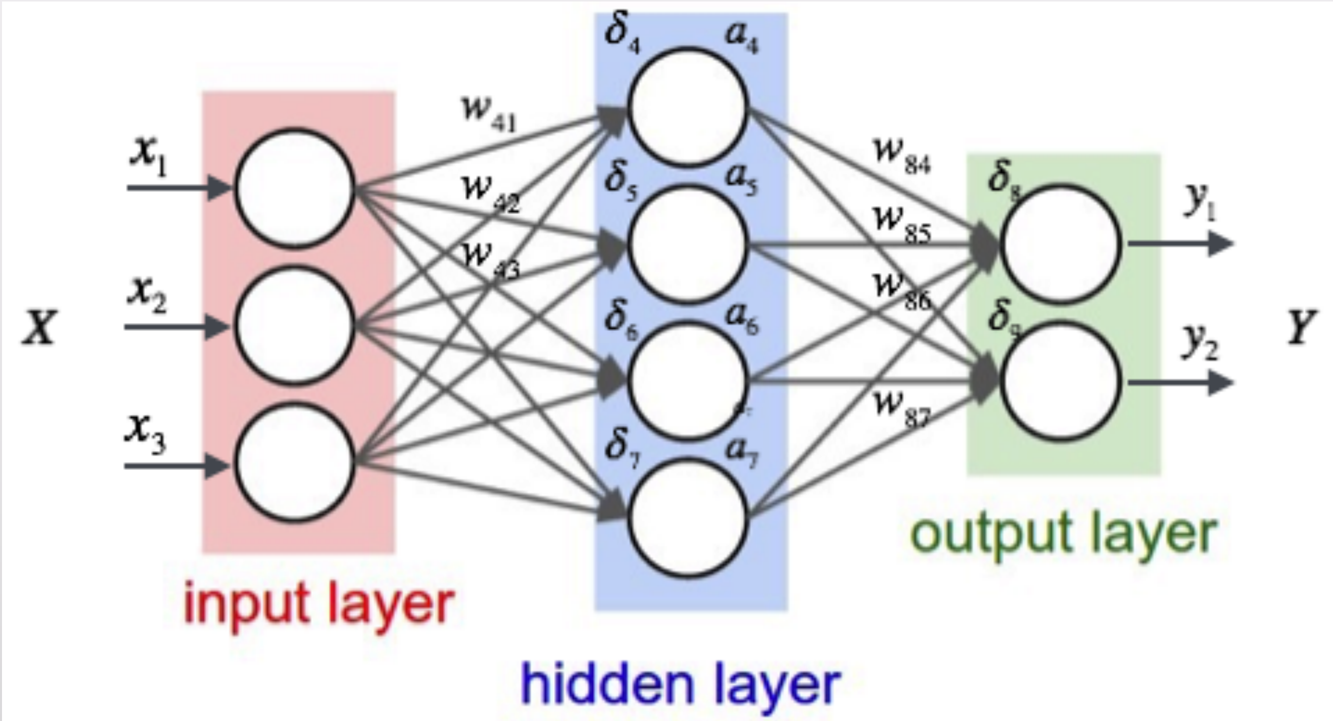
ExoGAN: Retrieving Exoplanetary Atmospheres

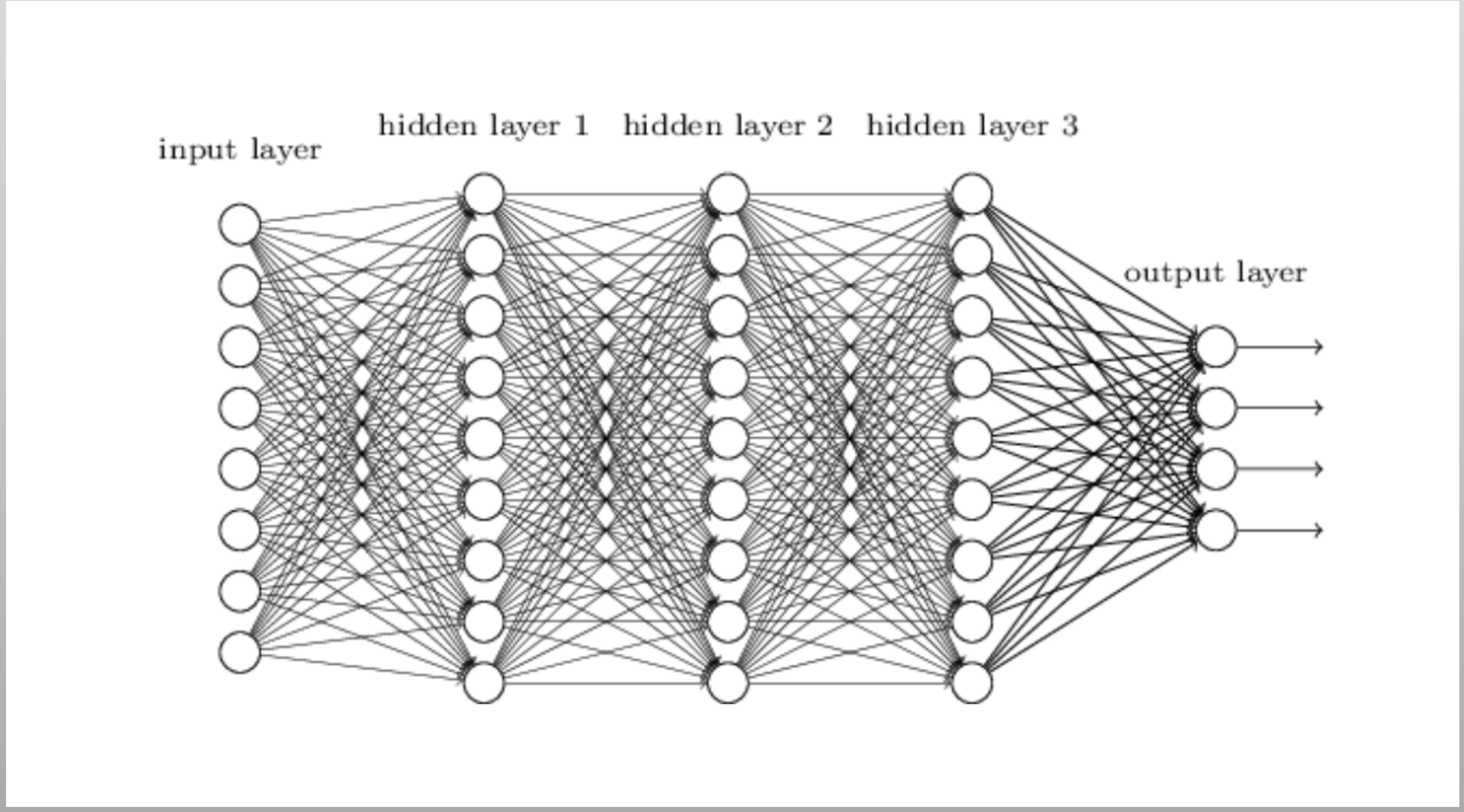
Using Deep Convolutional Generative Adversarial Networks

ARIEL meeting Roma – 03/10/2018

Tiziano Zingales

Neural networks structure





What is ExoGAN

- Generative Adversarial Network: are deep neural net architectures comprised of two nets, pitting one against the other (thus the “adversarial”).

