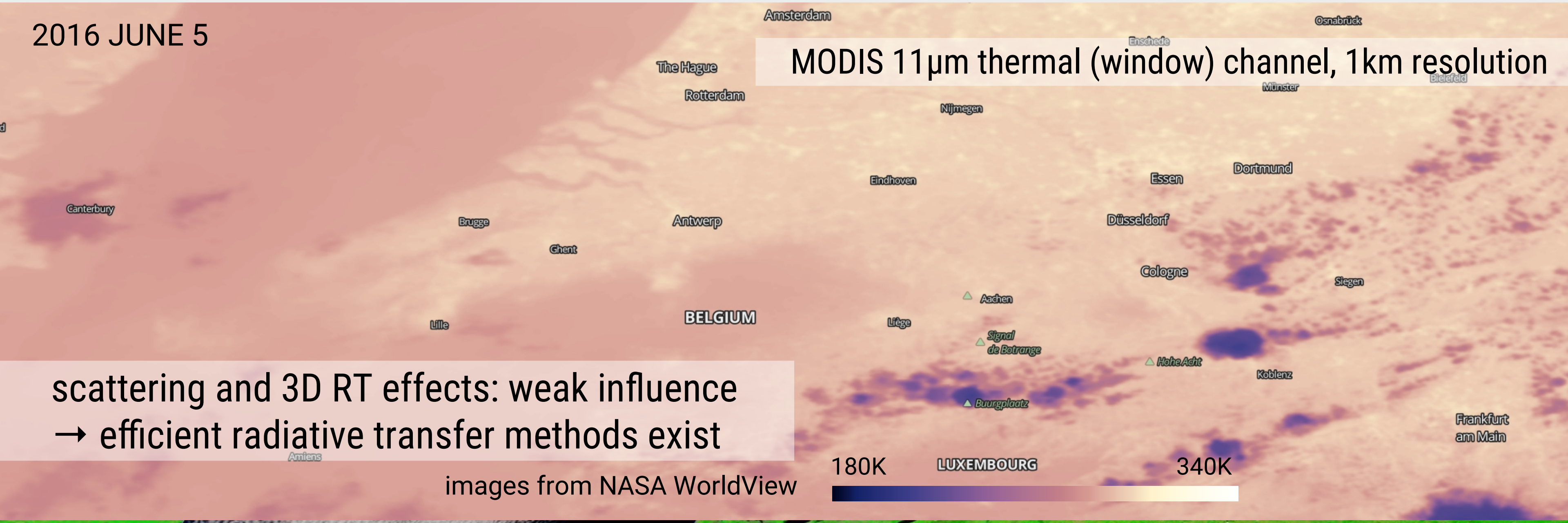
Fast methods for simulating visible satellite images

Leonhard Scheck[1], Florian Baur[2], Stefan Geiss[1], Martin Weissmann[3], Bernhard Mayer[4], Olaf Stiller[2], Christina Stumpf[2], Lilo Bach[2], Christina Köpken-Watts[2], Roland Potthast[2]

1) Hans-Ertel-Center for Weather Research / Ludwig-Maximilians-Universität, Munich, Germany
2) Deutscher Wetterdienst (DWD), Offenbach am Main, Germany
3) University Vienna, Vienna, Austria
4) Ludwig Maximilian University, Munich, Germany

2016 JUNE 5

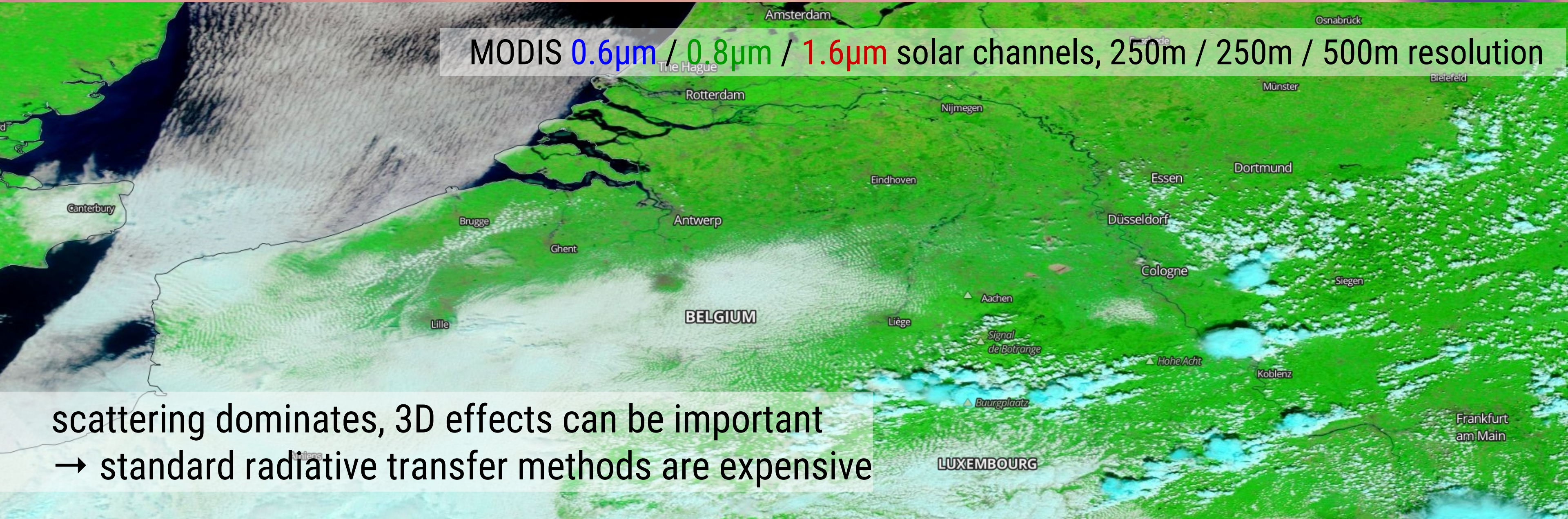
MODIS 11μm thermal (window) channel, 1km resolution



scattering and 3D RT effects: weak influence
→ efficient radiative transfer methods exist

images from NASA WorldView

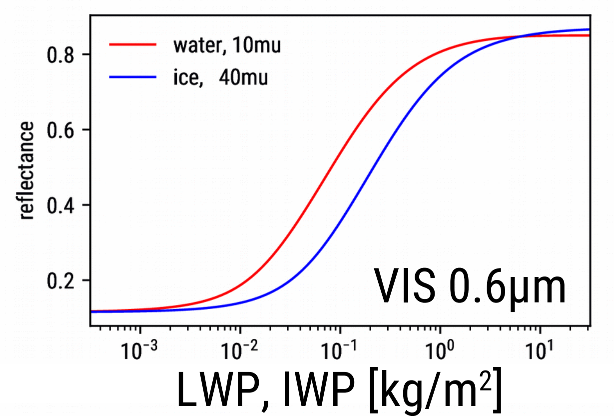
MODIS 0.6μm / 0.8μm / 1.6μm solar channels, 250m / 250m / 500m resolution

