Fast Inversion of Hyperspectral Observations of Green House Gases (GHG) using Gaussian Locally Linear Mapping (GLiMM)

SFTP 2025 – Grasse

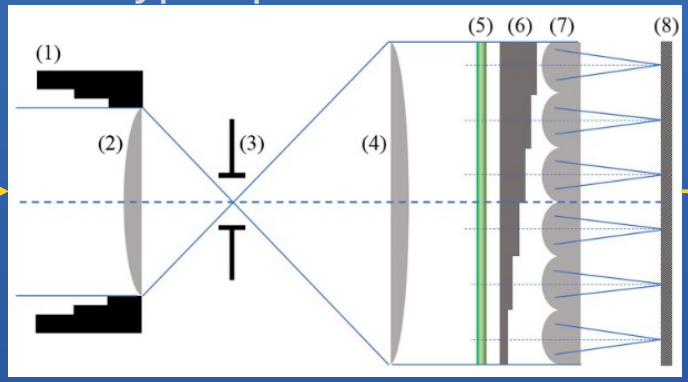
Eric Tatulli, Silvere Gousset, Sylvain Doute, Luc Meyer

IPAG, UGA/CNRS

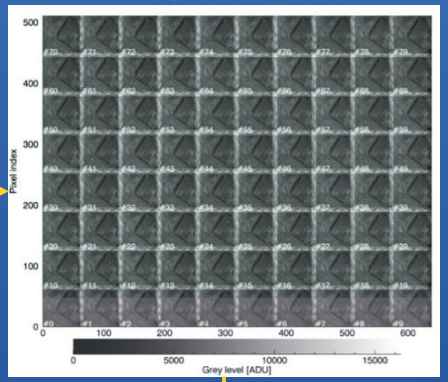
Nanocarb: principles

Scenery

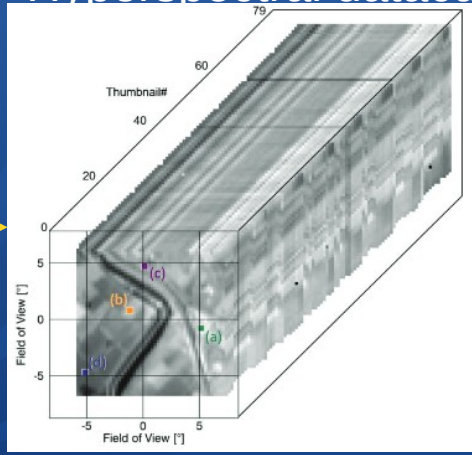
Hyperspectral Instrument

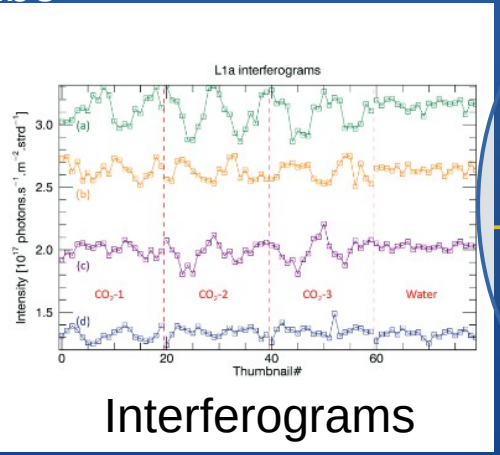


Thumbnails



Hyperspectral datacube





Interferograms

LM iterative algorithm

Atmospheric Inversion

GLiMM!

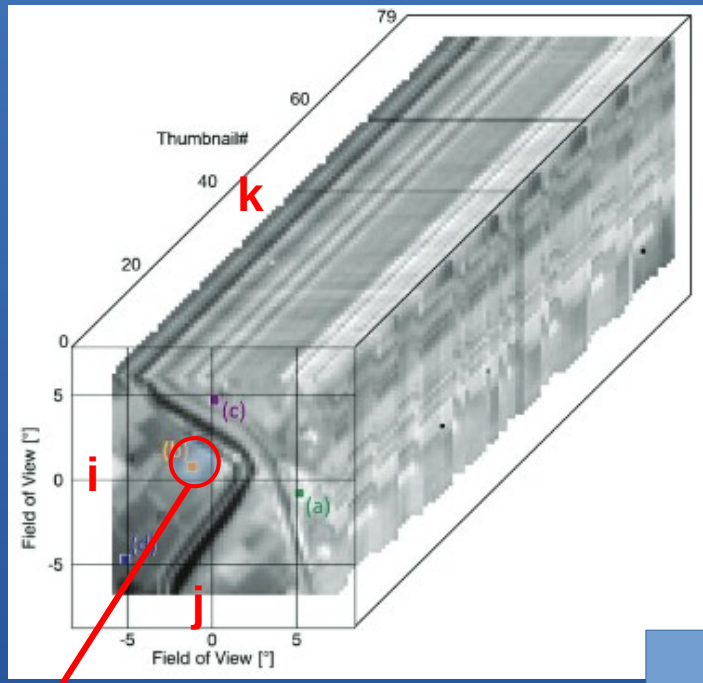
Albedo

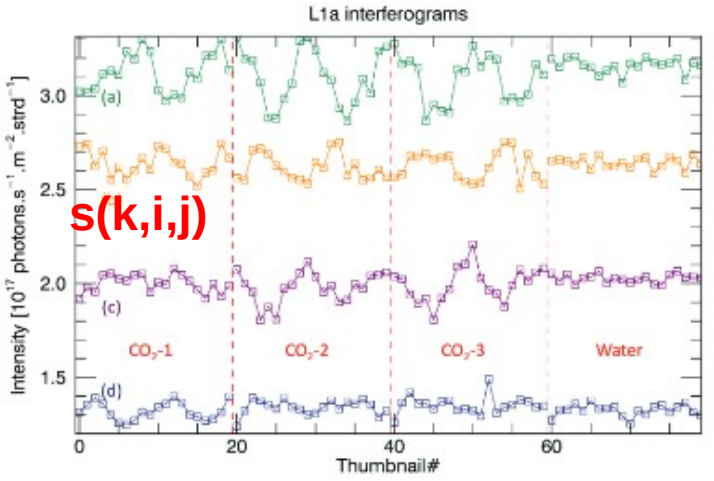
X_{CO2}

X_{CH4}

X_{H20}

...





Instrumental transfer function

Luminance

$$s^{k}_{(i,j)}\ (\mathrm{ADU}) = \tau.\kappa \int \eta^{k}_{(i,j)}(\sigma).L_{(i,j)}(\sigma)\ d\sigma + \varepsilon$$

Albedo, X_{CO2}, (X_{CH4}, $X_{H20...}$)

$$y = \mathbf{F}(x) + \varepsilon$$

Classical Inversion

- Levenberg-Marquardt (LM) iterative inversion
  - Modeling: $\boldsymbol{y} = \mathbf{F}(\boldsymbol{x}) + \boldsymbol{\varepsilon}$

    Measurement noise and covariance
  - Metrics:

$$\chi^2 = (\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{x}))^{\mathrm{T}} \mathbf{S}_{\mathrm{e}}^{-1} (\boldsymbol{y} - \mathbf{F}(\boldsymbol{x})) + (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{a}})^T \mathbf{S}_{\mathrm{a}}^{-1} (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{a}}),$$

    a priori and covariance
  - Linearization around x_a

$$\mathbf{F}(\boldsymbol{x}) = \mathbf{F}(\boldsymbol{x}_{\mathrm{a}}) + \frac{\partial \mathbf{F}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_{\mathrm{a}}} (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{a}})$$