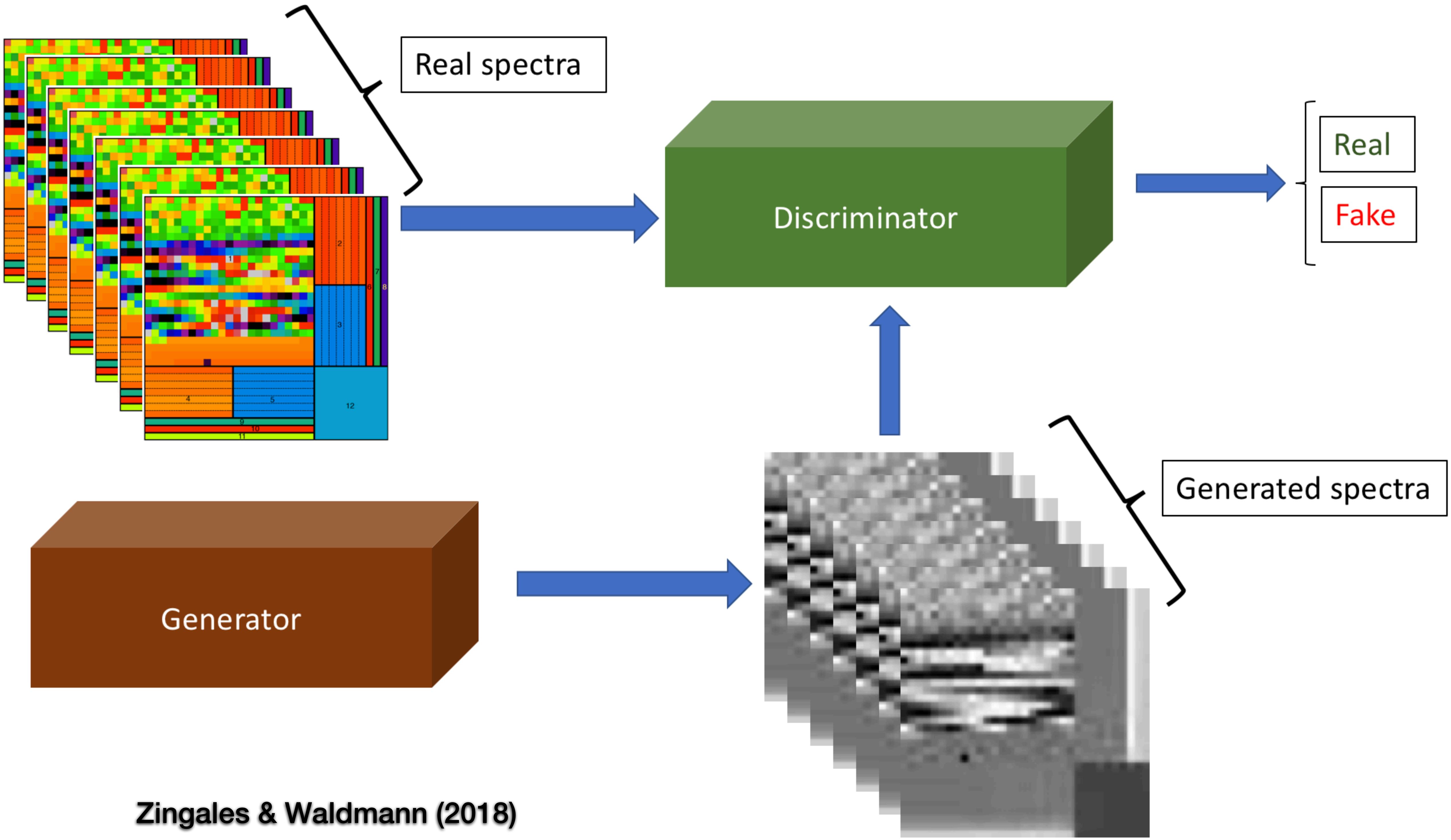
- It is made of two main neural nets (models):

  - ✴ Discriminative model: distinguish samples in disguise;

  - ✴ Generator model: Generate new samples that are as similar as the data.

```
# Original Version: Taehoon Kim (http://carpedm20.github.io)
#   + Source: https://github.com/carpedm20/DCGAN-tensorflow/blob/e30539fb5e20d5a0fed40935853da97e9e55eee8/model.py
#   + License: MIT
# [2016-08-05] Modifications for Completion: Brandon Amos (http://bamos.github.io)
#   + License: MIT
#
# [2017-11-02] Modifications for Exoplanetary science
# contributors: Tiziano Zingales (1, 2), Ingo Waldmann (1)
#   + License: (1) UCL, (2) INAF/OaPa
```

What is a GAN



Zingales & Waldmann (2018)

Police - counterfeit game

Absolutely no!

Police - counterfeit game

→

No!

Police - counterfeit game

Maybe, but no!

Police - counterfeit game

Yes!

Training the GAN

Binary cross-entropy of the Discriminator

$$V^{(D)} = -\left[\log D(x) + \log\left(1 - D\left(G(z)\right)\right)\right]$$

Where x is the input image, z is the latent variable and G(z) is the generated image

Usually we work with batches of data

$$V^{(D)} = -\left\{\sum_x \log D(x) + \sum_z \log\left(1 - D\left(G(z)\right)\right)\right\}$$

Training the GAN

- The Discriminator wants to minimise the entropy of the net;
- The Generator wants to fool the Discriminator, thus it wants to maximise the entropy of the net;
- It is natural defining the value function of the Generator as:

$$V^G = -V^D$$

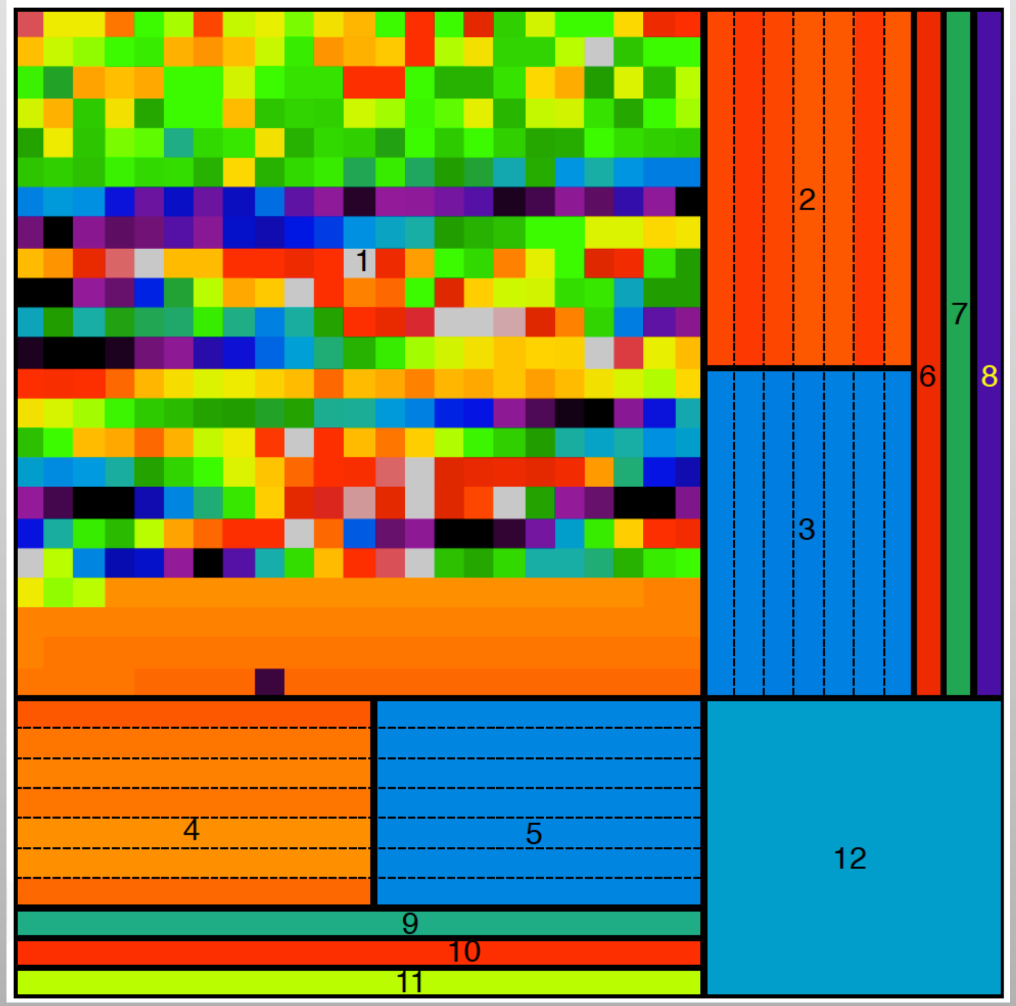
Some tests with human faces

750 iterations

2250 iterations

The training set

- 10 million exoplanetary atmospheres varying CO, CO_2, H_2O and CH_4 abundances, equilibrium temperature, radius and mass of the planet.

