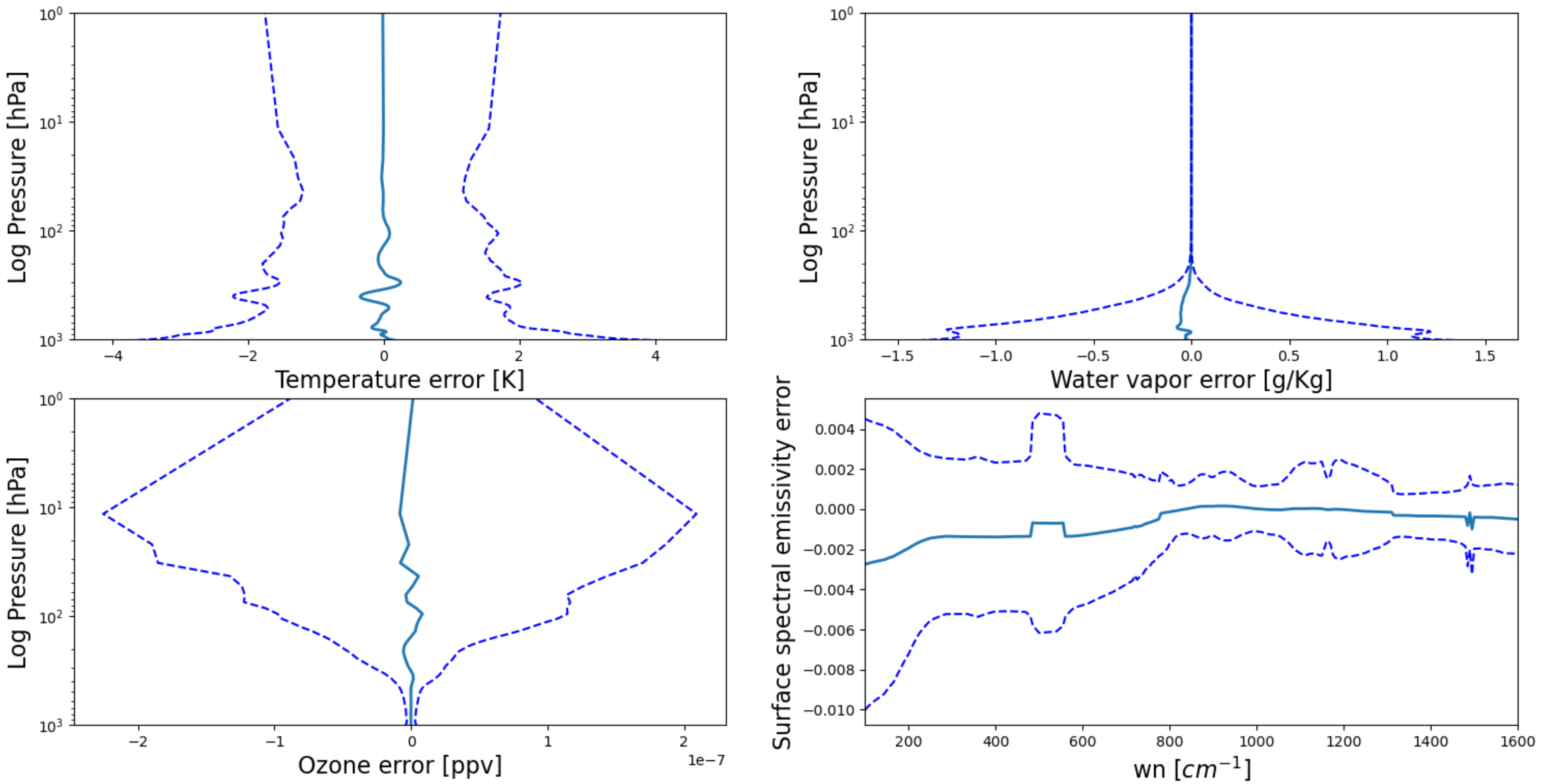
$$\textbf{Loss 2}: \mathrm{L}_{\mathbf{yy}} = \frac{1}{\mathrm{m}_1}\sum_{\mathrm{j}=1}^{\mathrm{m}_1}\left\|\mathbf{d}_{\mathbf{y}}\left(\mathbf{e}_{\mathbf{y}}(\mathbf{y})\right) - \mathbf{y}_{\mathrm{j}}\right\|_2^2$$

