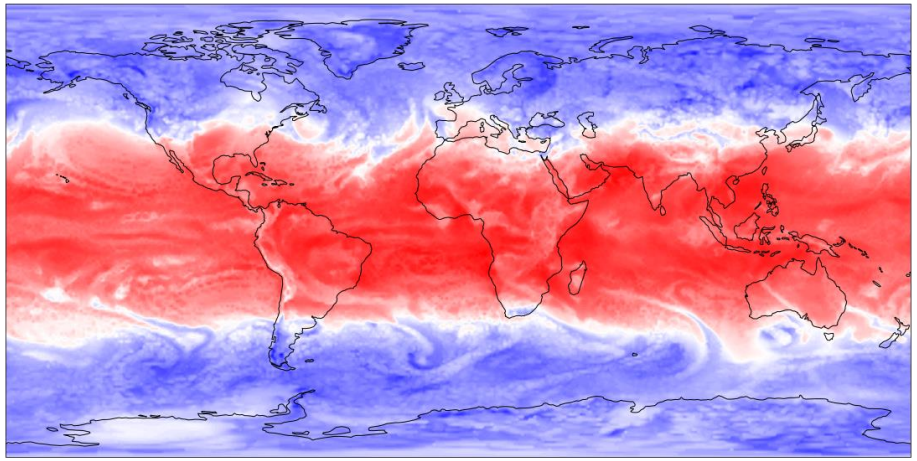
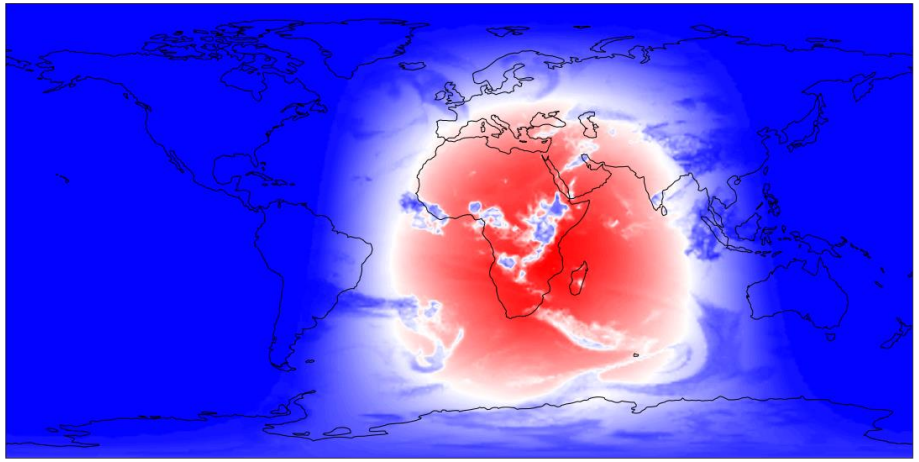
ETH zürich IAC Institute for Atmospheric and Climate Science SDSC