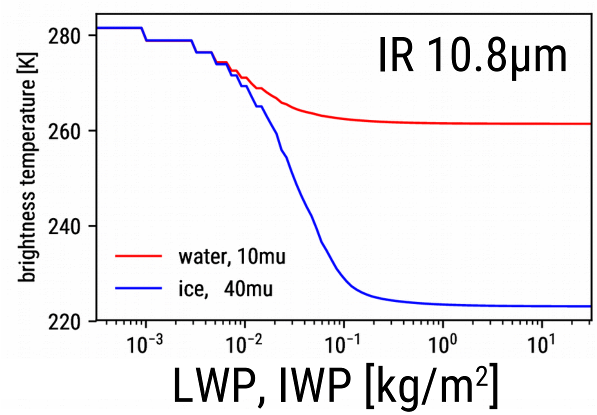


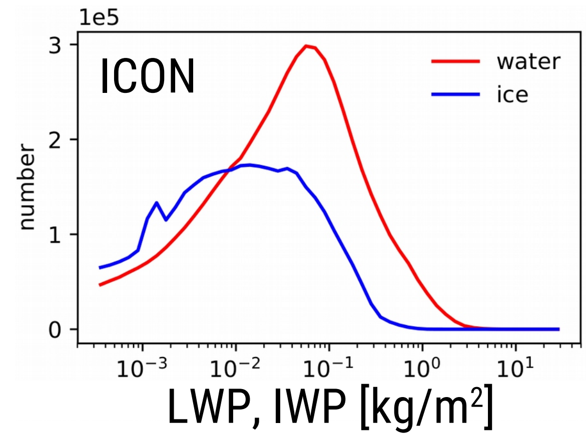
scattering dominates, 3D effects can be important
→ standard radiative transfer methods are expensive

Sensitivity to cloud water content: VIS vs. IR

0.6µm reflectance and 10.8µm brightness temperature as function of liquid / ice water path (LWP, IWP), distribution of total column cloud water content for 30 days of ICON-D2 (local area model) output from June 2016 (only 12 UTC).





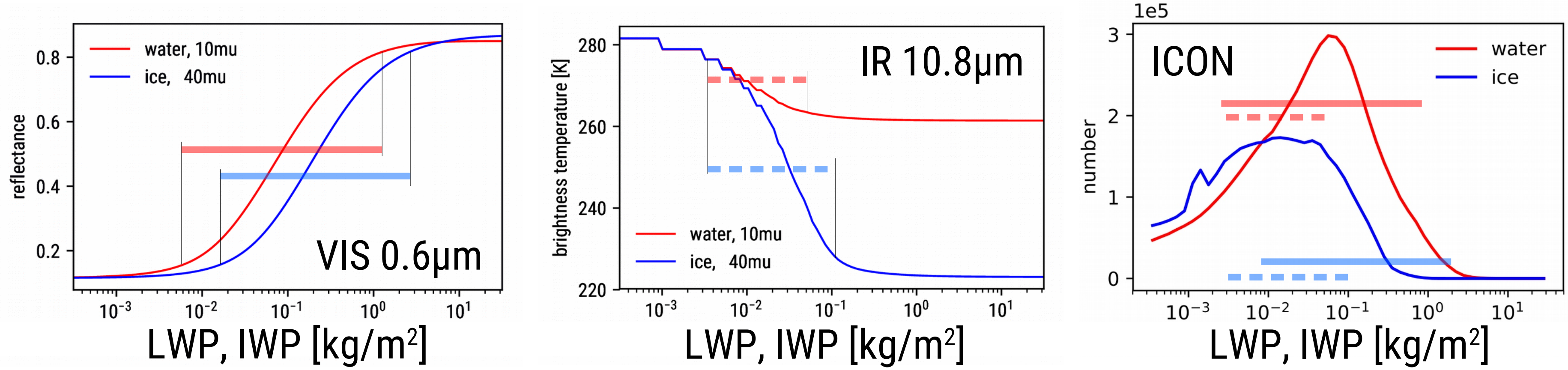


VIS 0.6µm: sensitive to changes in water path values between 0.01 kg/m² and 1 kg/m².
This is where
1) most of the clouds are (see ICON-D2 data) and
2) changes in the water path have strong impact on solar radiation

IR 10.8µm: saturates at water path values of 0.1 → cannot provide information on the thickness of the clouds that affect solar radiation most strongly

Sensitivity to cloud water content: VIS vs. IR

0.6μm reflectance and 10.8μm brightness temperature as function of liquid / ice water path (LWP, IWP), distribution of total column cloud water content for 30 days of ICON-D2 (local area model) output from June 2016 (only 12 UTC).

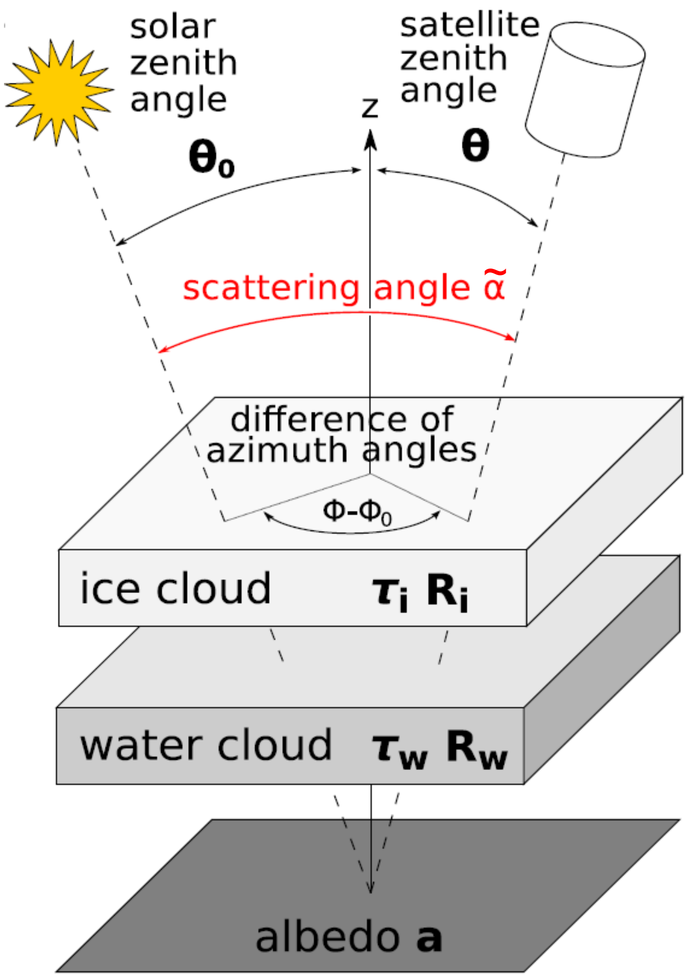


VIS 0.6μm: sensitive to changes in water path values between 0.01 kg/m² and 1 kg/m². This is where
1) most of the clouds are (see ICON-D2 data) and
2) changes in the water path have strong impact on solar radiation

IR 10.8μm: saturates at water path values of 0.1 → cannot provide information on the thickness of the clouds that affect solar radiation most strongly

Strategy for fast radiative transfer method MFASIS

Method for Fast Satellite Image Synthesis



Simplifications

- Simplified Equation:
3D RT → 1D RT (tilted independent columns)
Computational effort for a SEVIRI image of Germany:
CPU-days (3D Monte Carlo) → CPU-hours (1D DISORT)

- Simplified vertical structure:
Cloud water and ice can be separated to form two two homogeneous clouds at fixed heights without changing reflectance significantly
→ only 4 parameters (optical depth, particle size)
+ 3 angles, albedo → **8 parameters per column**