$$\textbf{Loss 3}: \mathrm{L}_{\mathbf{xy}} = \frac{1}{\mathrm{m}_1}\sum_{\mathrm{j}=1}^{\mathrm{m}_1}\left\|\mathbf{d}_{\mathbf{y}}\left(\mathbf{s}\left(\mathbf{e}_{\mathbf{x}}(\mathbf{x})\right)\right) - \mathbf{y}_{\mathrm{j}}\right\|_2^2$$

$$\textbf{Loss 4}: \mathrm{L}_{\mathbf{yx}} = \frac{1}{\mathrm{m}_1}\sum_{\mathrm{j}=1}^{\mathrm{m}_1}\left\|\mathbf{d}_{\mathbf{x}}\left(\mathbf{s}^{+}\left(\mathbf{e}_{\mathbf{y}}(\mathbf{x})\right)\right) - \mathbf{y}_{\mathrm{j}}\right\|_2^2$$

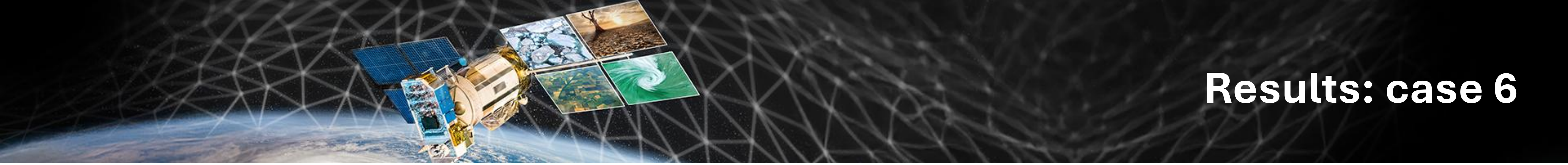
TOTAL LOSS: $\mathrm{L} = \gamma_{\mathrm{xx}}\mathrm{L}_{\mathbf{xx}} + \gamma_{\mathrm{yy}}\mathrm{L}_{\mathbf{yy}} + \gamma_{\mathrm{xy}}\mathrm{L}_{\mathbf{xy}} + \gamma_{\mathrm{yx}}\mathrm{L}_{\mathbf{yx}},$ with $\gamma_{\mathrm{xx}} = \gamma_{\mathrm{yy}} = \gamma_{\mathrm{yx}} = 1$ and $\gamma_{\mathrm{xy}} = 0.5$