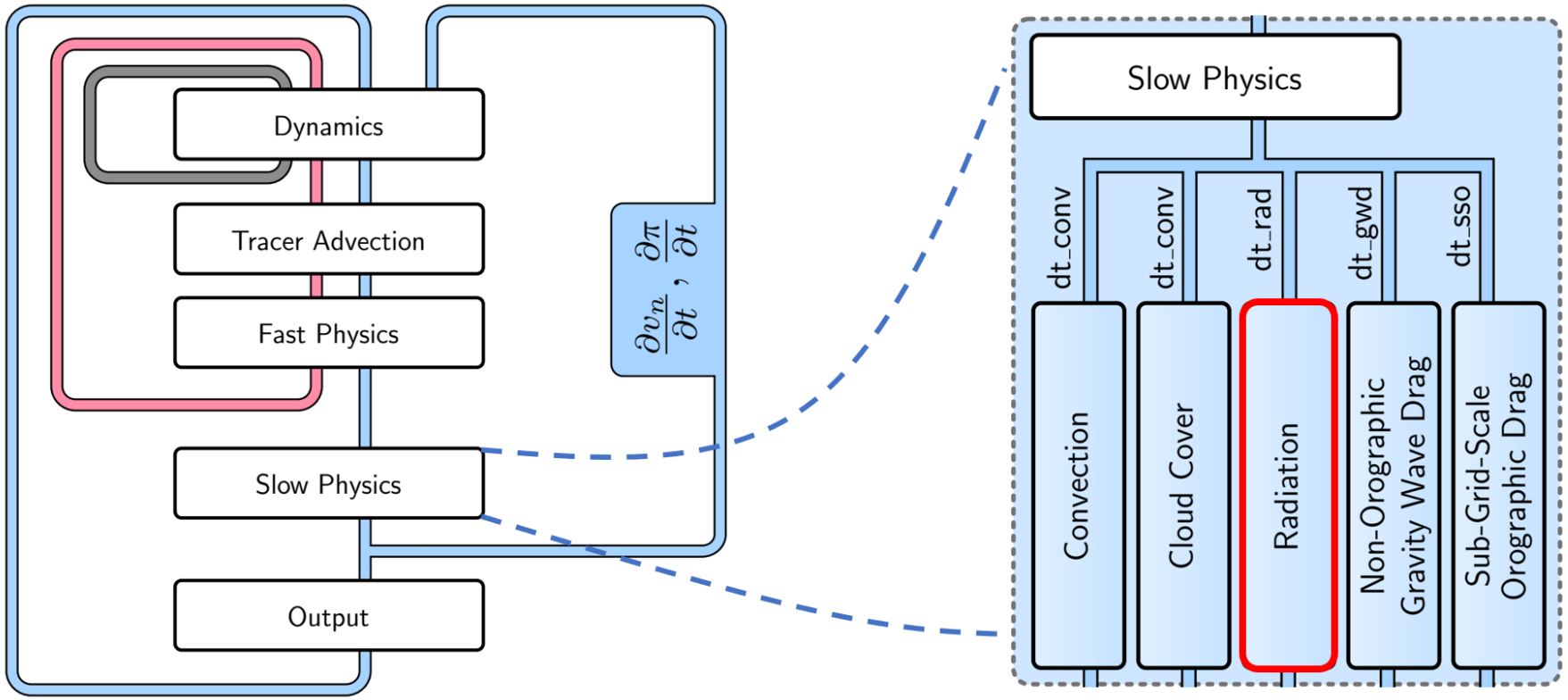
Building a physics-constrained, fast and stable machine learning-based radiation emulator