    Jacobian K

Classical Inversion

- Levenberg-Marquardt (LM) iterative inversion

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \left[(1+\gamma)\,\mathbf{S}_{\mathrm{a}}^{-1} + \mathbf{K}_i^{\mathrm{T}}\mathbf{S}_{\mathrm{e}}^{-1}\mathbf{K}_i\right]^{-1} \left(\mathbf{K}_i^{\mathrm{T}}\mathbf{S}_{\mathrm{e}}^{-1}\left(\boldsymbol{y} - \mathbf{F}\left(\boldsymbol{x}_i\right)\right) - \mathbf{S}_{\mathrm{a}}^{-1}\left(\boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{a}}\right)\right)$$

- Hypothesis
  - Initial values of X and associated covariance S_a
  - Iteration stop criterion
- References
  - Rodgers, C. D. (2000). *Inverse methods for atmospheric sounding: theory and practice*, World scientific.
  - Herbin et al. (2013); Dogniaux et al. (2020)

GLiMM principles

- Bayesian approach (gaussian mixtures)
  - X and Y are linked by combination of K affine transformations τ_k
    - Modeled by a missing variable Z such that Z = k if Y is the image of X by τ_k

$$Y = \sum_{k=1}^{K} \mathbb{I}_{Z=k} \left(\boldsymbol{A}_k X + \boldsymbol{b}_k + \epsilon_k \right)$$

$\mathbb{I}_{Z=k}$ Indicator function

$$p(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\theta}) = \sum_{k=1}^{K} p(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}, Z = k; \boldsymbol{\theta}) \, p(\boldsymbol{X} = \boldsymbol{x} | Z = k; \boldsymbol{\theta}) \, p(Z = k; \boldsymbol{\theta}).$$

Vector of model parameters

$$\boldsymbol{\theta} = \{ c_k, \boldsymbol{\Gamma}_k, \pi_k, \mathbf{A}_k, \boldsymbol{b}_k, \boldsymbol{\Sigma}_k \}_{k=1}^{K}$$

$$p(\boldsymbol{X} = \boldsymbol{x} | Z = k; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}; c_k, \boldsymbol{\Gamma}_k)$$

Normal law

$$p(Z = k; \boldsymbol{\theta}) = \pi_k,$$

Uniform law

References

- Deleforge et al (2015), *High-dimensional regression with gaussian mixtures and partially-latent response variables,* Stat. Comput.
- Kluger (2021);

GLiMM Inversion

Forward Modeling

$$p_G(\mathbf{y}|\mathbf{X}=\mathbf{x},\boldsymbol{\theta}) = \sum_{k=1}^{K} \eta_k(\mathbf{x})\mathcal{N}(\mathbf{y}; \boldsymbol{A}_k\mathbf{x}+\boldsymbol{b}_k, \boldsymbol{\Sigma}_k)$$

$$\text{with } \eta_k(\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}; \boldsymbol{c}_k, \boldsymbol{\Gamma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}; \boldsymbol{c}_j, \boldsymbol{\Gamma}_j)}$$

Estimation through Ntrain simulations of observations

Needs a grid of X parameters: *Sobol distribution*

Inverting

$$p_G(\mathbf{x}|\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}^*) = \sum_{k=1}^{K} \eta_k^*(\mathbf{y})\mathcal{N}(\mathbf{x}; \boldsymbol{A}_k^*\mathbf{y}+\boldsymbol{b}_k^*, \boldsymbol{\Sigma}_k^*)$$

$$\text{with } \eta_k^*(\mathbf{y}) = \frac{\pi_k \mathcal{N}(\mathbf{y}; \boldsymbol{c}_k^*, \boldsymbol{\Gamma}_k^*)}{\sum_{j=1}^{K} \pi_j^* \mathcal{N}(\mathbf{y}; \boldsymbol{c}_j^*, \boldsymbol{\Gamma}_j^*)}.$$

Predicting

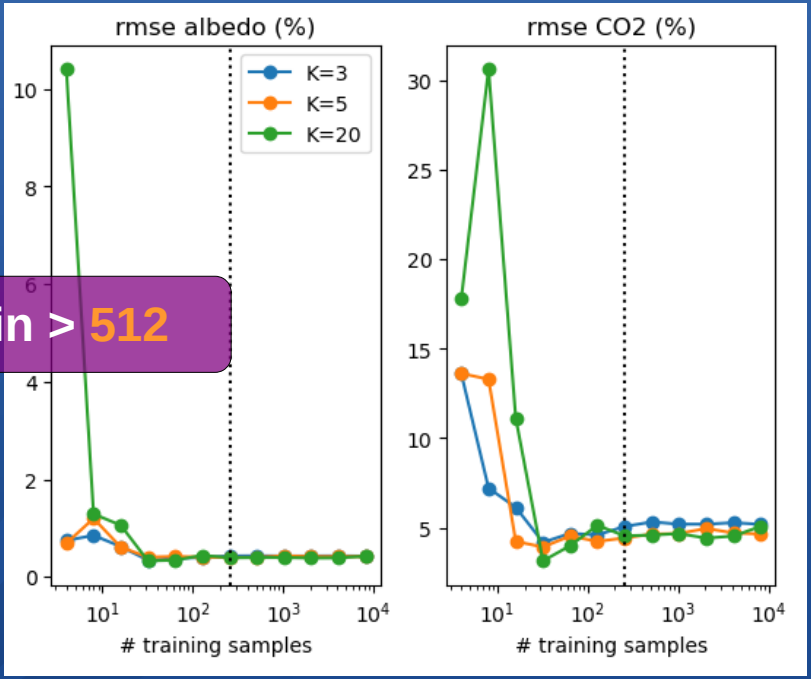
$$\bar{\mathbf{x}}_G(\mathbf{y}) = \mathbb{E}_G\left[\mathbf{X} \mid \mathbf{Y}=\mathbf{y}, \boldsymbol{\theta}^*\right] = \sum_{k=1}^{K} \eta_k^*(\mathbf{y})\left(\boldsymbol{A}_k^*\mathbf{y}+\boldsymbol{b}_k^*\right).$$

single iFOV observations

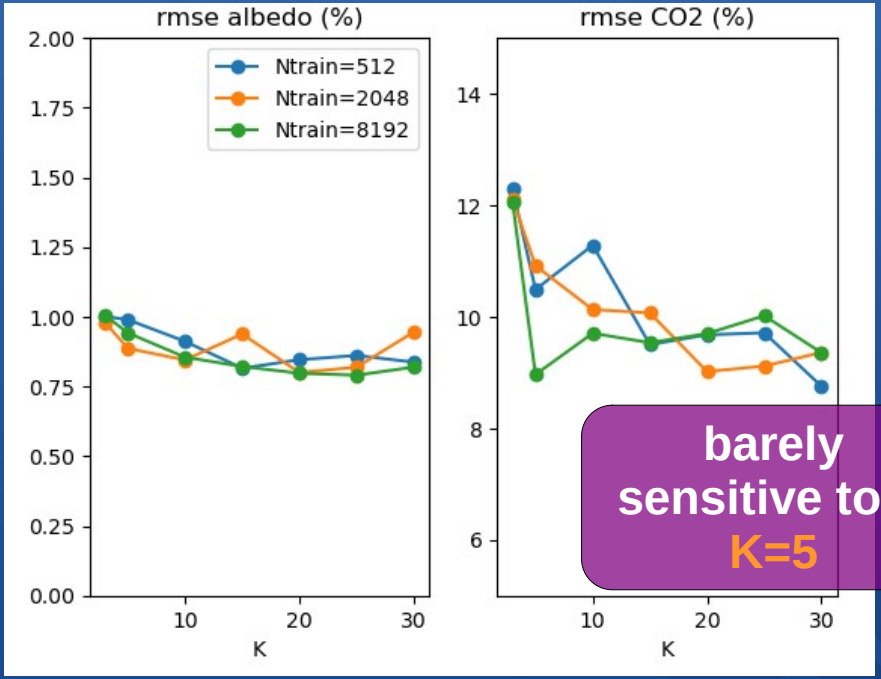
- Modeling - Training
  - Y = simulated interferograms, X = [CO2 scale factor, albedo]
  - Atmospheric model (4AOP) + Instrumental model (Nanocarb)
  - Aerosols: assumed perfectly known (dedicated SPEX instrument)
  - GLiMM input parameters [K, Ntrain]
  - X grid: CO2 [0.8-1.4], albedo [0.1-0.8]
- Testing: N=100 random draws of noise (photon + detector)
- Comparing with Levenberg Marquardt Inversion
  - x_a = [1.0, true albedo], σ_a = [0.1, 0.1]