Avg results

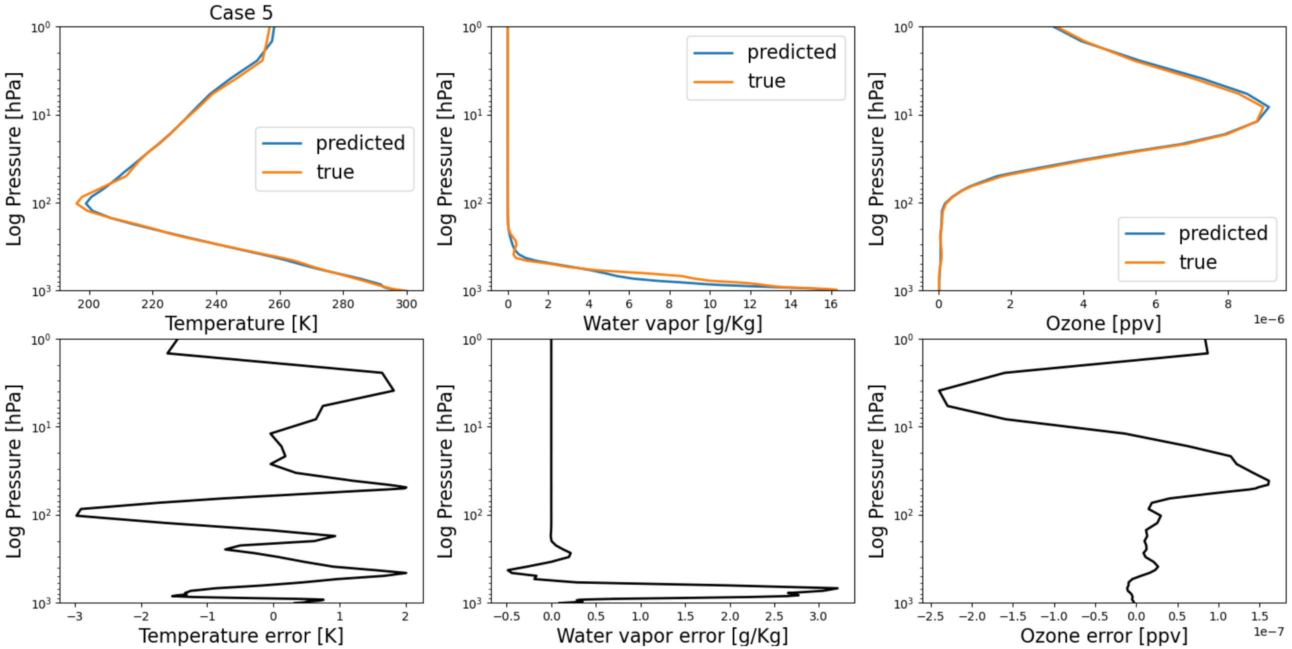


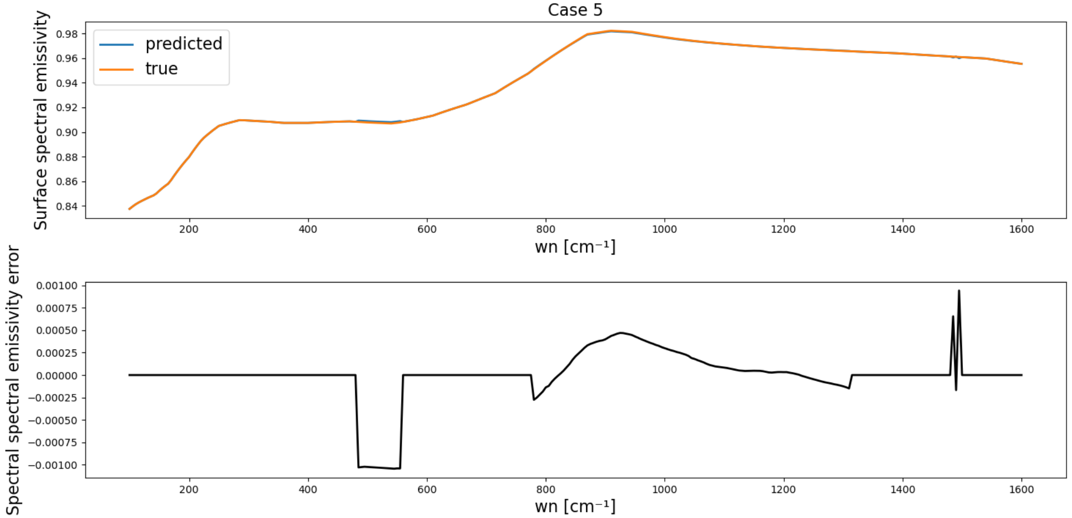
Solid line: MBE
Dashed lines: MBE ± MAE

***Surface temperature:**
MBE = 0.028K
MAE = 3.173K.



Results: case 6





Training II

- Retrieved $\mathbf{c}_{\text{liq}}, \mathbf{c}_{\text{ice}}, \mathbf{r}_{\text{liq}}, \mathbf{r}_{\text{ice}}$ are not physically consistent → **second training phase! Independent**, autoencoder & latent networks **frozen**.