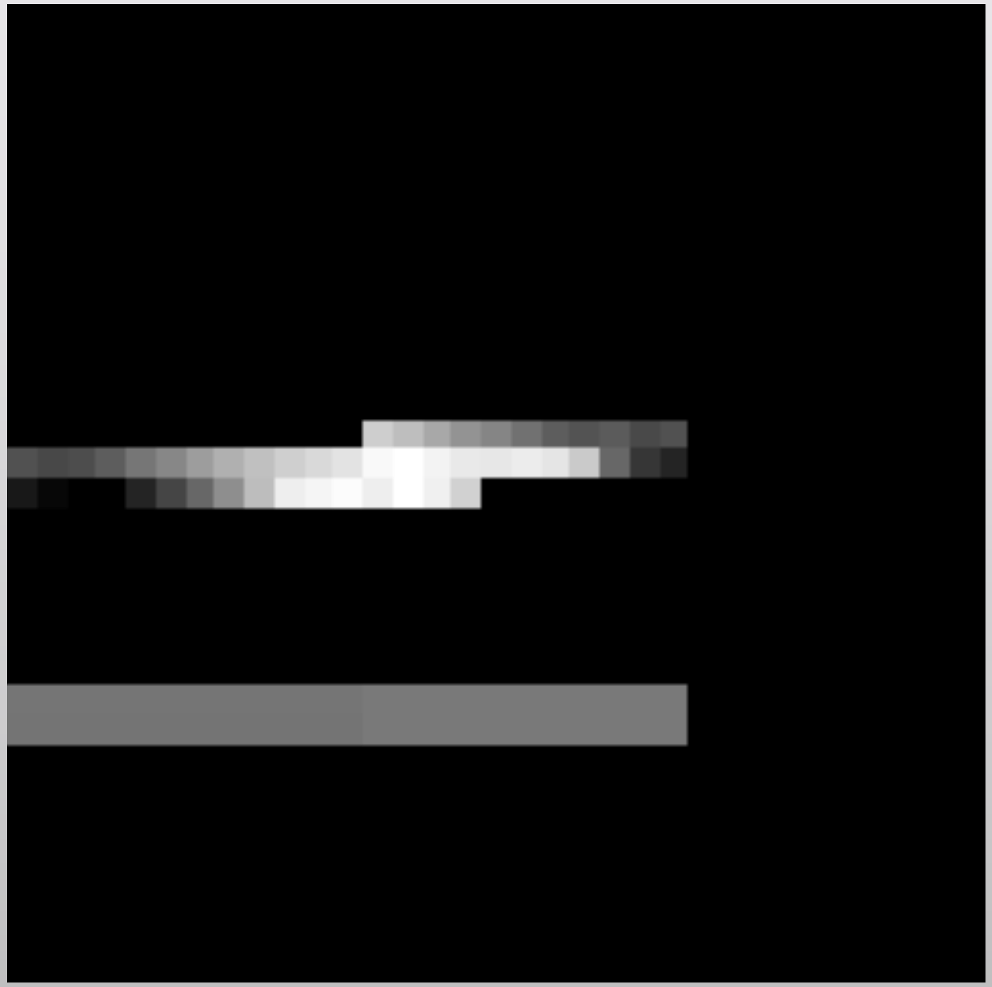


Reconstruction

- Input spectra do not show the parameters value, but a GAN can reconstruct it.
- From the available information it is possible to reconstruct the missing one (inpainting).