GLiMM settings

- Choosing **K** and **Ntrain** (in terms of RMSE)

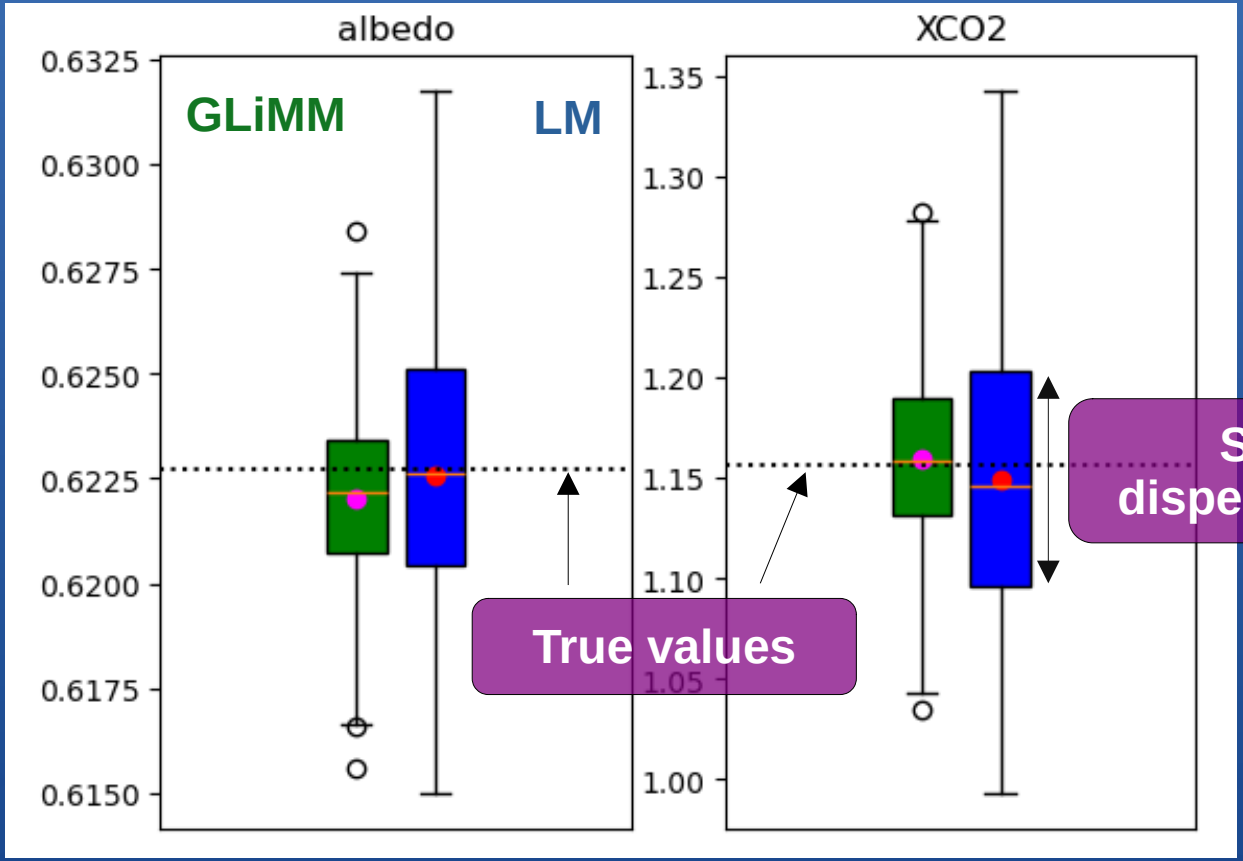


Ntrain > 512



barely sensitive to K
K=5

Inversion results

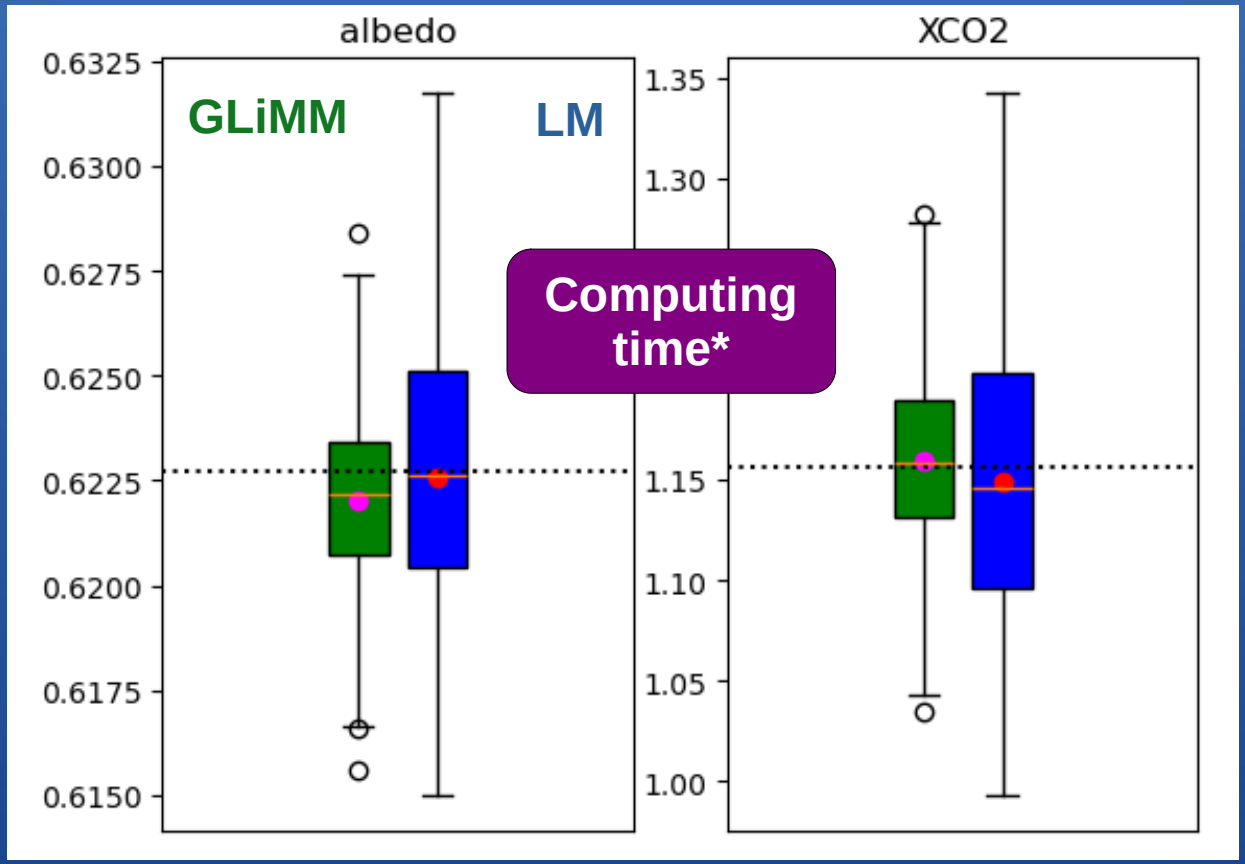


Inversion results

GliMM
~ 0.4s

Running grid of simulations + training

~6 min



Computing time*