- **4** Feedforward neural networks (MLP, fully-connected):

[4] → [16] → [8] → [1]

LINEAR + SOFTPLUS

INPUT: $(c^i)^{\text{inp}}$ or $(r^i)^{\text{inp}}$, p^i, T^i, w^i_{vap} **OUTPUT**: $(c^i)^{\text{new}}$ or $(r^i)^{\text{new}}$

- Training: **entire training set**; Inference: applied **only on inconsistent points**.
- **Loss function:**

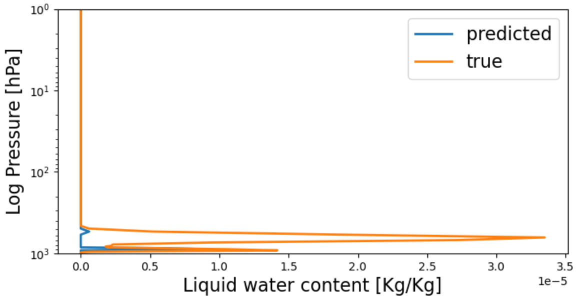
$$\mathcal{L} = \underbrace{\mathcal{L}^{\text{MSE}}_{\mathbf{c}_{\text{liq}}} + \mathcal{L}^{\text{MSE}}_{\mathbf{c}_{\text{ice}}} + \mathcal{L}^{\text{MSE}}_{\mathbf{r}_{\text{liq}}} + \mathcal{L}^{\text{MSE}}_{\mathbf{r}_{\text{ice}}}}_{\text{global MSE over all points}} + \gamma_c \underbrace{\frac{1}{N_{\text{liq}}} \sum_{i \in M_{\text{liq}}} (\hat{c}^i_{\text{liq}}\, \hat{r}^i_{\text{liq}})^2 + \frac{1}{N_{\text{ice}}} \sum_{i \in M_{\text{ice}}} (\hat{c}^i_{\text{ice}}\, \hat{r}^i_{\text{ice}})^2}_{\text{penalty on inconsistencies only}}$$

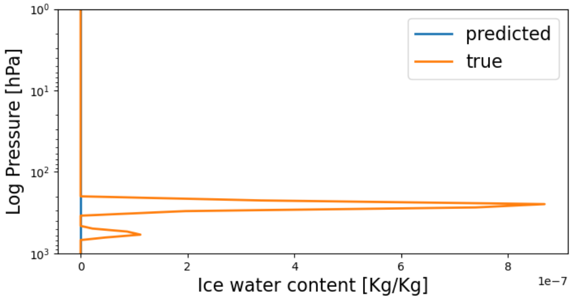
- Batch size: 512, Adam optimizer.
- **Results:** Loss: 5.6 → 0.05
- Inconsistent points: 401,681 (train), 71,048 (test) → **All inconsistencies corrected**

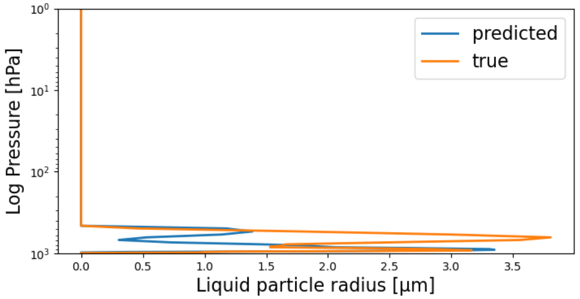
*** SoftPlus function: $f(x) = \ln(1 + e^x)$ → it avoids discontinuity**