G. Bertoli, F. Ozdemir, F. Perez Cruz, S. Schemm, E. Szekely





ICON : Icosahedral Nonhydrostatic Weather and Climate Model



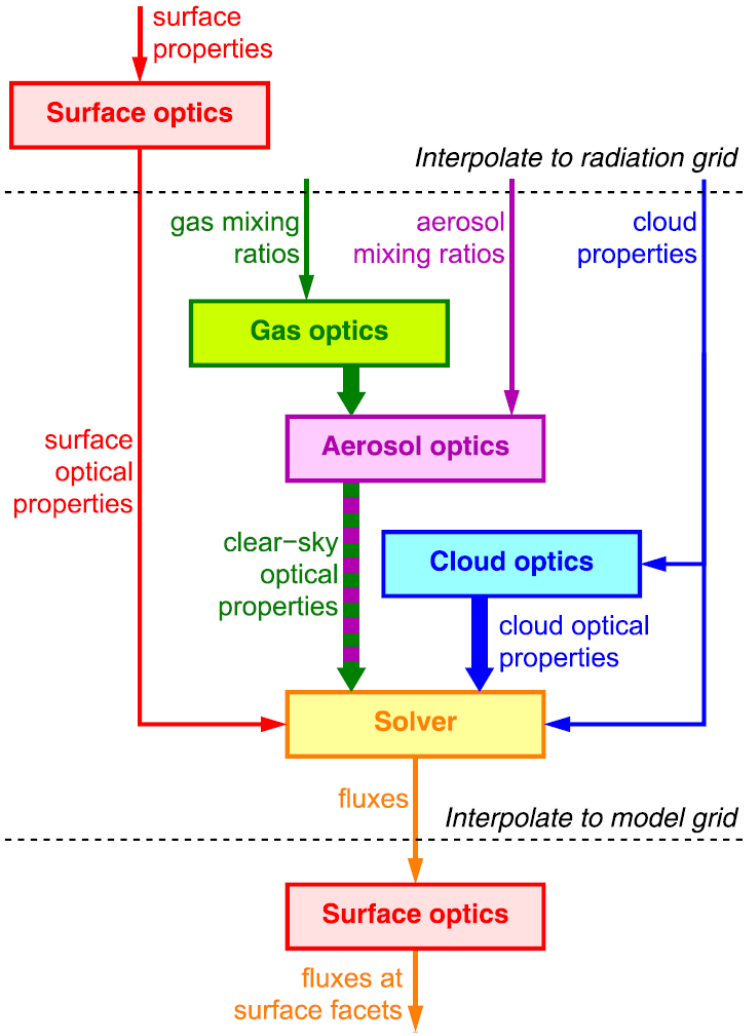
ICON grid



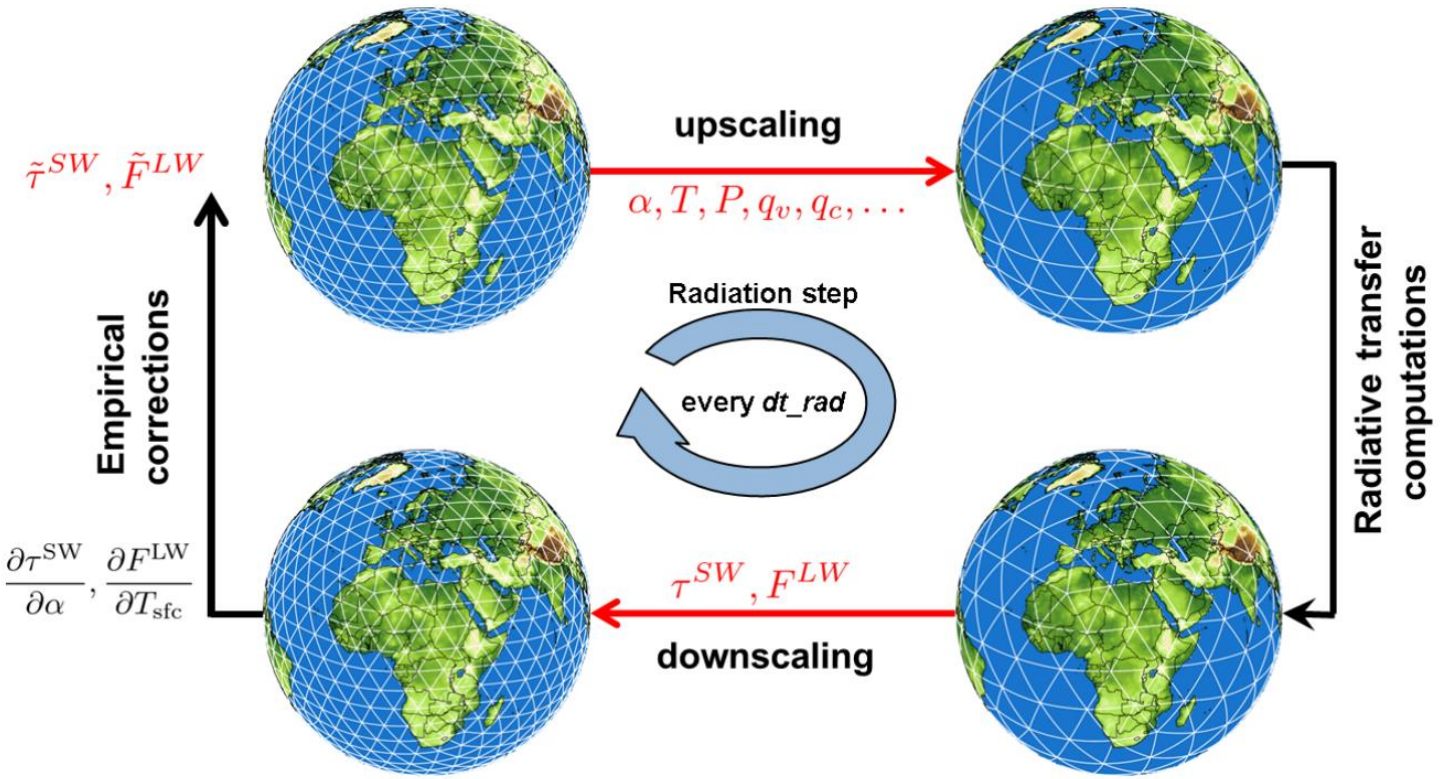
F. Prill, D. Reinert, D.Rieger, G. Zängl, Working with the ICON Model, 2020