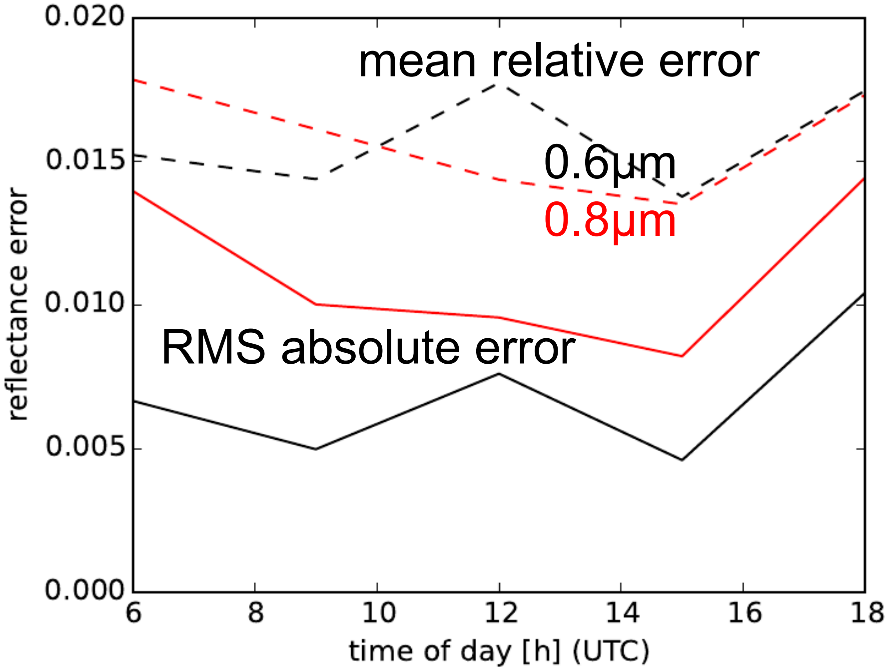
Reduction of computational effort

Compute **reflectance look-up table (LUT)** with discrete ordinate method (DISORT) for all parameter combinations
→ effort for looking up reflectances: CPU-minutes

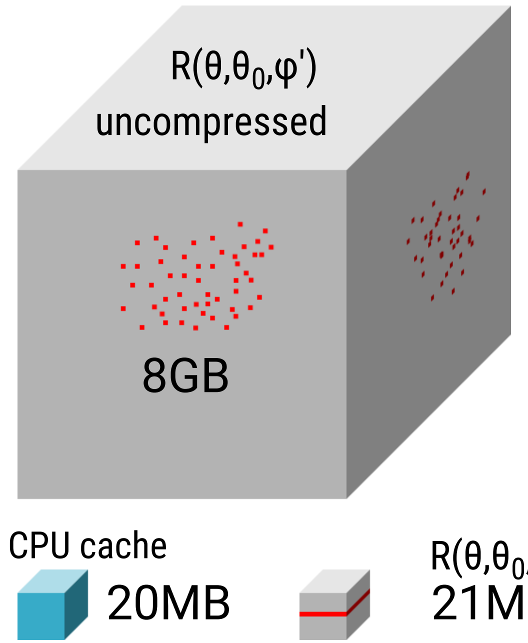
Problem: Table is huge! (about 8GB) → not suitable for online operator, slow interpolation → **compress table to 21MB** using truncated Fourier series → CPU-seconds

Accuracy and computational effort

Error of MFASIS (8 parameters/pixel) **with respect to DISORT** (full profiles available)
(model data: COSMO-DE fcsts for 10-28 June 2012)



Computational effort per column:
DISORT (16 streams): 2.3 x 10^{-2} CPUsec
MFASIS (21MB table): 2.5 x 10^{-6} CPUsec
(on Xeon E5-2650, for 51 level COSMO data)



Impact of compression on performance?

Without compr.: LUT >> cache → slow…

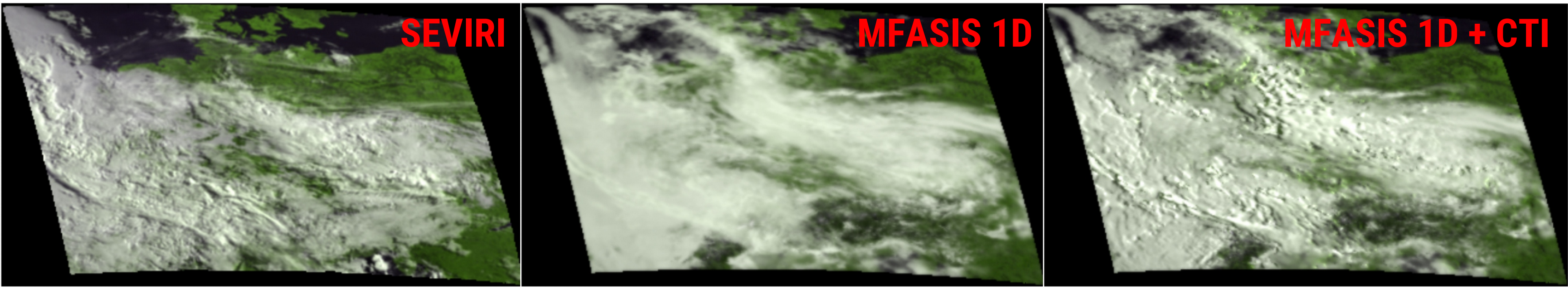
compression → cache used efficiently

Relative error < SEVIRI calibration error (~4%) for almost all pixels

NWP-SAF → MFASIS has been included in RTTOV 12.2 by DWD + MetOffice

Existing and planned extensions

- **Approximation for impact of inclined cloud tops (3D radiative transfer effect):**

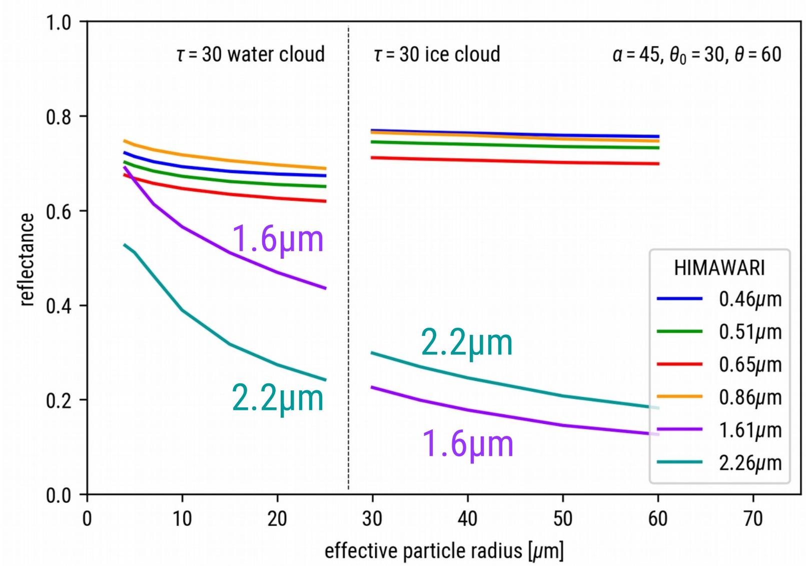


Increases information content, reduces error (Scheck et al. 2018)

- **Solar infrared channels:**
  high sensitivity to effective particle radius,
  1.6µm channels allows for distinguishing water from ice clouds

  Requires better way to determine representative value from profile of effective radii.

  Work in progress... (F. Baur)



Existing and planned extensions

- **Water vapor sensitive channels** (MSG 0.8µm, MTG 0.9µm): Linear correction implemented (see poster C. Stumpf). Nonlinear solution would require *additional LUT dimensions:* water vapor content above / below clouds
- **Blue / UV-A channels:** May require Rayleigh scattering correction (air mass above cloud as *additional input variable?*)
- **Mixed-phase cloud correction:** Current solution (interpret ice in mixed-phase cloud as water, see poster C. Stumpf) removes largest errors. Clean solution: mixed-phase ice content as *additional LUT dimension*
- **Aerosols:** *Many species, optical properties depend on size and humidity, vertical order varies*

→ **Many planned extensions require additional input variables / LUT dimensions.**

Impact on LUT size: M additional dimensions with N values → LUT factor N^M larger

→ **Additional LUT dimensions are not impossible, but also not elegant / slow / hard to handle**

Could we use machine learning to replace the LUT?

Which ML approach? Simple and very popular (→ easy to use libraries, hardware support):
(Deep) feed forward neuronal network = multilayer perceptron

Potential main advantages:
- Much less than the 8GB DISORT data generated for the LUT may be required for training
- The network parameters should also be much smaller than the 21MB compressed LUT
→ additional dimensions feasible

First goal: Replace current MFASIS LUT by a feed forward network (no additional dimensions).