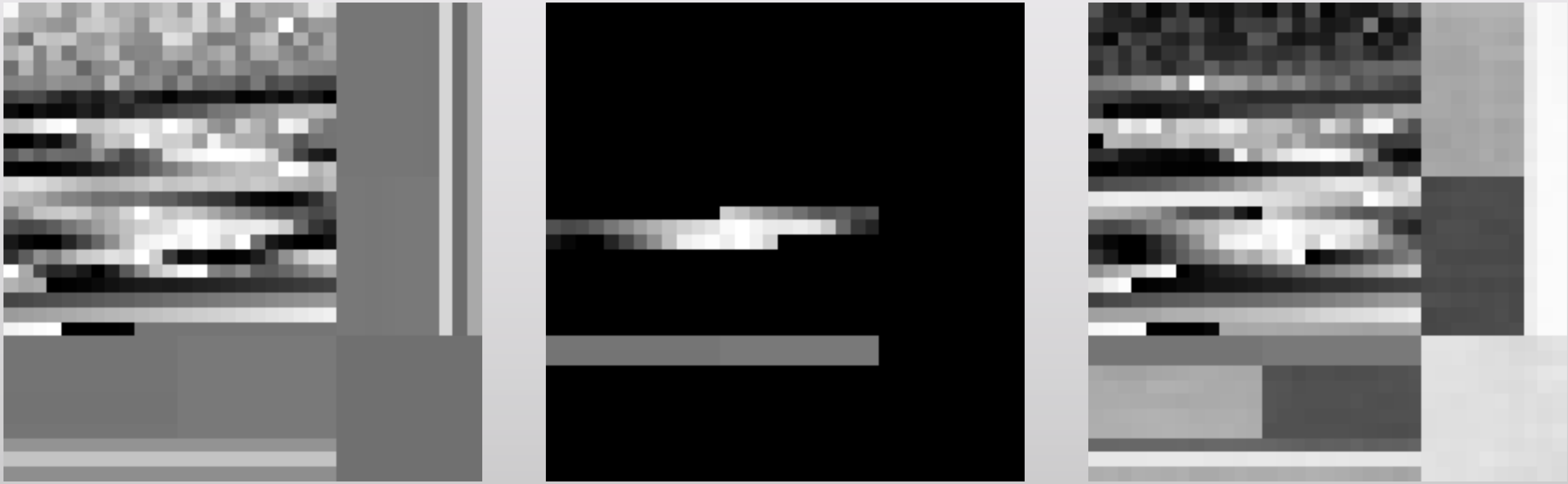
Reconstruction

With M the mask (or the missing data) and y the input (incomplete) image



Brandon Amos (http://bamos.github.io)
Completion Algorithm

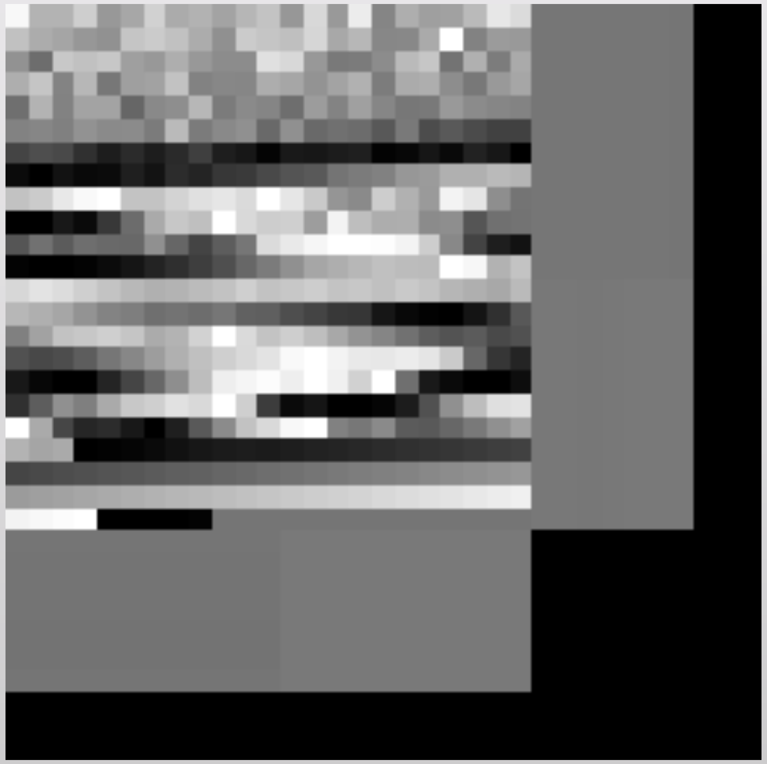
Reconstruction



Input WFC3 mask Reconstruction

Reconstruction

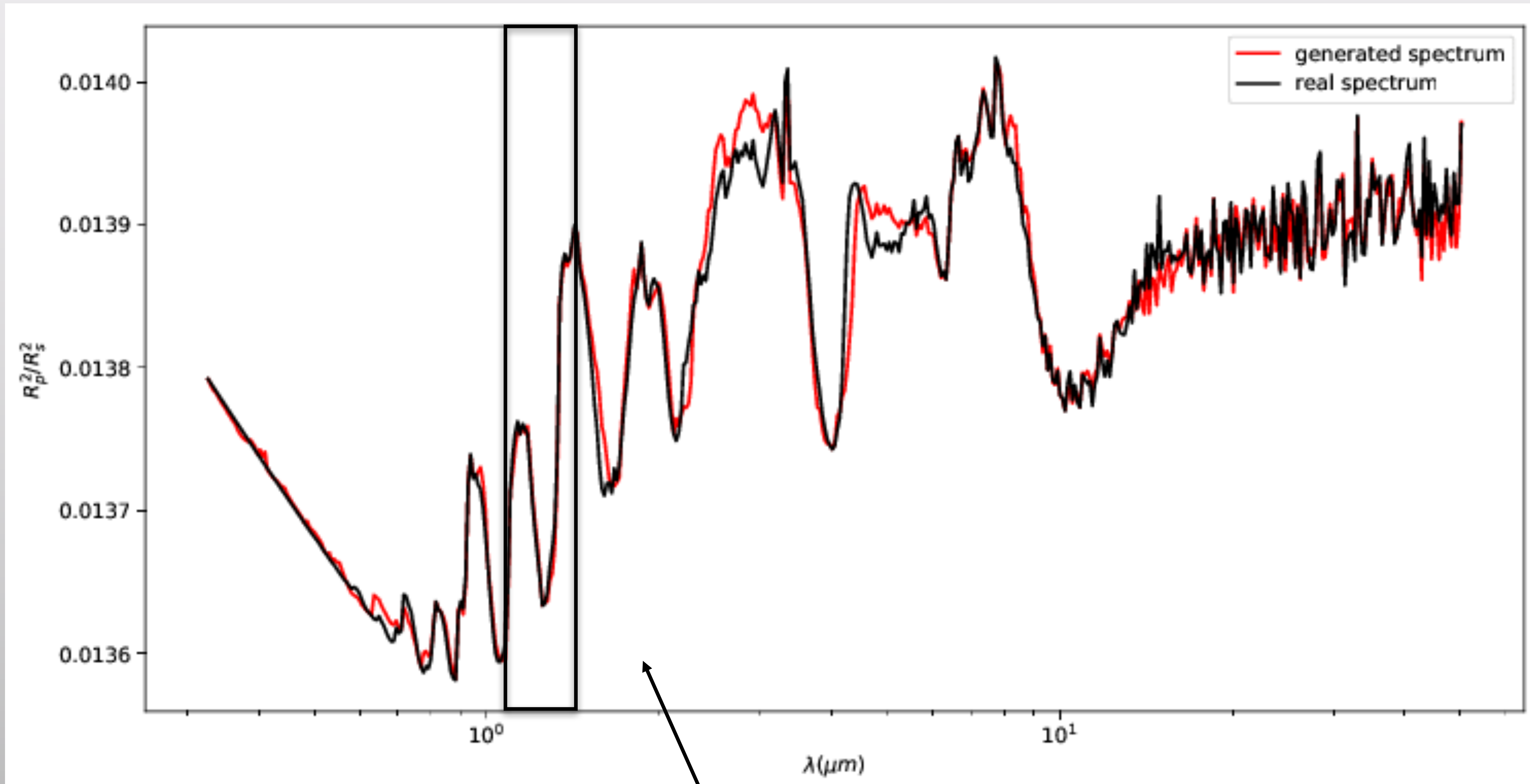
Input



Parameters mask

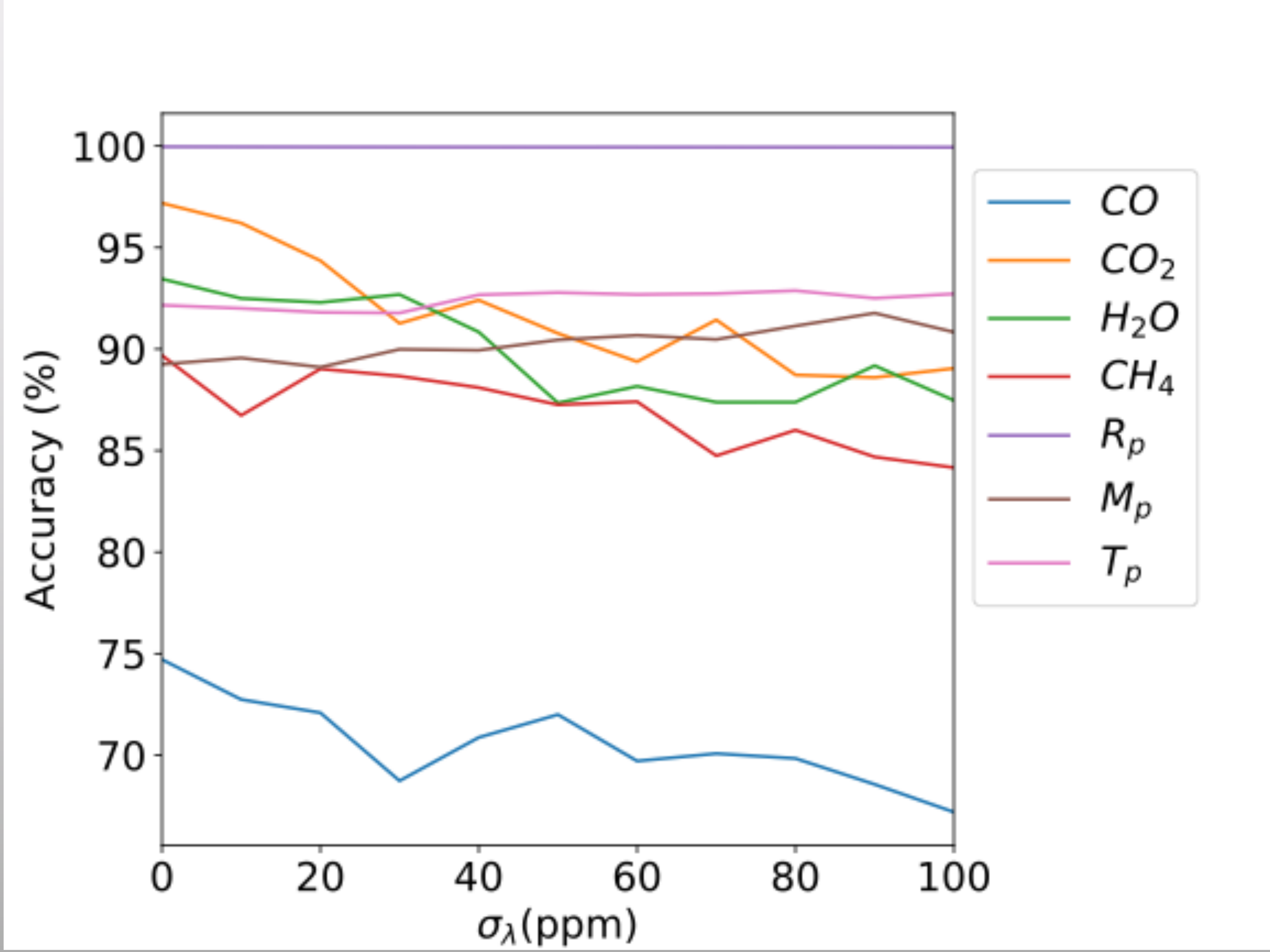
Reconstruction

Spectral reconstruction

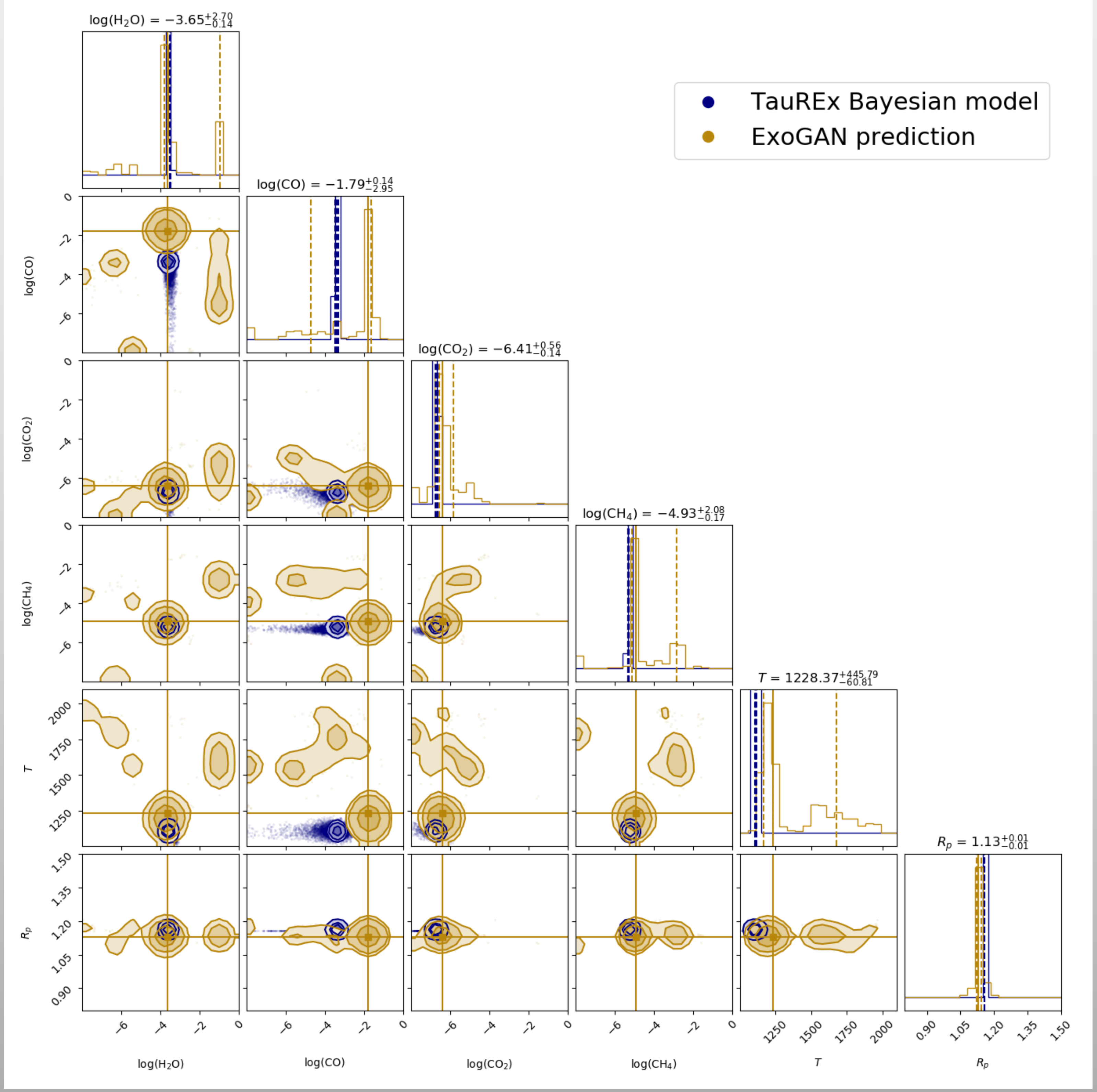


HST/WFC3

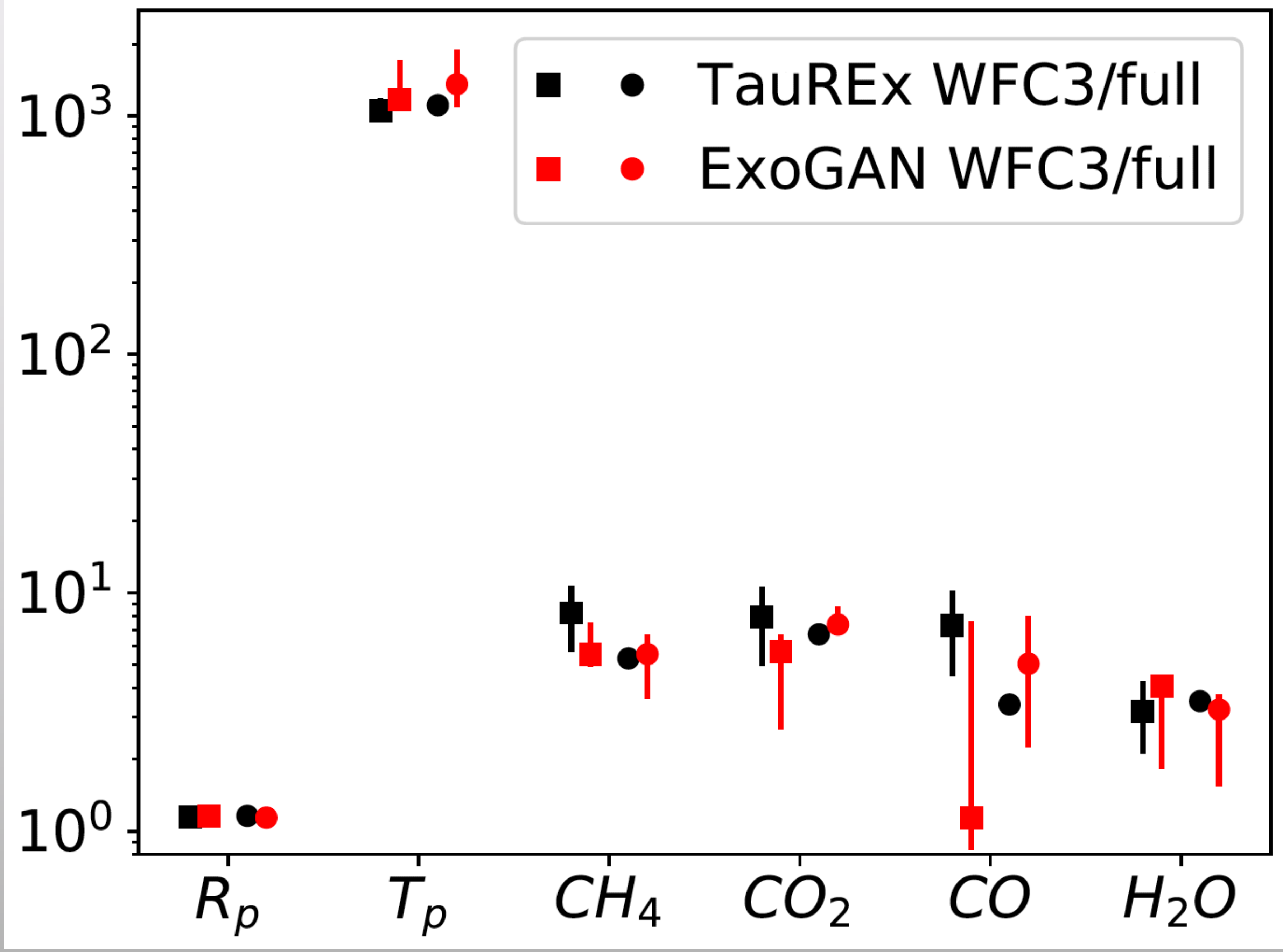
Accuracy and tests



Accuracy and tests



Accuracy and tests



Conclusion