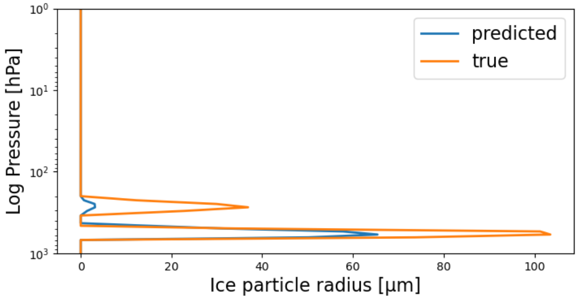
Results: case 6

BEFORE

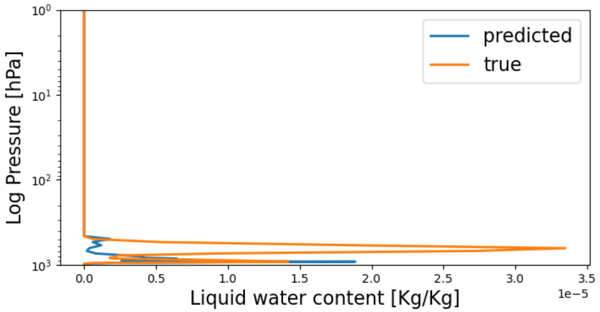


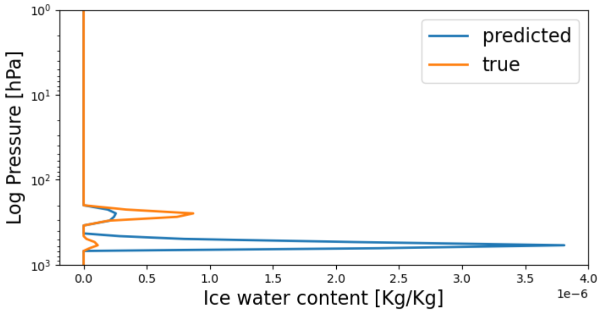


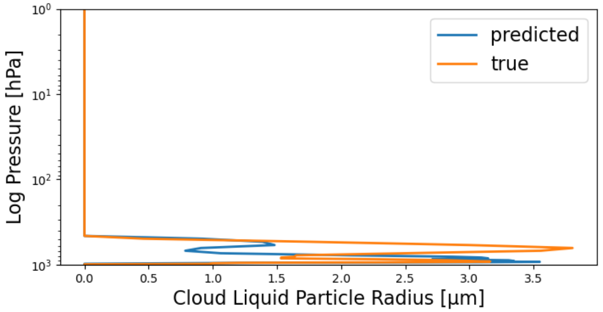


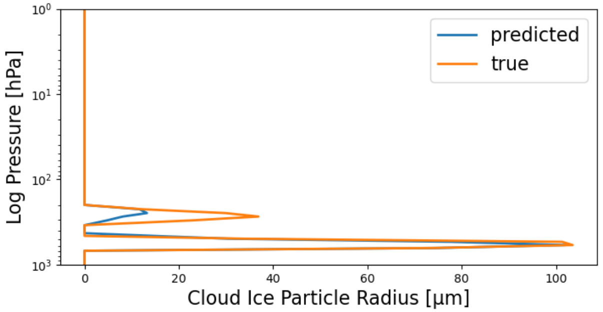


AFTER









Computational times

- Training the first 4 models → it depends on the epochs and laptops → some hours
- Training the last 4 networks → it depends on the epochs and laptops → some minutes