EcRad

ecRad architecture:



Coarser grid used for ecRad in ICON:



- τ^{SW} : shortwave transmissivity
- F^{LW} : longwave flux
- α : albedo
- T_{sfc} : surface temperature

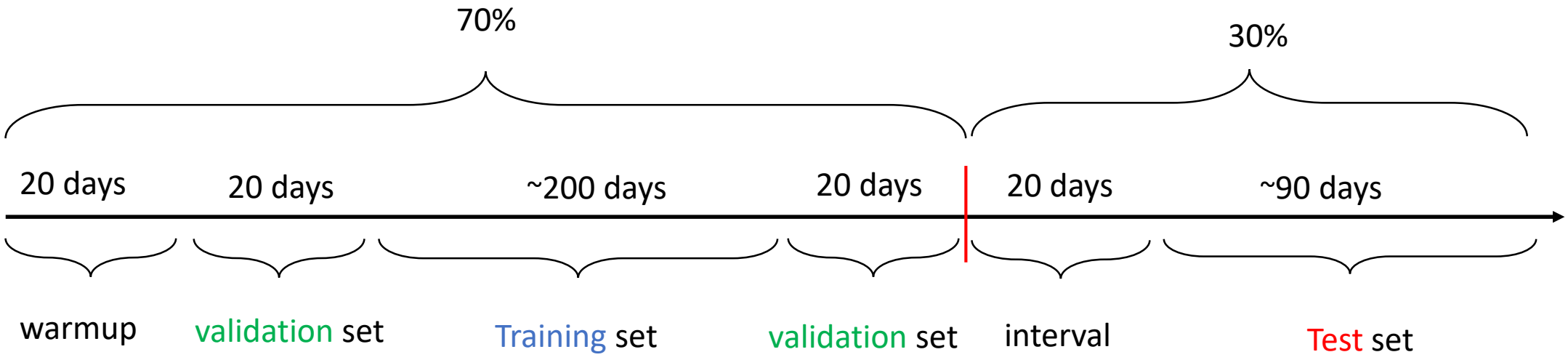
R. J. Hogan, A. Bozzo, A Flexible and Efficient Radiation Scheme for the ECMWF Model, 2018, Journal of Advances in Modeling Earth Systems

F. Prill, D. Reinert, D.Rieger, G. Zängl, Working with the ICON Model, 2020

Training set, validation set and test set

Data : 12 months simulation stored every three hours.

Data set :



Outputs:

1) Shortwave down
2) Shortwave up
3) Longwave down
4) Longwave up

Inputs:

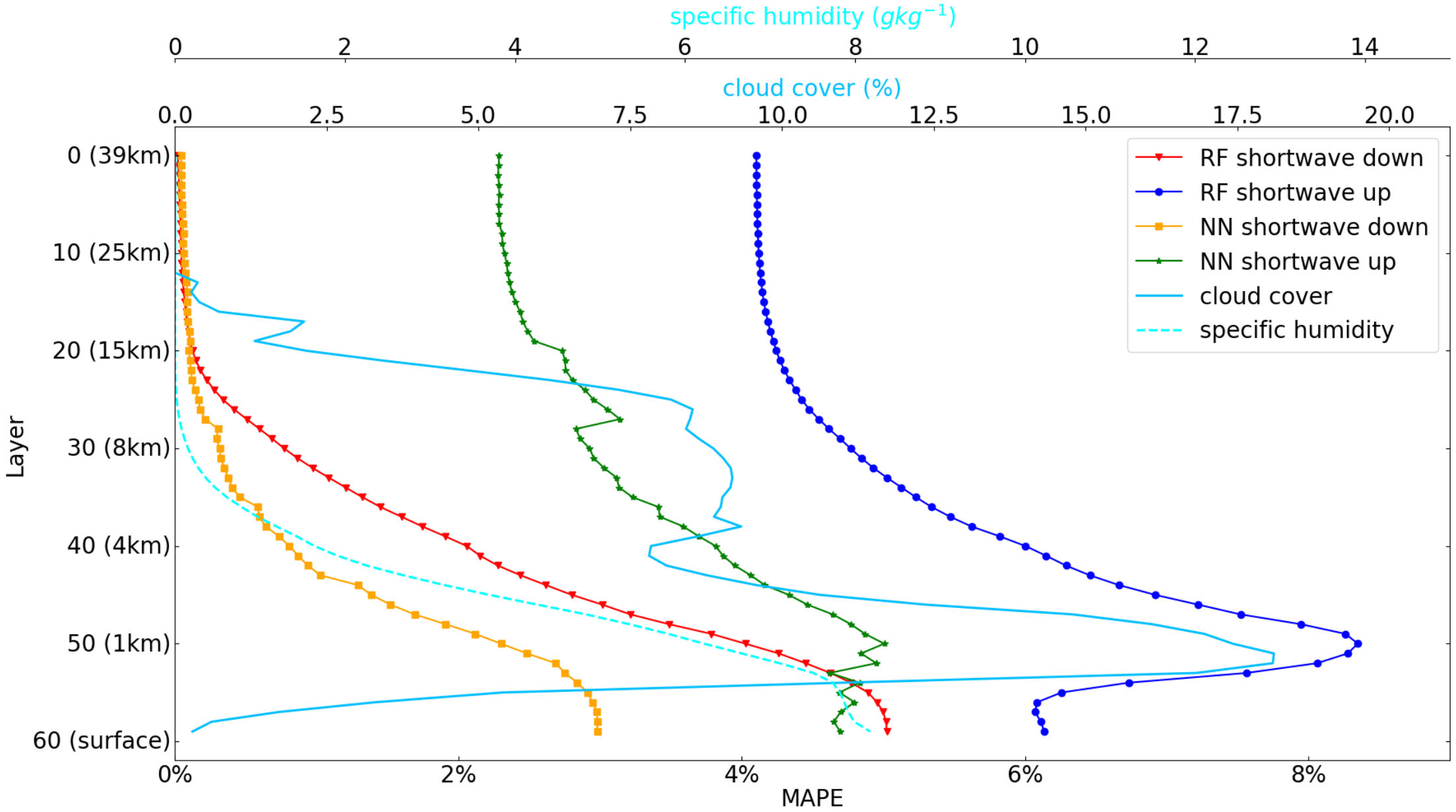
- Temperature
- Pressure
- Cloud cover
- Specific humidity
- Specific cloud water content
- Specific cloud ice content
- Cosine of solar zenith angle
- Albedos