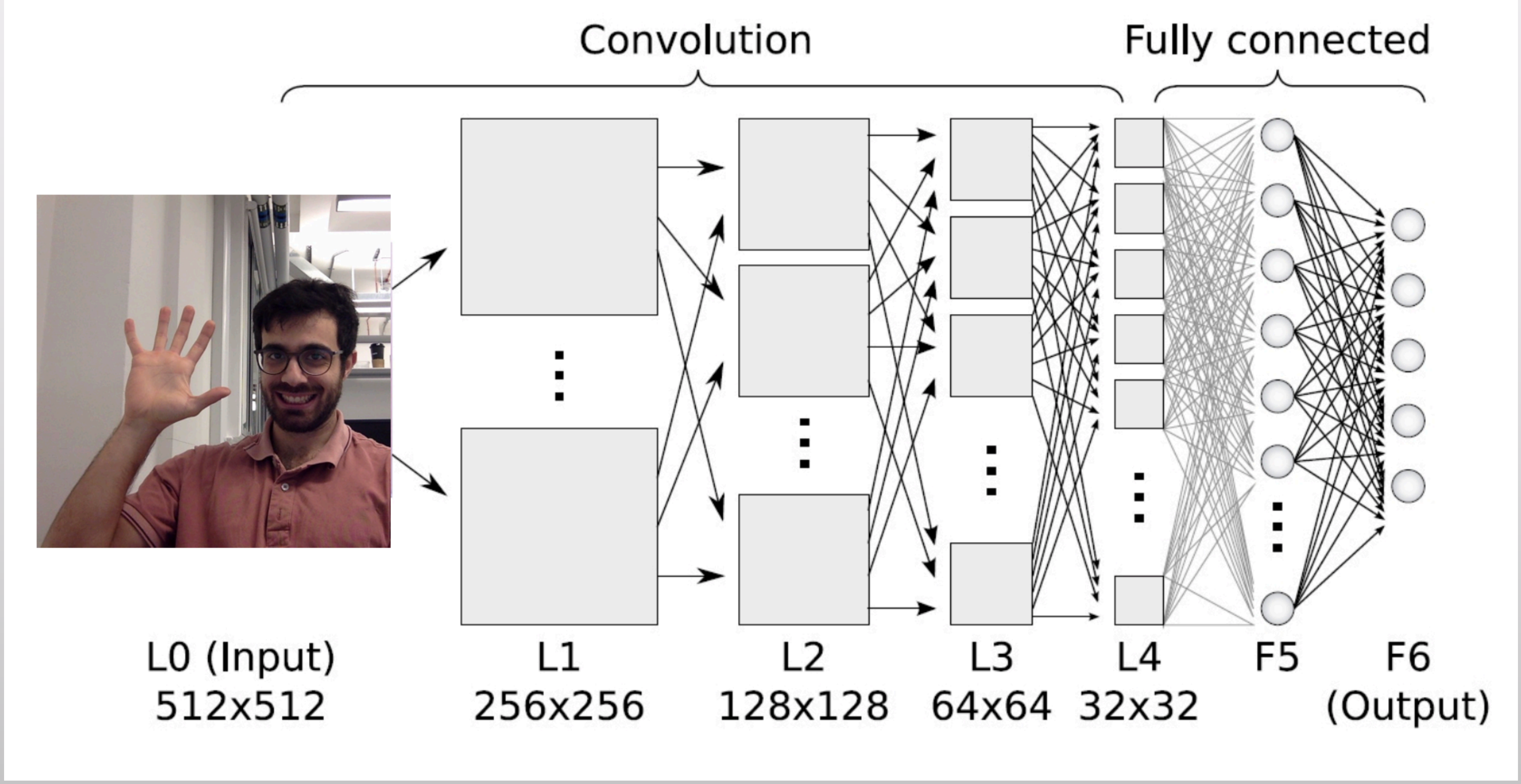
- Neural networks can help us constraining the parameter space in a smarter way;
- GANs allow us to speed up a spectral retrieval by using informative priors;
- Algorithms optimised with AI can improve performances and precision of atmospheric retrievals.

Why using a neural net for astrophysical purposes?

- Because of the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies;

- Speed up the computing time required by many codes in astrophysics.

Convolutional Neural Networks



Training the GAN