In contrast to typical ML problems:

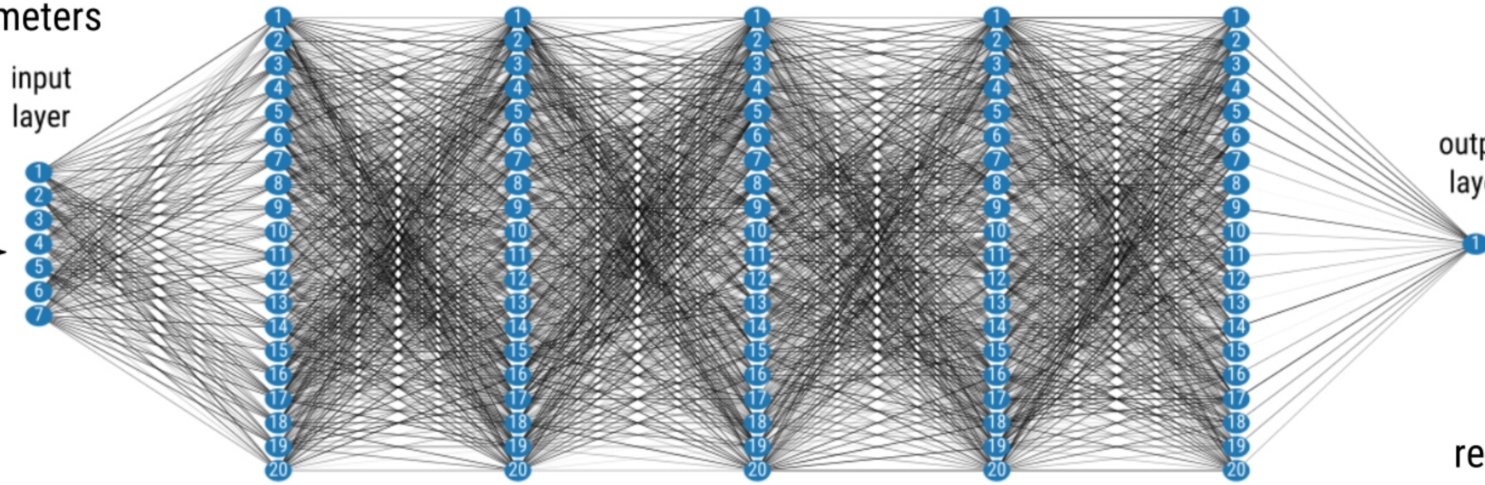
- We have already a complete training data set: The uncompressed LUT from MFASIS (8GB) for which the resolution in all dimensions was optimized to meet the target accuracy
- We have to compete with MFASIS, which is already quite fast (NN evaluation speed matters)
- To match the speed of MFASIS, the network is not allowed to be large ($O(10^3)$ parameters) (image recognition: typically > 10^6 parameters)

→ To be investigated: **Can a sufficiently fast (=small) network be sufficiently accurate?**

Feed forward network

simplification: same number of nodes in all hidden layers

normalized MFASIS input parameters



reflectance

$$s_{l+1,n} = f(b_{l,n} + \sum s_{l,m} * w_{l,n,m})$$ = Output signal of node n, layer l+1

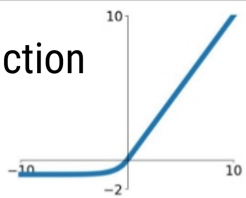
bias

weight

Signal from node m in layer l

ELU activation function

$$\begin{cases} x & x \geq 0 \\ e^x - 1 & x < 0 \end{cases}$$



N nodes/layer, **L** layers → number of parameters **P ≈ LN²** ~ effort for evaluation

(in this example P ≈ 2000)

roughly...

Training the network

The following methods were used:

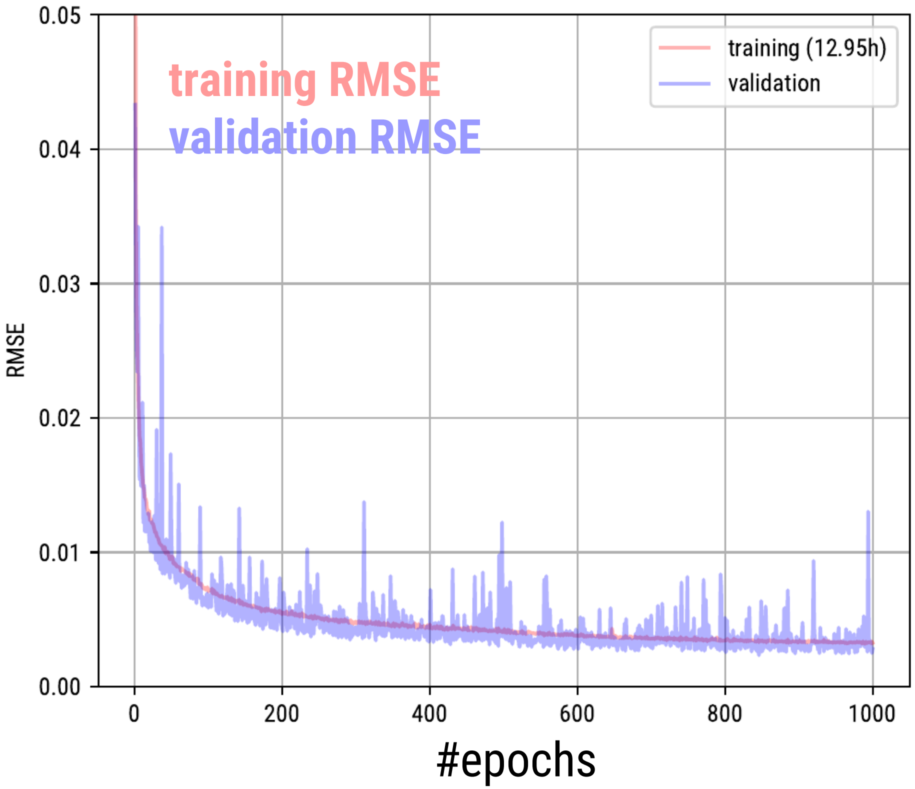
- **Backpropagation method:** Compute gradient of loss function (reflectance error) with respect to each weight by the chain rule, iterating backwards one layer at a time from the last layer to avoid redundant calculations of intermediate terms in the chain rule.
- **Stochastic gradient descent:** Use batch of randomly selected (input,output) pairs
  Batch size has strong impact on training performance.
  batch size = 1 : High probability to end up in local minimum, computationally inefficient
  batch size = all (Gradient descent) : Large amounts of data in each iteration, inefficient
  Optimal batch size: somewhere between… 256 turned out to be good here.
- **Adam** (Adaptive Moment Estimation): Method in which the learning rate (step size) is adapted for each of the parameters. An overall learning rate still has to be specified.