LM
~ 15min

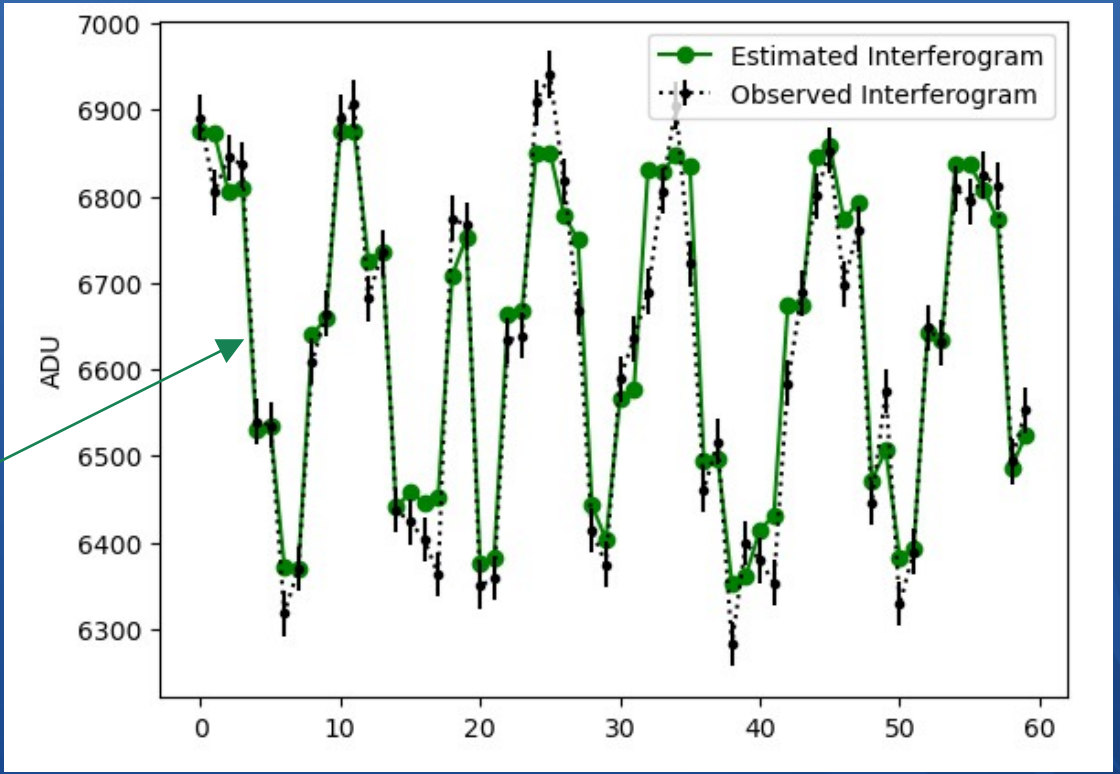
* PC Intel Core i7-13800H CPU @ 1.4GHz – 30 GB of RAM

GLiMM Forward Model

Estimated X

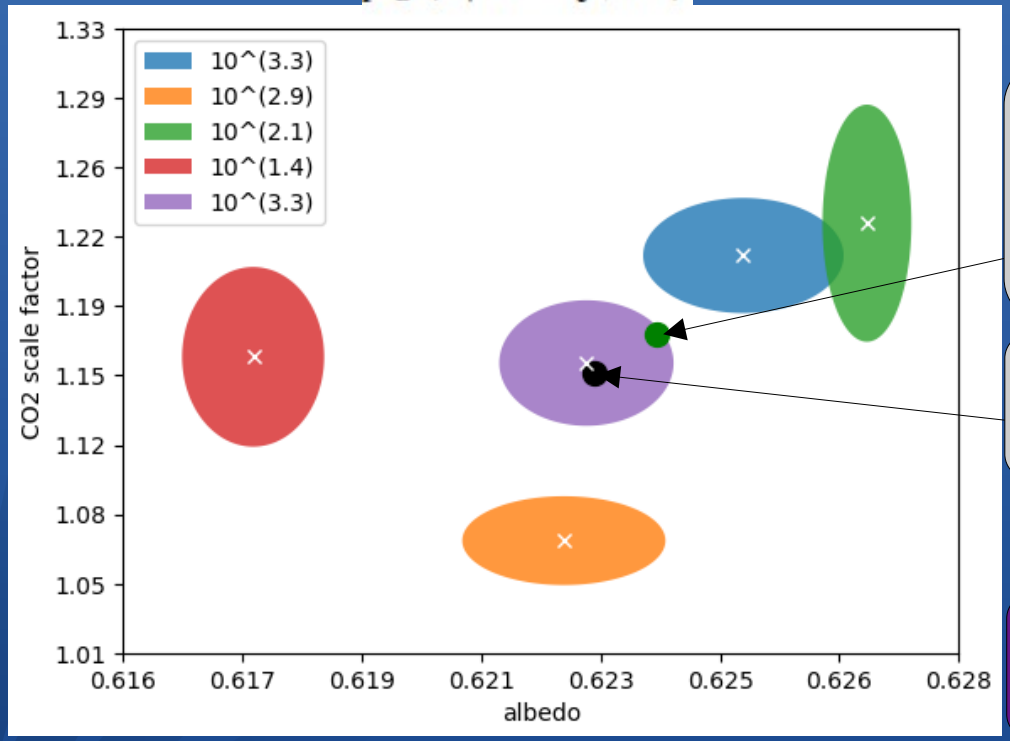
$$\mathbb{E}\left[Y|X=\mathbf{x}\right]=\sum_{k=1}^{K}\alpha_k(\mathbf{x})\left(\boldsymbol{A}_k\mathbf{x}+\boldsymbol{b}_k\right)$$