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\theta_{d,k} = \theta_{d,k-1} - \eta \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\theta_{g,k} = \theta_{g,k-1} - \eta \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

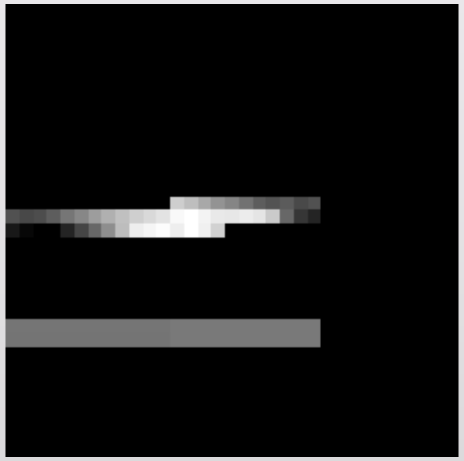
end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Goodfellow et al. (2014)

Reconstruction

With M the mask (or the missing data) and y the input (incomplete) image



$$\mathcal{L}_{contextual}(z) = \| M \odot G(z) - M \odot y \|_1$$

$$\mathcal{L}_{perceptual}(z) = \log\left(1 - D(G(z))\right)$$

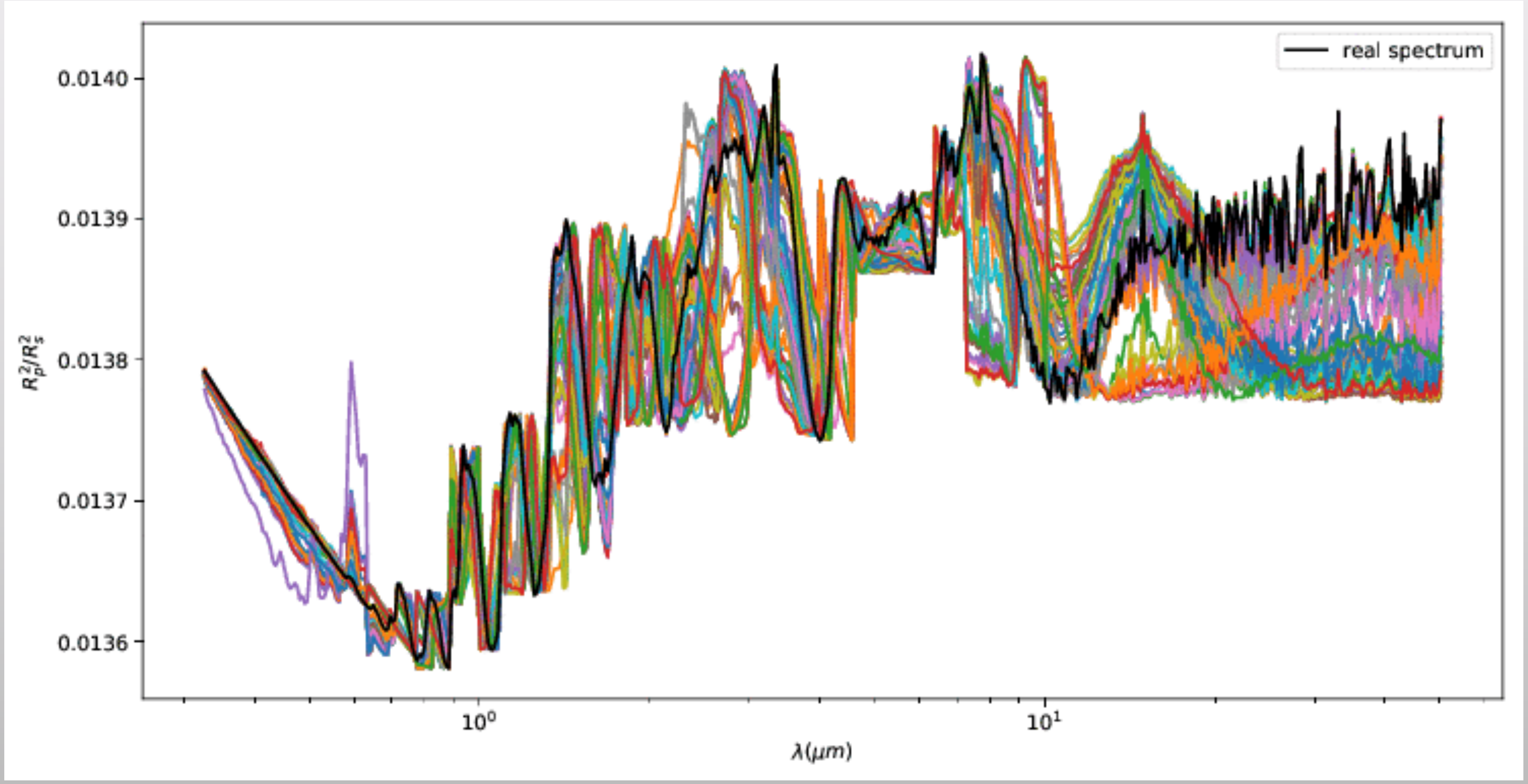
$$\mathcal{L} = \mathcal{L}_{contextual}(z) + \lambda \mathcal{L}_{perceptual}(z)$$