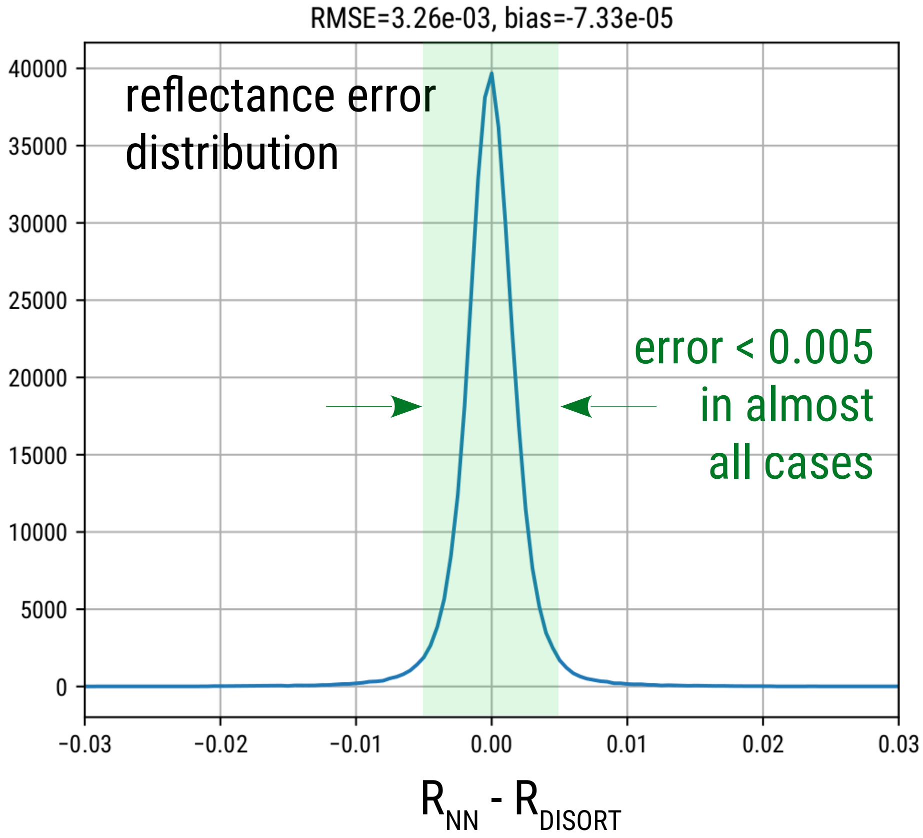
Implemented in **Tensorflow** (easy to use Python interface, good documentation, GPU support) but also in many other open source machine learning libraries (we do not depend on Google).

Important parameters:

f_T = Fraction of the full 8GB data set that is actually used for training
LR = learning rate (controls step size in gradient descent)
NE = Number of "epochs" (how often is the training data used)

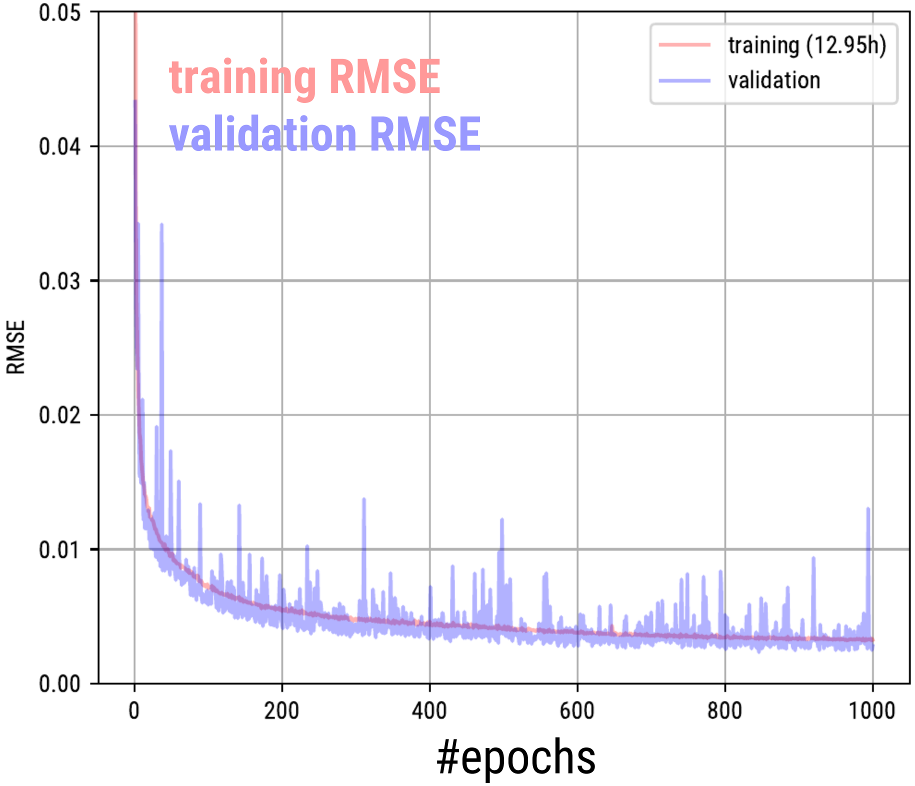
Example: P=3000, L=5, N=26, f_T=1%, trained for 13h

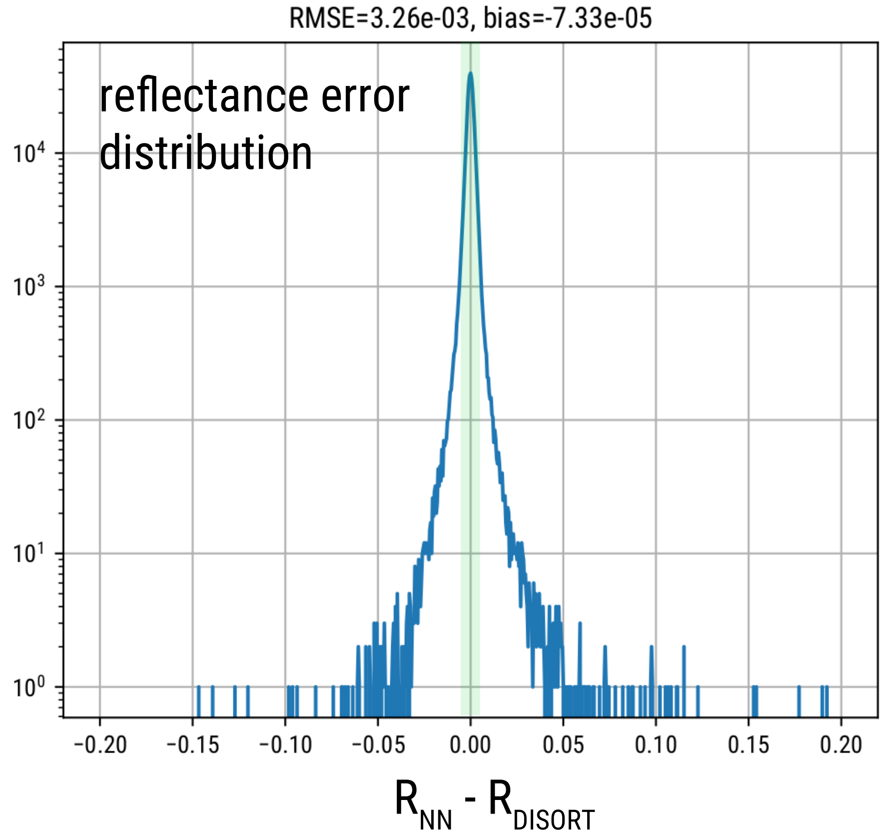




Reflectance RMSE for independent validation data set (also 1%) after 13h training is 0.003 (similar to fit + interpolation error of MFASIS).