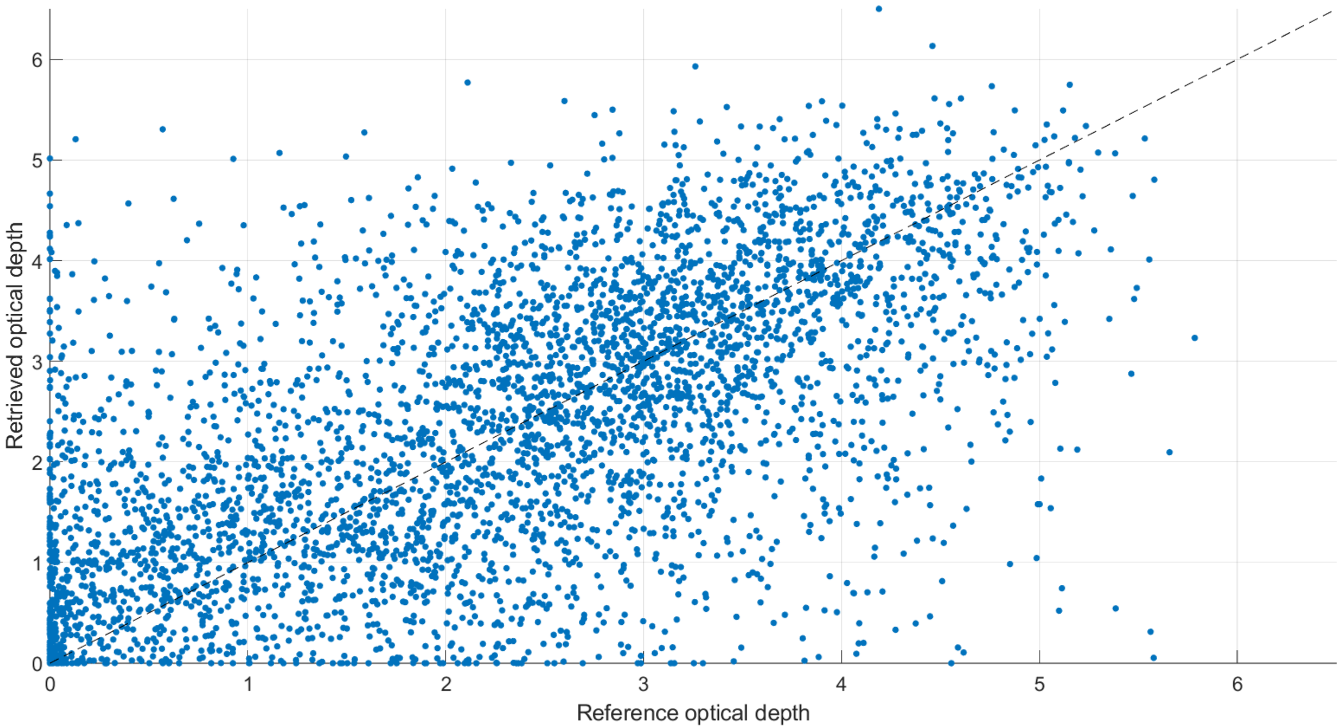
- Testing → **Total time taken for all predictions (4862): 2.06 seconds***
  Average time per batch (512): 0.21 seconds*

* standard laptop

Optical depth reconstruction

$$\tau_{ice} = \frac{3}{2}\sum_{i} \frac{Q_{ext}^{i} c_{ice}^{i} \rho_{air}}{r_{ice}^{i} \rho_{ice}} \Delta z^{i},$$

with i layer, ρ_{air} and ρ_{ice} air and ice density, respectively, Δz^{i} geometric thickness and Q_{ext}^{i}. The same for τ_{liq}.



MBE = 3.40
MAE = 20.12

* Logarithmic scale with a unit offset

Scene Classification Performance

$\tau = \tau_{liq} + \tau_{ice}$ used to assess the scene classification between clear and cloudy conditions.

CATEGORY	DEFINITION	ACCURACY
Binary: Clear / Cloudy	Clear if $\tau \le 0.03$, Cloudy if $\tau > 0.03$	**94.82 %**
Three classes: Clear / Thin / Thick	Clear: $\tau \le 0.03$ • Thin: $0.03 < \tau \le 1$• Thick: $\tau > 1$	**82.91 %**

Cloud Position Accuracy: Fourier Insights

- **Goal:** assess whether the network captures the *vertical structure and position* of clouds

- **Method:** compute **Fourier sine-series coefficients** $b_k, k = 1, \dots, 10$ of true and predicted profiles:

$$b_k \approx \frac{2}{p^N - p^1} \sum_{i=1}^{N} \omega^i c_{liq}^i \, sin\left(\frac{k\,\pi\,(p^i - p^1)}{p^N - p^1}\right), \qquad \omega^i = \begin{cases} \frac{1}{2}(p^2 - p^1) & i = 1 \\ \frac{1}{2}\left(p^{i+1} - p^{i-1}\right) & i = 2, \dots, N-1 \\ \frac{1}{2}(p^N - p^{N-1}) & i = N \end{cases}$$