$$\hat{z} = \arg\min_{z} \mathcal{L}(z)$$

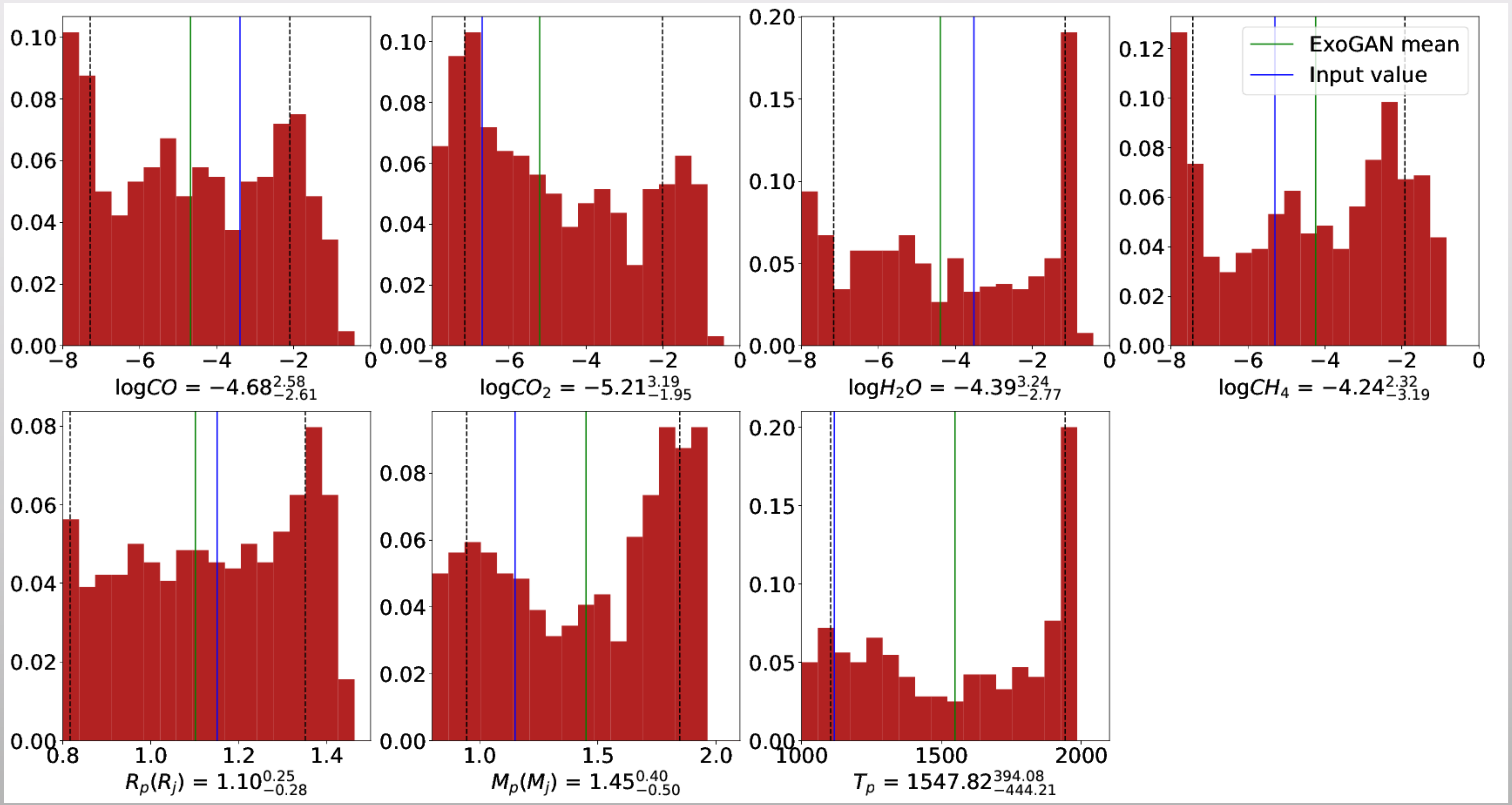
$$x_{reconstructed} = M \odot y + (1 - M) \odot G(\hat{z})$$

Facial reconstruction

Spectral reconstruction



The results



ExoGAN mean
Input value
$\log CO = -4.68^{2.58}_{-2.61}$
$\log CO_2 = -5.21^{3.19}_{-1.95}$
$\log H_2O = -4.39^{3.24}_{-2.77}$
$\log CH_4 = -4.24^{2.32}_{-3.19}$
$R_p(R_j) = 1.10^{0.25}_{-0.28}$
$M_p(M_j) = 1.45^{0.40}_{-0.50}$
$T_p = 1547.82^{394.08}_{-444.21}$