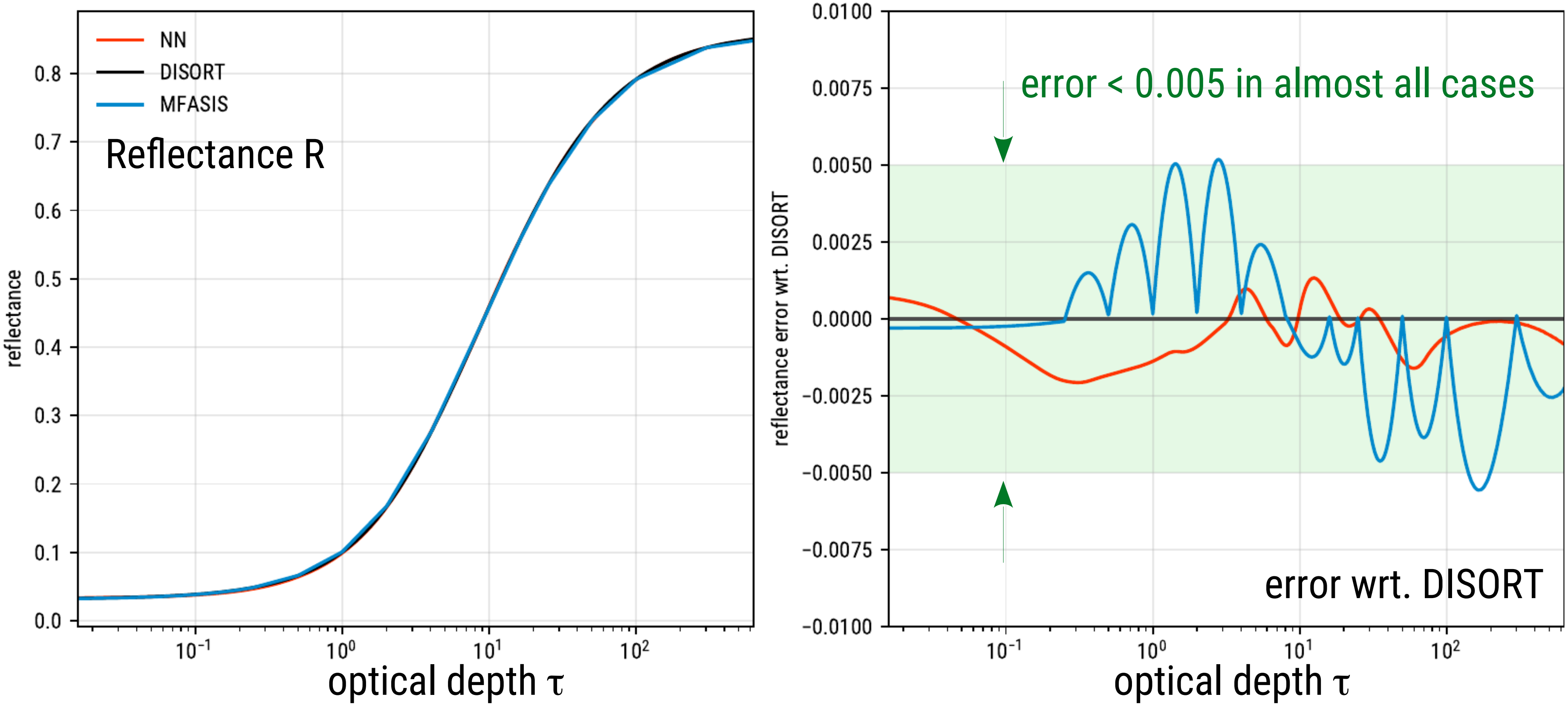
Example: P=3000, L=5, N=26, f_T=1%, trained for 13h





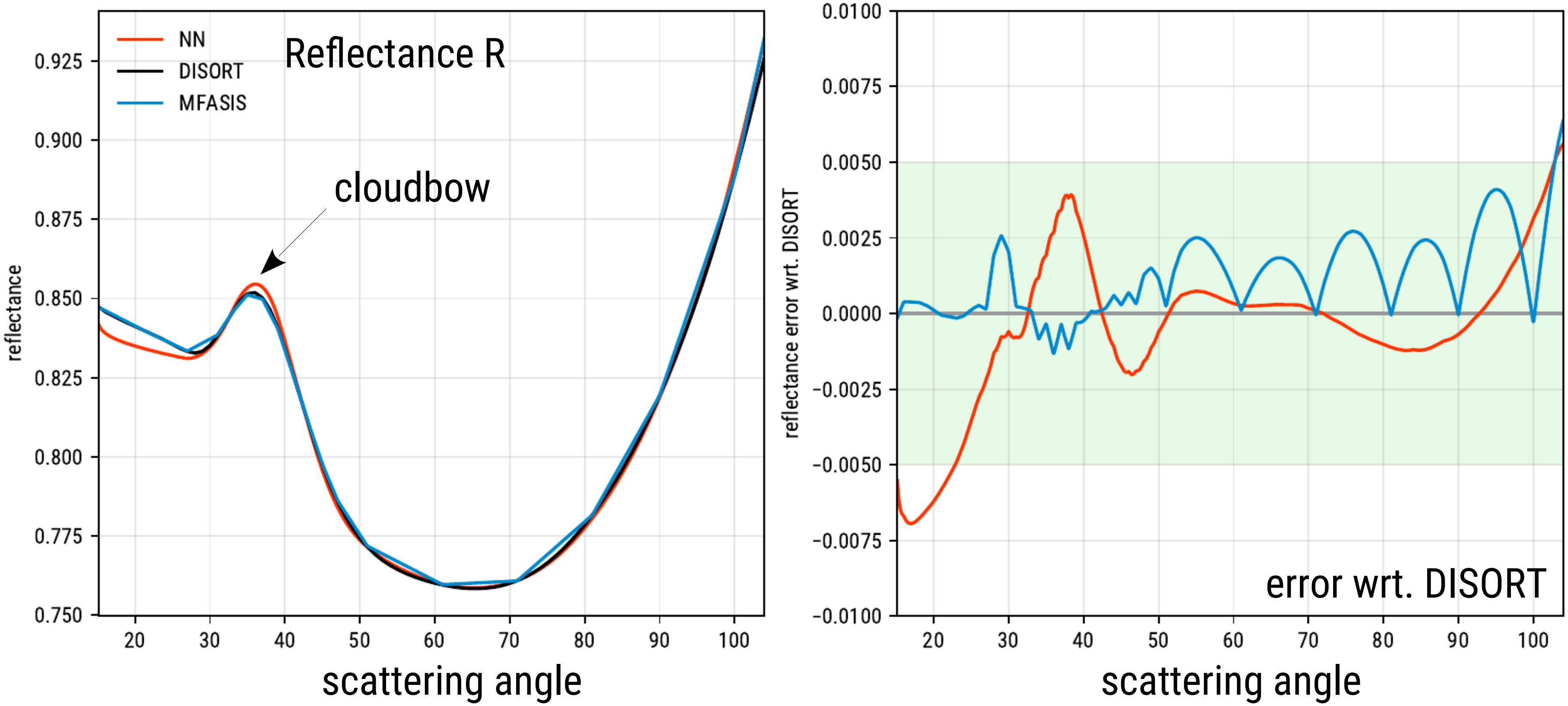
Reflectance RMSE for independent validation data set (also 1%) after 13h training is 0.003 (similar to fit + interpolation error of MFASIS).