Two strategies for the machine learning emulation

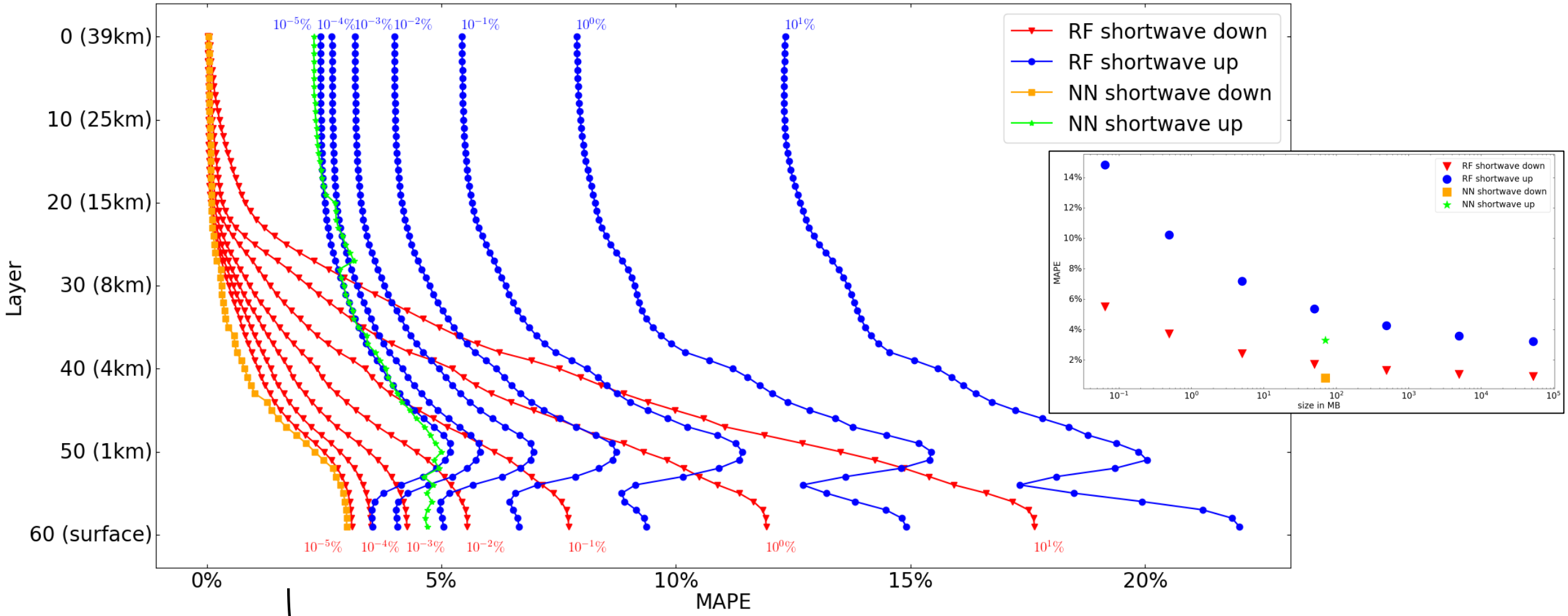
a) Random Forest

b) Neural Network

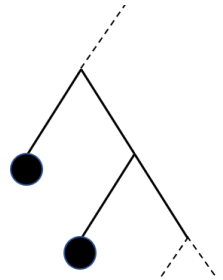
Comparison of the two methods



MAPE and forest size

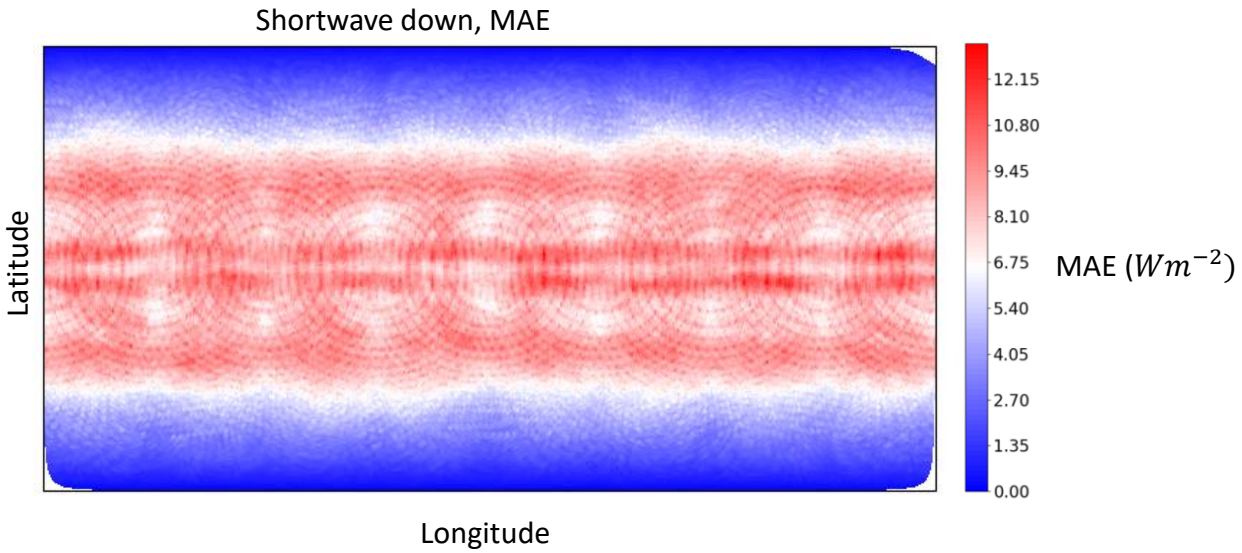


Minimum percentage of the training set in the leaf nodes.