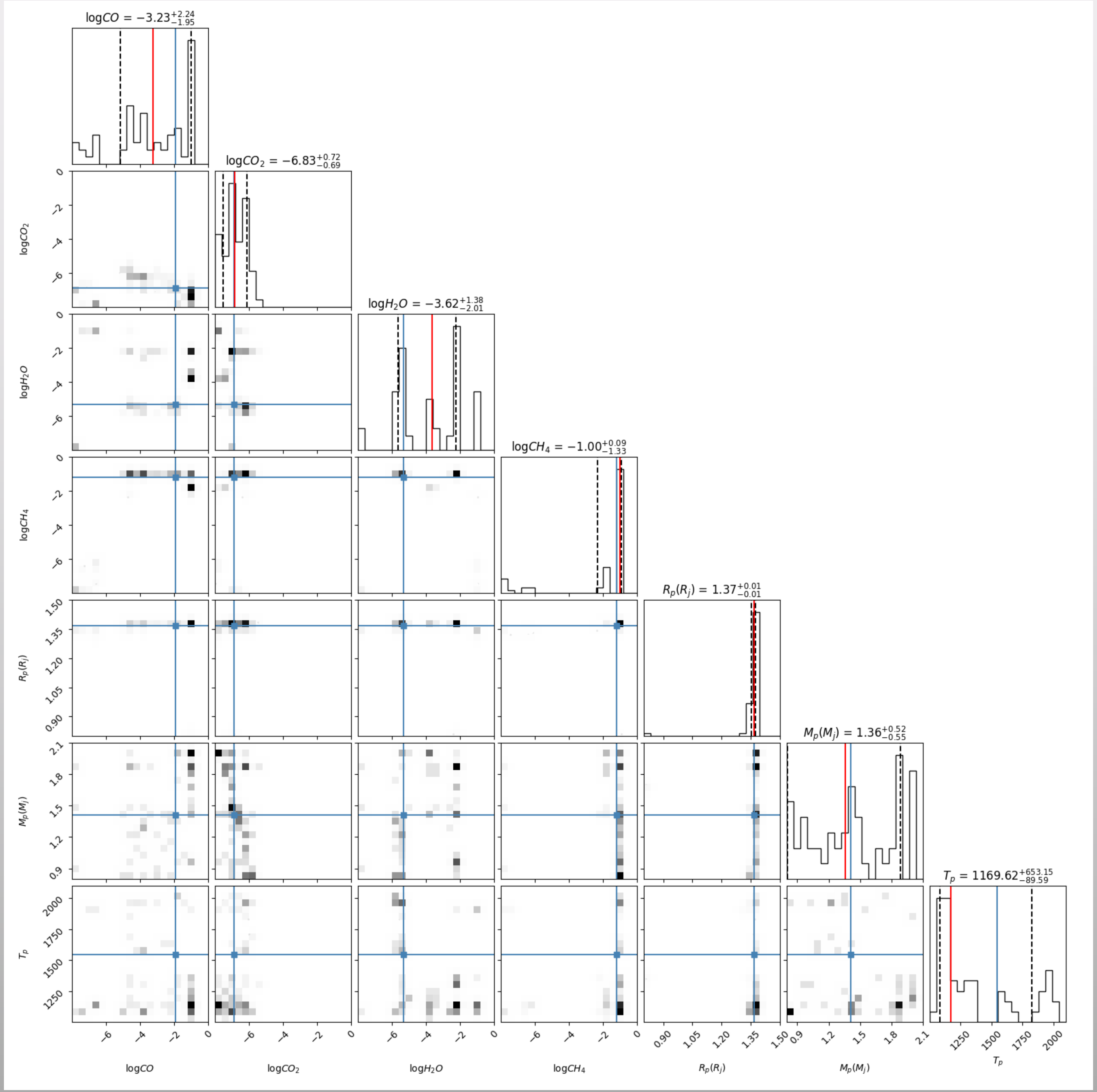
Accuracy and tests

$$A(\phi, \sigma_\phi) = \frac{1}{N} \sum_{i}^{N} \frac{(\phi_{i,recon} - \phi_i)^2}{{\phi_i}^2 + \sigma_{\phi_i}^2}$$

Training set parameters			
Variable	$A(0\,\sigma_\phi)$	$A(1\,\sigma_\phi)$	$A(2\,\sigma_\phi)$
CO	64.4%	74.9%	80.8%
CO_2	93.7%	96.4%	97.3%
H_2O	86.3%	92.9%	94.8%
CH_4	80.3%	88.4%	91.9%
R_p	99.8%	99.8%	99.8%
M_p	88.8%	90.5%	91.6%
T_p	89.4%	91.9%	93.1%

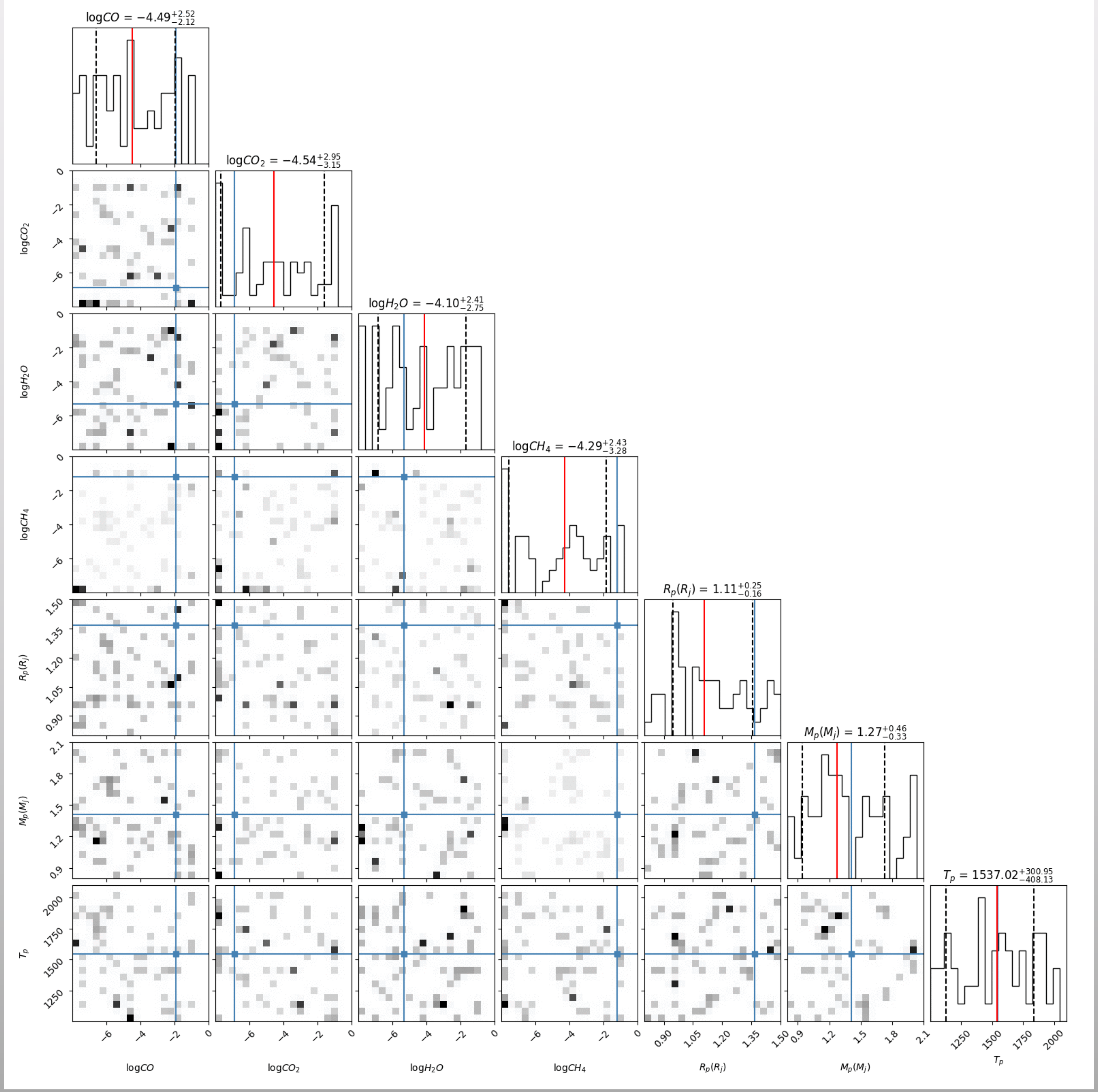
Test set parameters			
Variable	$A(0\,\sigma_\phi)$	$A(1\,\sigma_\phi)$	$A(2\,\sigma_\phi)$
CO	62.8%	72.6%	78.2%
CO_2	94.2%	96.6%	97.4%
H_2O	89.6%	92.8%	93.9%
CH_4	80.3%	88.2%	91.6%
R_p	100.0%	100.0%	100.0%
M_p	88.0%	89.7%	90.8%
T_p	90.4%	92.2%	93.2%

The results



$\log CO = -3.23^{+2.24}_{-1.95}$
$\log CO_2 = -6.83^{+0.72}_{-0.69}$
$\log H_2O = -3.62^{+1.38}_{-2.01}$
$\log CH_4 = -1.00^{+0.09}_{-1.33}$
$R_p(R_j) = 1.37^{+0.01}_{-0.01}$
$M_p(M_j) = 1.36^{+0.52}_{-0.55}$
$T_p = 1169.62^{+653.15}_{-89.59}$
$\log CO$
$\log CO_2$
$\log H_2O$
$\log CH_4$
$R_p(R_j)$
$M_p(M_j)$
T_p

The results



$\log CO = -4.49^{+2.52}_{-2.12}$
$\log CO_2 = -4.54^{+2.95}_{-3.15}$
$\log H_2O = -4.10^{+2.41}_{-2.75}$
$\log CH_4 = -4.29^{+2.43}_{-3.28}$
$R_p(R_j) = 1.11^{+0.25}_{-0.16}$
$M_p(M_j) = 1.27^{+0.46}_{-0.33}$
$T_p = 1537.02^{+300.95}_{-408.13}$
$\log CO$
$\log CO_2$
$\log H_2O$
$\log CH_4$
$R_p(R_j)$
$M_p(M_j)$
T_p