Fit of the interferogram



GLiMM backward model

$p_G(\mathbf{x}|\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}^*)$



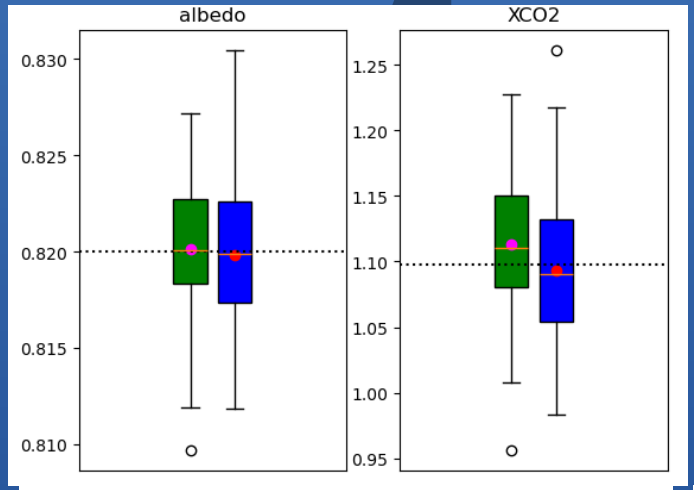
Estimated value

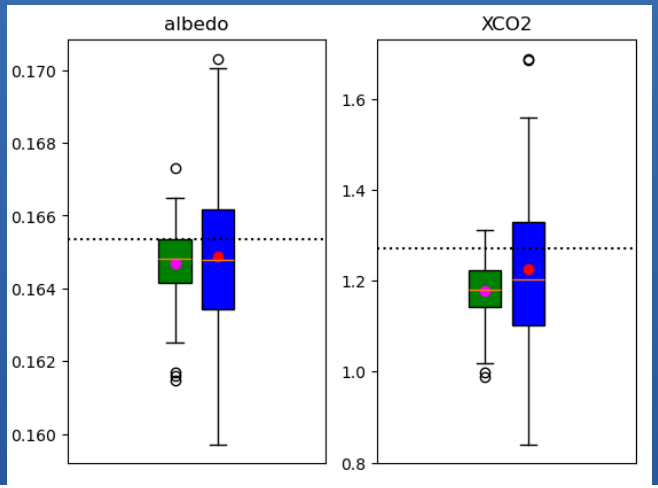
$$\bar{\mathbf{x}}_G(\mathbf{y}) = \mathbb{E}_G\left[\mathbf{X} \mid \mathbf{Y}=\mathbf{y},\boldsymbol{\theta}^*\right] = \sum_{k=1}^{K} \eta_k^*(\mathbf{y})\left(\boldsymbol{A}_k^*\mathbf{y}+\boldsymbol{b}_k^*\right).$$

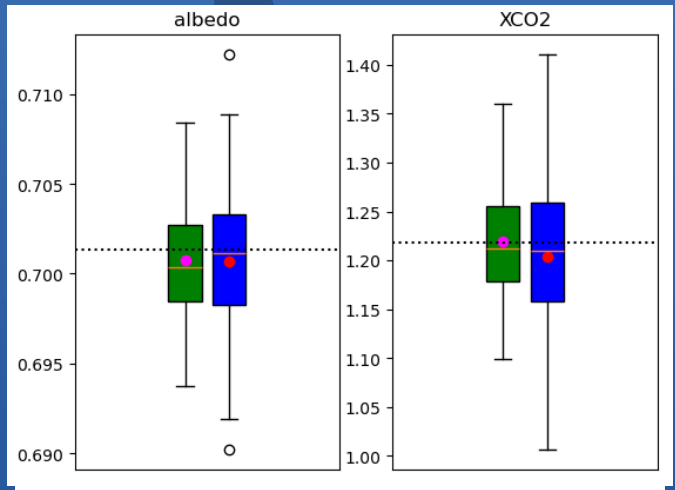
True value

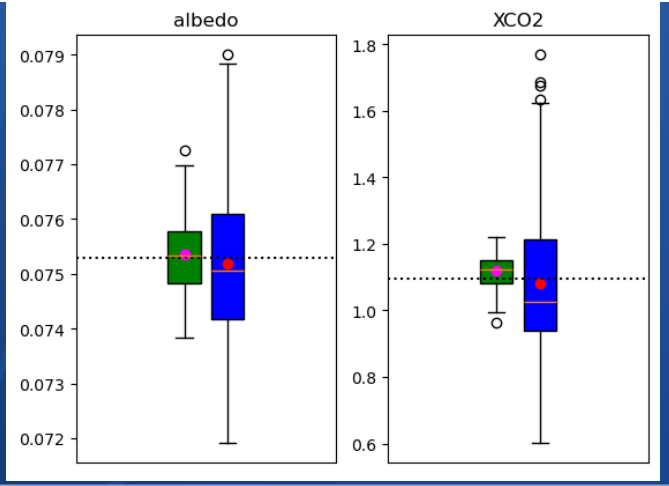
Several outcomes with similar probability
→ more subtle ways of estimating X