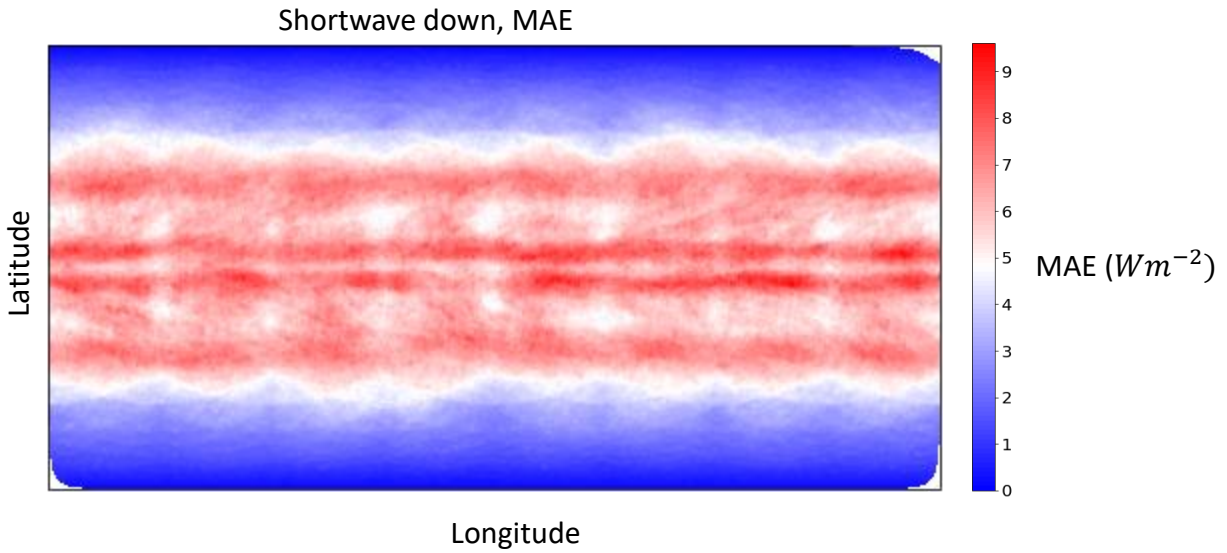


Removing the sun

We predict directly the fluxes $F_{sw}^{\uparrow}$, $F_{sw}^{\downarrow}$.



1. We divide by $\cos(\theta)$.
2. We predict $\frac{F_{sw}^{\uparrow}}{\cos(\theta)}, \frac{F_{sw}^{\downarrow}}{\cos(\theta)}$.
3. We multiply by $\cos(\theta)$ to recover $F_{sw}^{\uparrow}$, $F_{sw}^{\downarrow}$.



Remark : When $\cos(\theta) < 10^{-4}$, we choose $F_{sw}^{\uparrow} = 0$, $F_{sw}^{\downarrow} = 0$.

25% error reduction for free!

Ongoing work: increasing the model stability

1) Energy conservation with a custom loss function
2) Convolutional neural network
3) Low resolution additional inputs

1) Neural network: physics informed loss function

Energy balance for an atmospheric column:

$$\frac{1}{g}\frac{\partial}{\partial t}\int_0^{p_s}\left(c_pT+\Phi_s+k\right)dp+\frac{1}{g}\nabla_p\cdot\int_0^{p_s}\mathbf{U}\left(c_pT+\Phi+k\right)dp\ =(R_T-R_s)+LP-F_{sh}$$

Φ : geopotential
k : kinetic energy
c_p : specific heat of air

U : horizontal wind
R : net irradiance
L : latent heat of vaporization

P : precipitation rate
L : latent heat of vaporization

Loss function: $\mathcal{L} = \mathcal{L}_{\text{data}} + \alpha\,\mathcal{L}_{\text{energy}}$