Example: P=3000, L=5, N=26, f_T=1%, trained for 13h



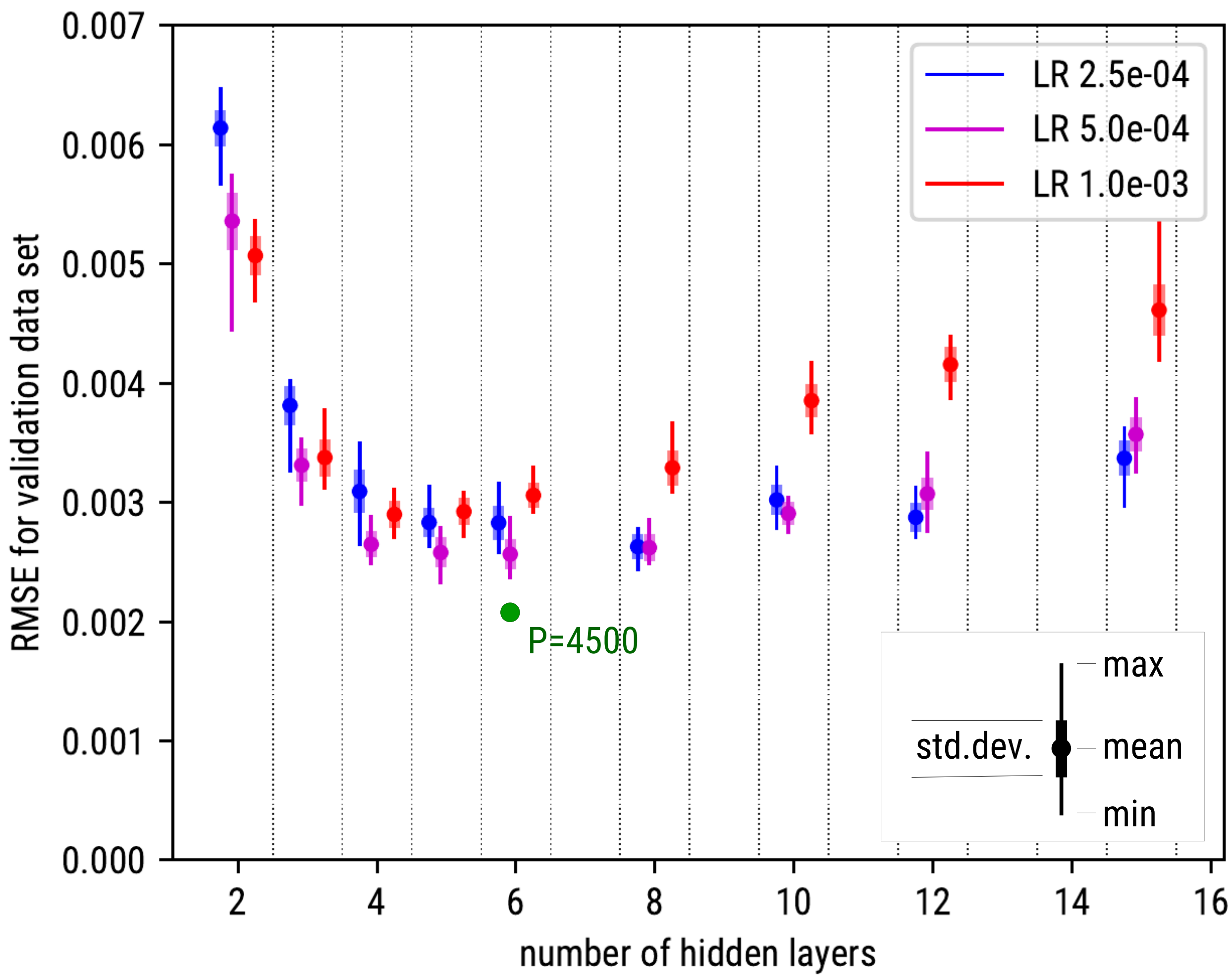
Reflectance as a function of water cloud optical depth:
Errors with respect to DISORT reference result are within range 0.01
MFASIS errors are very small at points contained in LUT

Example: P=3000, L=5, N=26, f_T=1%, trained for 13h



Reflectance as a function of scattering angle for dense water cloud:
Somewhat larger errors inside cloudbow (scattering angle < 40°)
Details depend on selection of training data.

Optimal number of layers / learning rate, robustness



For constant number of parameters **P = 3000** we varied the learning rate and the number of hidden layers (→ also nodes/layer varied).

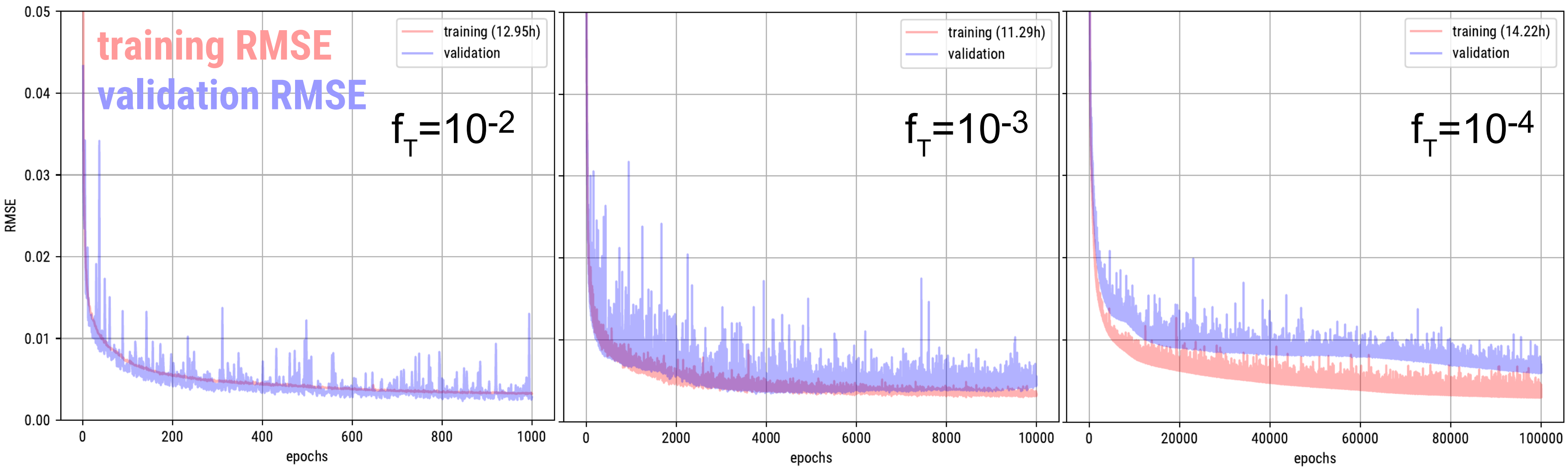
For each setting 8 networks were trained using different parts of the 8GB data (always 1%).

4 – 8 layers and a learning rate of 0.0005 yield best results (lowest RMSE for a 1% validation data set) and a standard deviation of about 10%.

Increasing number of param. by **50%** → RMSE reduced by about **15%**

Required size of training data set

Test with the same network structure (L=5, N=26) and similar training time (11-14 hours), but different sizes of the training data set (validation data set size = training data set size).



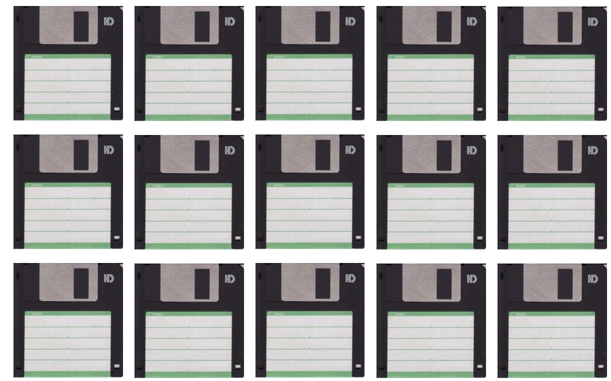
Using 1% of the 8GB: Ok, RMSE for validation data set ≈ RMSE for training data set (+ noise)
0.1% : Still ok, but first signs of overfitting (NN adapts too strongly to details of training data)
0.01% : Clear overfitting problem (but validation RMSE still decreases at end of training)

Overfitting can be reduced using regularization methods (L2, dropout, … – work in progress) which should also increase robustness of the results.

Training and LUT/NN data size comparison for $f_T=10^{-3}$

uncompressed LUT 8GB

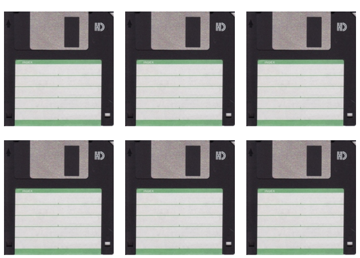
/ 381



compressed MFASIS LUT: 21MB

/ 266667

/ 1000 statistical learning



Training data: 8MB

/ 267

Storing the function instead of the data

without images!