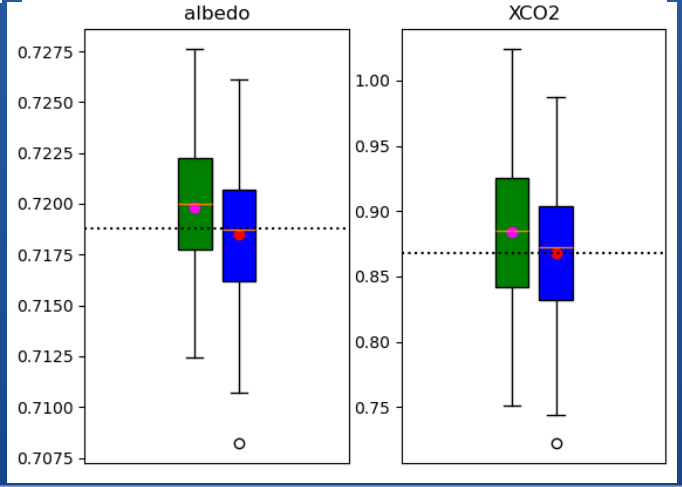
Random examples

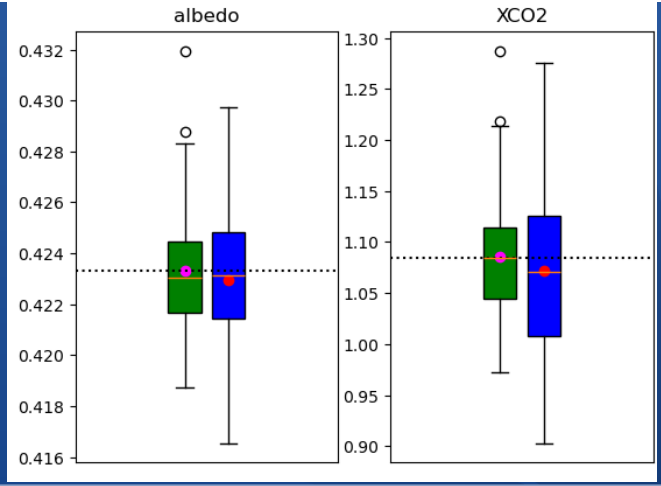




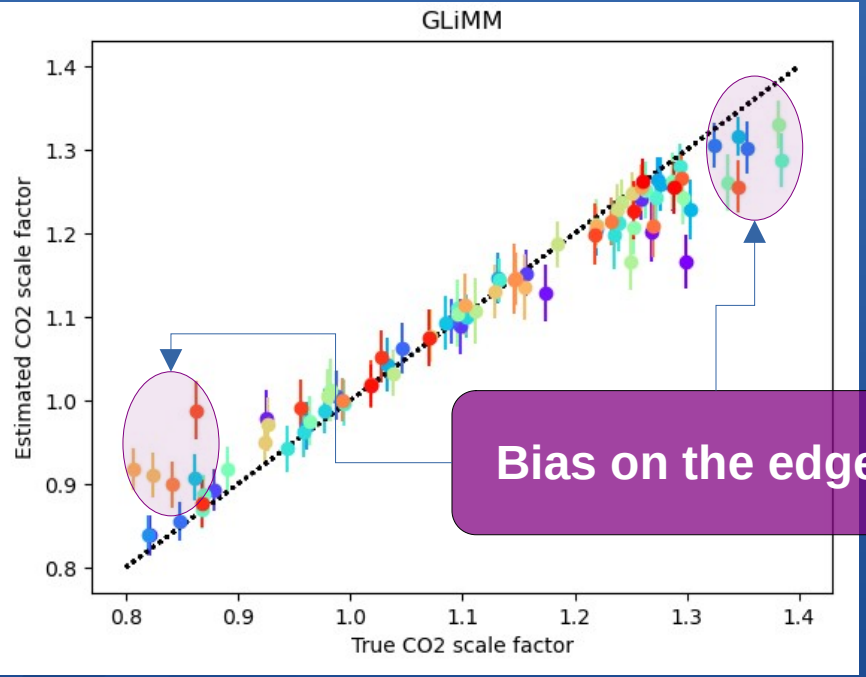




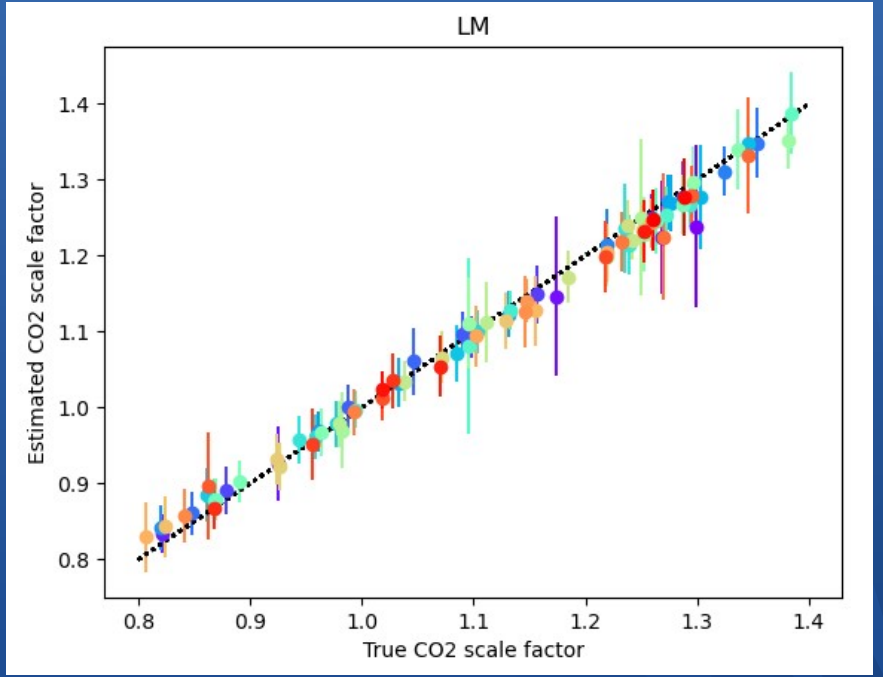




Spanning the CO2 range



Bias on the edges?



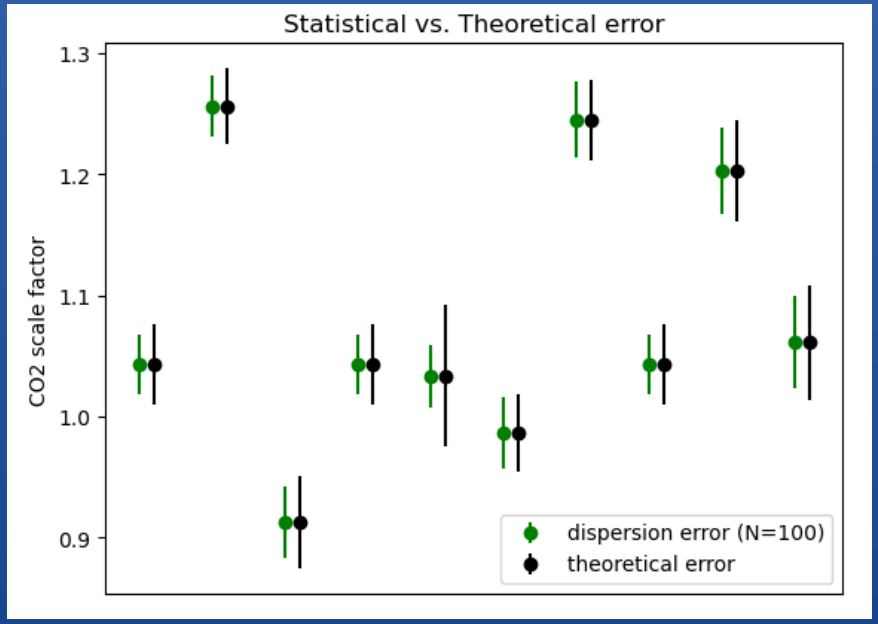
GLiMM vs. LM at a glance

	Time (100 samples)	Underlying Assumptions	Training set	Bias
LM	~ 15 min	x_a, S_a stop criterion	-	No
GLiMM	~ 0.4 sec (+ 6 min)	grid of parameters K, Ntrain	Yes (> 512)	On edges? (calibratable?)

Accuracy of CO2 estimates

- Statistical dispersion vs. theoretical error

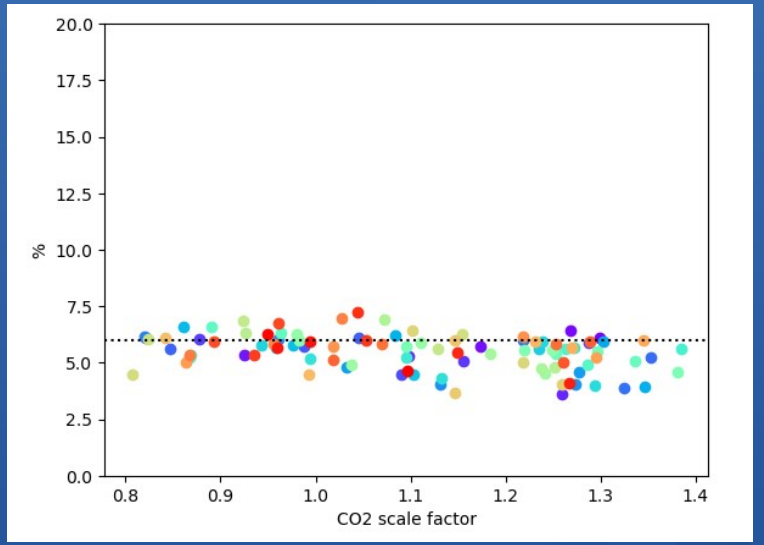
$$Var(X|Y=\mathbf{y}) = \sum_{k=1}^{K} \alpha_k^*(\mathbf{y}) \left[\boldsymbol{\Sigma}_k^* + (\boldsymbol{A}_k^*\mathbf{y} + \boldsymbol{b}_k^*)^{\mathrm{t}}(\boldsymbol{A}_k^*\mathbf{y} + \boldsymbol{b}_k^*)\right] - \overline{\mathbf{x}}_G(\mathbf{y})^{\mathrm{t}}\overline{\mathbf{x}}_G(\mathbf{y})$$



GLiMM provides **reliable error bars** with proper order of magnitude

Accuracy of CO2 estimates

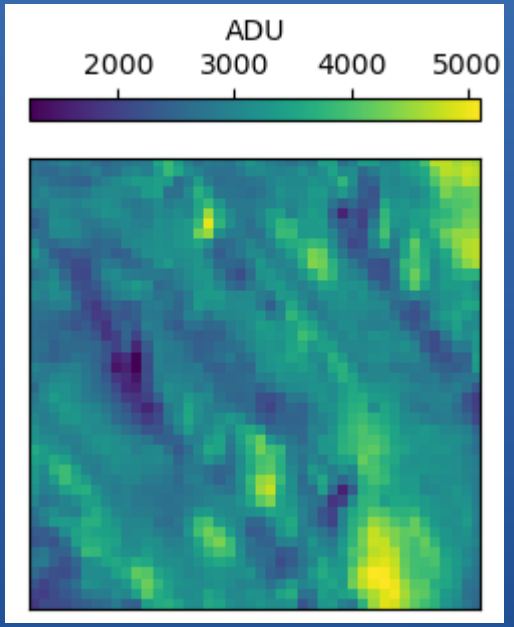
- CO2 accuracy of a single observation
  - $\sigma CO_2/X_{CO2} \sim 6\%$