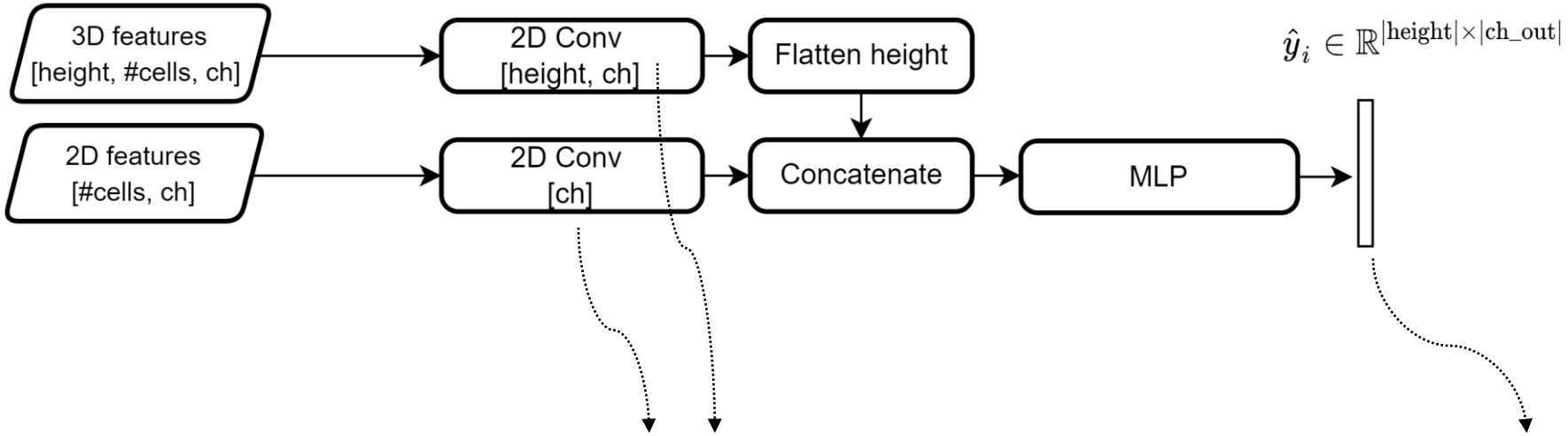
$$\mathcal{L}_{\text{data}} = \frac{1}{N}\sum_{n=1}^{N}\left\|y_{ML,n}-y_{ecRad,n}\right\|^2$$

$\boldsymbol{N}$ = # training points

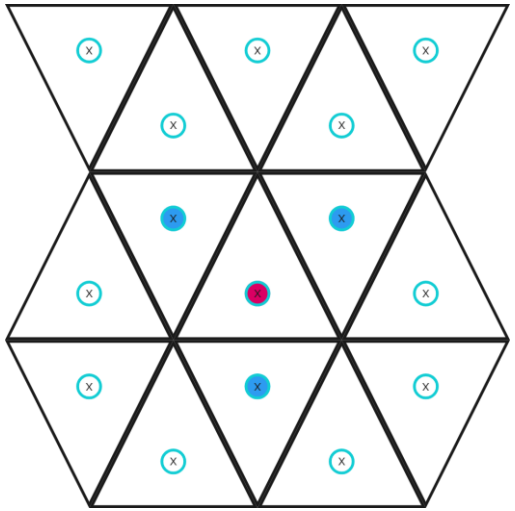
$$\mathcal{L}_{\text{energy}} = \frac{1}{N}\sum_{n=1}^{N}\left((R_{T,n}-R_{S,n})\big|_{ML}-(R_{T,n}-R_{S,n})\big|_{EcRad}\right)^2$$

2) Convolutional neural network

Architecture:



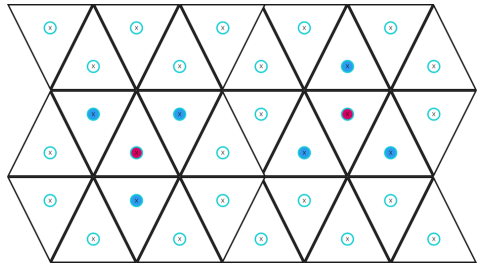
Convolution patch



$\mathcal{L} = \mathcal{L}_{\text{data}} + \alpha \mathcal{L}_{\text{energy}}$

Remark:

All three neighboring cells share the same weight.



3) Hybrid approach

1) Use ecRad to compute fluxes at low resolution.

2) Use the low resolution fluxes as additional inputs:

3) Compute the high resolution fluxes.

Inputs of the hybrid approach

Low resolution ICON grid:

Shortwave down
Shortwave up



High resolution ICON grid:

Cloud cover, Specific humidity, pressure, temperature, etc.



Conclusion

Ongoing work:

- Port the python neural network and random forest into Fortran with the Infero software from ECMWF
- Plug the machine learning emulator into ICON.

Next steps

- Online numerical experiments.

email : guillaume.bertoli@env.ethz.ch