Network Parameters: 30KB

Nonlinear, tangent linear and adjoint Fortran codes

Nonlinear (“inference”): Tensorflow predict() is slow for small networks (at least in version 1.4)

→ Implemented and optimized Fortran code

- Vectorization (using AVX512 units) → factor ~5 gain in speed
- NN with 3000 Parameters is **somewhat faster than MFASIS-LUT**
- A further factor ~2 could be gained by avoiding exp() in activation function

TL & AD: Variational / hybrid DA algorithms require tangent linear (TL, **H**) and adjoint (AD, $\mathbf{H}^T$) versions of nonlinear observation operators **H** (even if adjoint of the NWP model is not required).

→ Implemented TL & AD versions of the nonlinear Fortran forward code

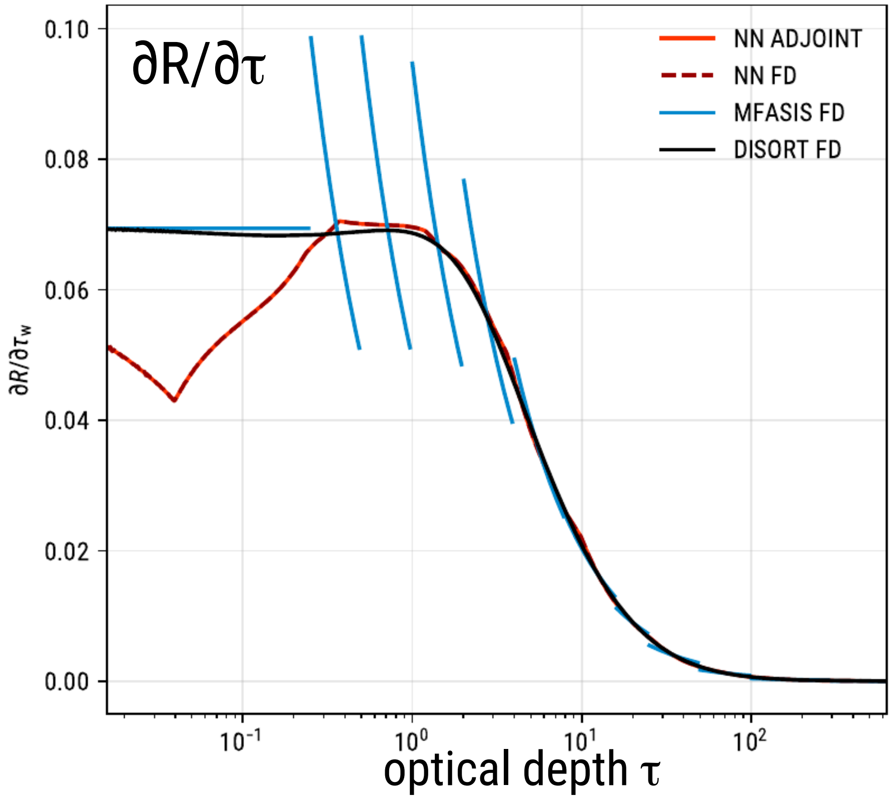
Code passes scalar product test $< H^T Ha, b > = < Ha, Hb > = < a, H^T Hb >$ for randomly chosen a, b and reproduces derivatives computed with finite differences.

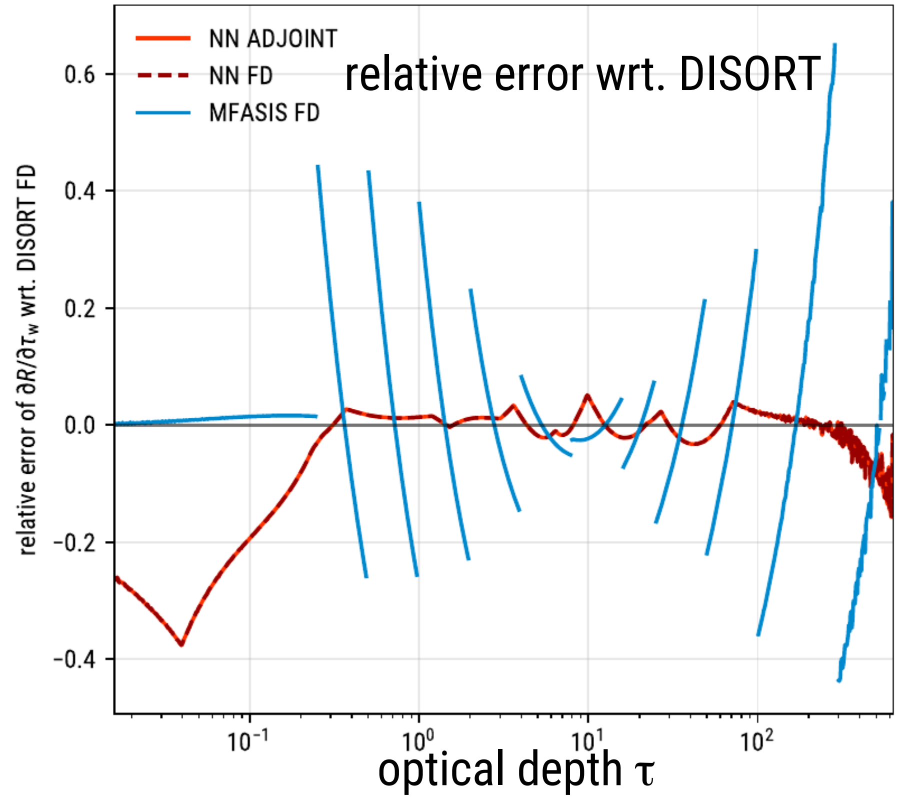
Computational effort: TL is 30% slower, AD 65% slower than nonlinear code

Nice additional benefit of using neural networks in variational DA context:

Whatever a neuronal network is trained for, the TL/AD code will always be the same.

Tangent linear and adjoint versions





Derivative of reflectance with respect to water cloud optical depth computed using finite differences (FD) of NN, MFASIS and DISORT results as well as the NN adjoint code.
Adjoint is full consistent with NN FD and continuous (not for MFASIS – piecewise linear interp.)
Noise for high optical depths, stronger deviation from DISORT only for very low optical depths.

Outlook

Next steps in operator development
- Extension of MFASIS to solar infrared channels
- NN approach: Full evaluation of errors and robustness, optimizations
- Water vapor and mixed phase cloud ice as additional inputs
- Neural-network based MFASIS for aerosols

Current applications
- Using visible channels for convective-scale DA (see Poster L. Bach)
- Evaluation of visible channels for global DA (see Poster C. Stumpf)
- Assimilation of visible and thermal channels (J. Schroettle)
- Model tuning and DA using vis. channels aimed at improving forecasts for photovoltaics power production (S. Geiss, A. de Lozar, A. Seifert)

Publications:

- Scheck, Frerebeau, Buras-Schnell, Mayer (2016): *A fast radiative transfer method for the simulation of visible satellite imagery*, Journal of Quantitative Spectroscopy and Radiative Transfer, 175, 54-67.
- Scheck, Hocking, Saunders (2016): *A comparison of MFASIS and RTTOV-DOM*, NWP-SAF visiting scientist report, http://www.nwpsaf.eu/vs_reports/nwpsaf-mo-vs-054.pdf
- Scheck, Weissmann, Mayer (2018): *Efficient methods to account for cloud top inclination and cloud overlap in synthetic visible satellite images*, JTECH, 35(3):665–685.
- Scheck, Weissmann, Bach (2020): *Assimilating visible satellite images for convective-scale numerical weather prediction: A case study*, submitted to Q. J. R. Meteorol. Soc.
- Schroettle, J., M. Weissmann, L. Scheck, A. Hutt, (2020): *Assimilating visible and thermal radiances in idealized simulations of deep convection,* submitted to Mon. Wea. Rev.