- 4 ppm accuracy ↔ 1% in scale factor
  - Need average over > 150 observations (SNR>2)
  - Temporal (1 iFOV over ~200 snapshots)
  - Spatial (1 snapshot over 50x50 pixels)

CO2 Plume Detection

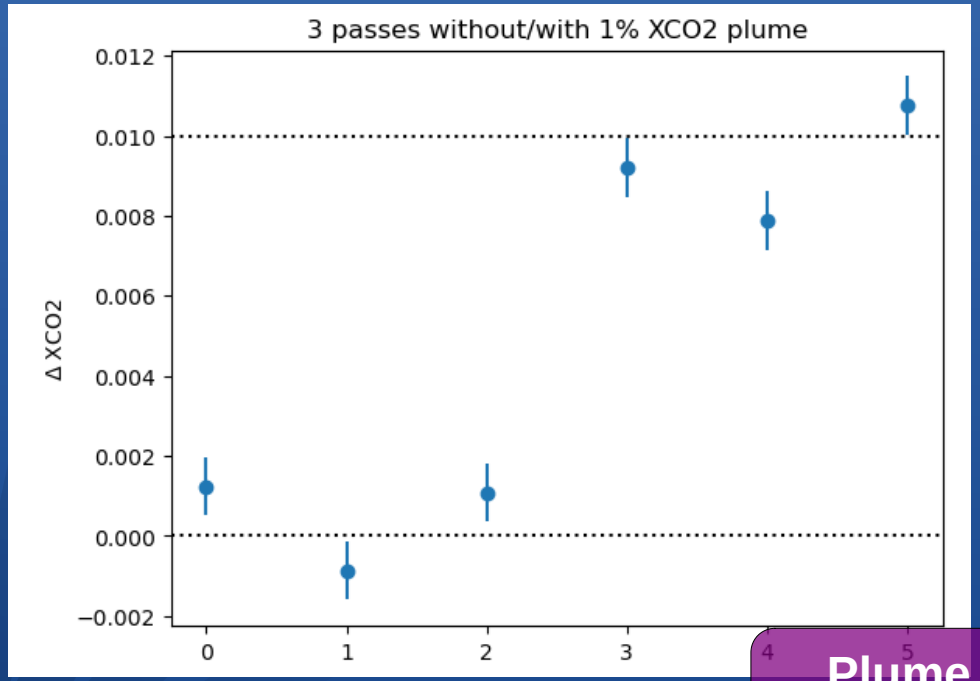
- Flying over a CO2 plume
  - Belchatow 2020 scenery
  - 44 x 44 Snapshots
  - 1% CO2 scale factor variation
    (no plume vs. plume)

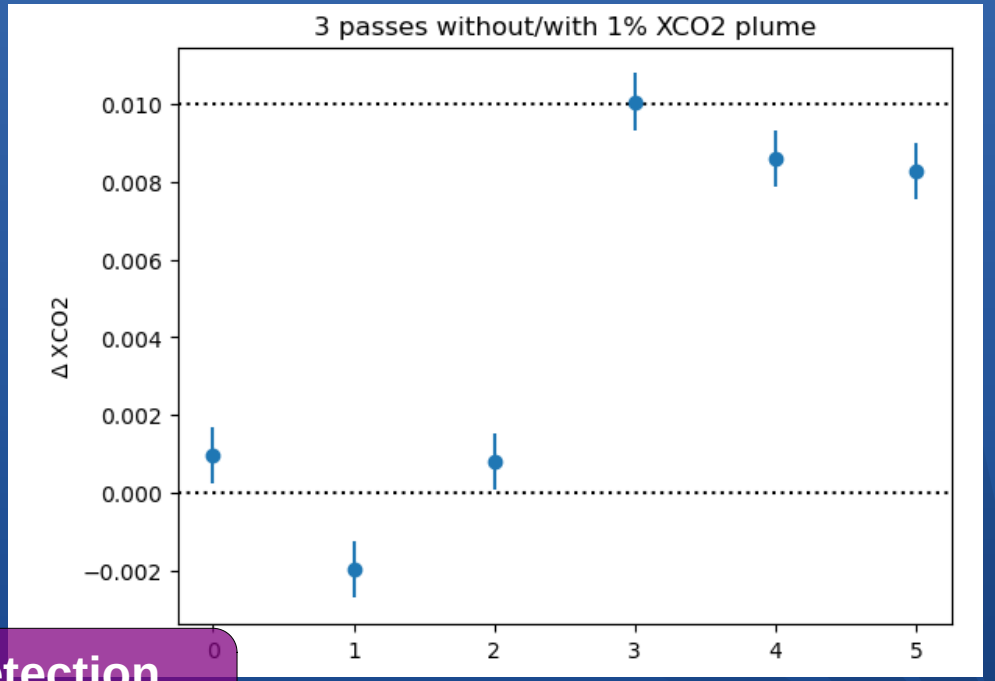


- Are we able to detect the plume?
  - Applying GLiMM to each pixel of the image
  - Y = interferogram + viewing angle, X = [CO2, albedo]
  - Airplane passes without then with CO2 plume