- **Normalization:** $\boldsymbol{b} = [\boldsymbol{b_1}, \boldsymbol{b_2}, \dots, \boldsymbol{b_K}], \quad \widetilde{\boldsymbol{b}} = \frac{\boldsymbol{b}}{\boldsymbol{max_i\,|b_i|}}$

- **Cosine similarity:** for each cloud variable and case, compute

$$\boldsymbol{cos_{sim}} = \frac{\widetilde{\boldsymbol{b}}^{\boldsymbol{true}} \widetilde{\boldsymbol{b}}^{\boldsymbol{pred}}}{\left\|\widetilde{\boldsymbol{b}}^{\boldsymbol{true}}\right\| \left\|\widetilde{\boldsymbol{b}}^{\boldsymbol{pred}}\right\|}$$

Var	$\boldsymbol{cos_{sim}}$
c_{ice}	0.67
r_{ice}	0.67
c_{liq}	0.73
r_{liq}	0.73

Take-home messages

- **Latent-twin architecture** successfully reconstructs atmospheric profiles and cloud properties.

- **Additional neural networks and physical constraints** improve the cloud-property retrieval.

- **Optical depth retrieval** enables accurate **scene recognition**:
  - ~95% accuracy in distinguishing clear vs. cloudy scenes
  - ~83% accuracy in identifying **thick / thin / clear** classes

- Retrievals are **instantaneous**, suitable for **real-time applications**.

- **Fourier analysis** confirms that the model **correctly captures the vertical cloud position** on average.

- Profile amplitude retrieval still requires **further improvement**.

- **Ongoing collaboration** with *Michele Martinazzo (University of Bologna), Chiara Zugarini (University of Florence),* and *Marco Menarini (University of Bologna)* aims to apply this architecture to **clear-sky retrievals** using **real IASI measurements** and and to perform a **comprehensive error and robustness analysis** of the method.

Bibliography

[1] Rodgers, C. D.: Inverse Methods for Atmospheric Sounding, World Scientific, https://doi.org/10.1142/3171, 2000.
[2] Matthias Chung *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 045055 DOI: 10.1088/2632-2153/ad95dd
[3] G. Martin, D. Johnson, and A. Spice. "The Measurement and Parametrization of Effective Radius of Droplets in Warm Stratocumulus Clouds". In: Journal of Atmospheric Sciences 51 (13 1994), pages 1823–1842.
[4] K. Wyser. "The Effective Radius in Ice Clouds". In: Journal of Climate 11.7 (1998), pages 1793–1802. issn: 08948755, 15200442.
[5] Masiello G., Serio C., Maestri T., Martinazzo M., Masin F., Liuzzi G., et al. (2024). The new σ-IASI code for all sky radiative transfer calculations in the spectral range 10 to 2760 cm^{-1}: σ-IASI/F2N. JOURNAL OF QUANTITATIVE SPECTROSCOPY & RADIATIVE TRANSFER, 312, 1-19.
[6] Huang, X. L., X.H. Chen, D. K. Zhou, X. Liu, An observationally based global band-by-band surface emissivity dataset for climate and weather simulations, Journal of the Atmospheric Sciences, 73, 3541-3555,.1, 2016.
[7] L. Sgheri, C. Sgattoni, and C. Zugarini. "Determination of emissivity profiles using aBayesian data-driven approach". In: Mathematics and Computers in Simulation 229(2025), pages 512–524. issn: 0378-4754.
[8] *C. Sgattoni, L. Sgheri, and M. Chung. "A data-driven approach for fast atmospheric radiative transfer inversion". In: Inverse Problems 41.8 (2025), page 085006.*
[9] *C. Sgattoni, L. Sgheri and M. Chung. "A physics-aware data-driven surrogate approach for fast atmospheric radiative transfer inversion". Living Planet Symposium 2025, Vienna.*

ACKNOWLEDGMENTS:

This work has been partially funded by **NRRP-EMM** (National Recovery and Resilience Plan - Earth Moon Mars) project, Mission 4, Component 2, Investment 3.1, Project IR000038, CUPC53C22000870006 and **INdAM-GNCS.**

Thank you for your attention!