CO2 Plume Detection

- Scenery with/without CO2 plume

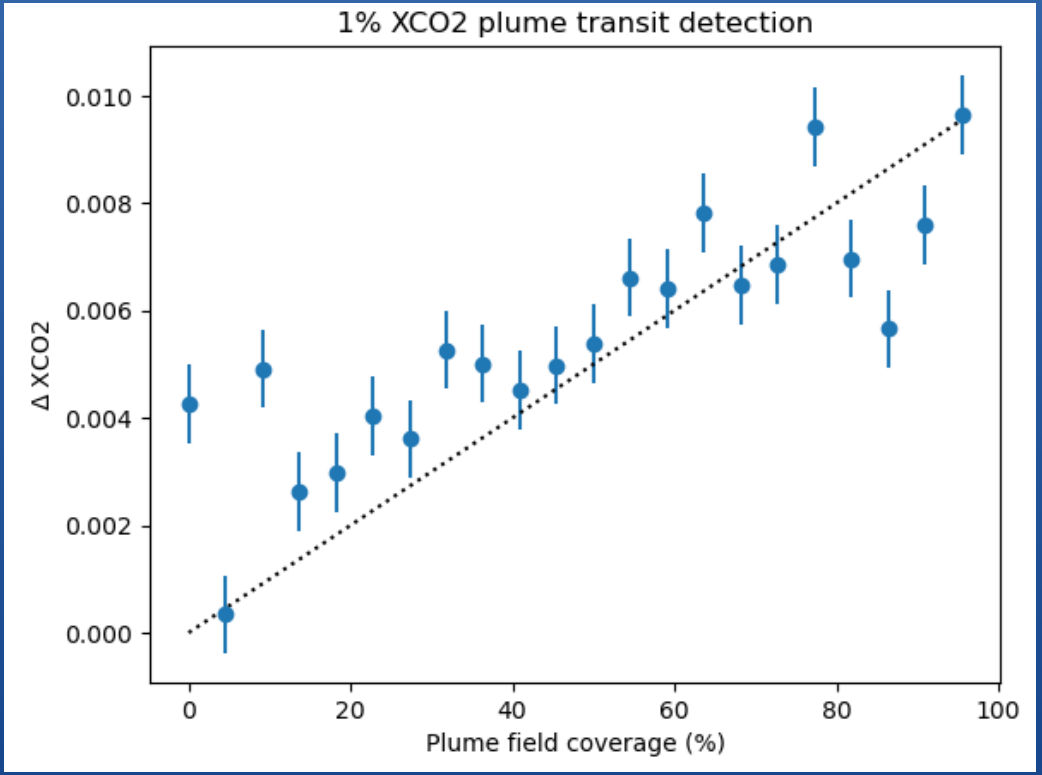




Plume Detection
SNR ~ 3-4

CO2 Plume Detection

- Smooth transition of the CO2 plume over the scenery

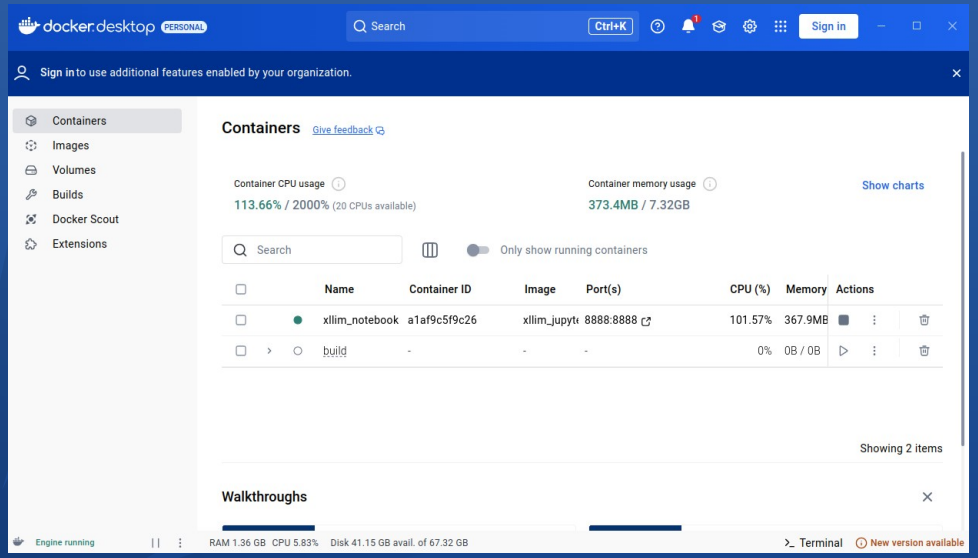


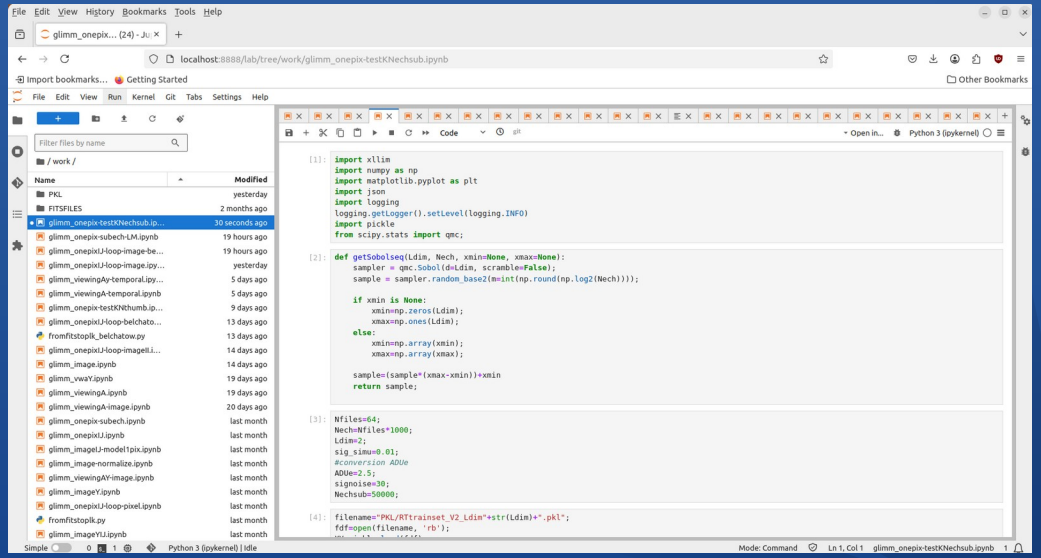
Conclusions

- GLiMM machine learning approach
  - Useful tool for **quick inversion** of complex (non linear, non analytic) models $Y=F(X) + \varepsilon$
  - Applied to hyperspectral observations of GHG:
    - Obtains similar estimations (and statistics) than LM iterative approach
    - Can handle a large flow of data in a reasonable amount of time
- Current Limitations - Perspectives
  - Requires a simulated training set → modeling errors?
  - Need of a unified model for images
  - Need to include aerosols

Using GLiMM

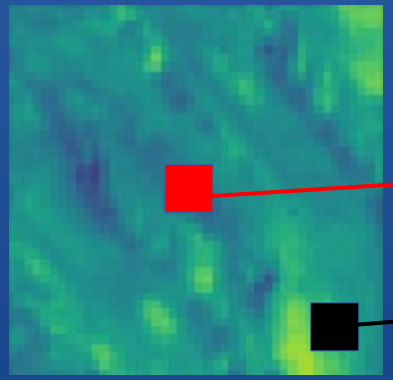
- Planet-GLLiM
  - https://gitlab.inria.fr/xllim/planet-gllim
    - C++ with Python bindings implementation of Gaussian Locally-Linear Mapping.
    - compiled shared library in a Docker container
    - Python environment in your browser

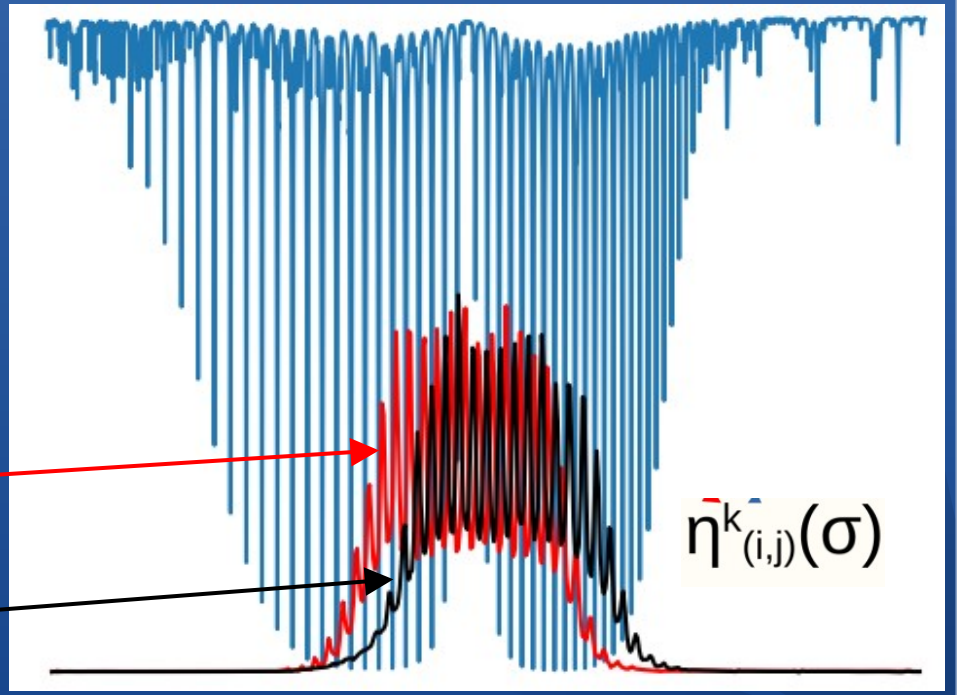




Appendix

- Effect of instrumental transfer function
  - Shift of shape and spectral bandwidth with viewing angle
  - Implicit in GLiMM training
  - Interpolation not efficient





Appendix

- Effect of instrumental transfer function
  - Shift of shape and spectral bandwidth with viewing angle
  - Implicit in GLiMM training
  - Interpolation not efficient