Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Home Affairs FDHA
Federal Office of Meteorology and Climatology MeteoSwiss

Stochastic machine learning for atmospheric fields with generative adversarial networks

Jussi Leinonen
With contributions from Alexis Berne (EPFL), Daniele Nerini (MeteoSwiss), Tianle Yuan (NASA-GSFC/UMBC), Alexandre Guillaume (NASA-JPL)

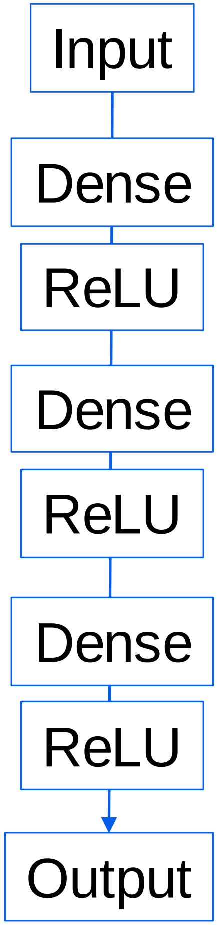
Joint IS-ENES3/ESiWACE2 Virtual Workshop, 17.03.2021

Neural networks

- A neural network is a series of fixed mathematical operations (“*layers*”) with *trainable parameters* and a training objective
- The fundamental types of layers are:
    - Affine transformations **y**=**Wx**+**b**, with **W** and **b** trainable
    - *Nonlinearities*, e.g. tanh or ReLU
- All layers are piecewise *differentiable*, so we can compute analytically the derivative of the objective w.r.t. each weight
    - ➢ We can optimize weights with gradient descent, *automatic differentiation* available in many packages (e.g. TensorFlow, PyTorch)

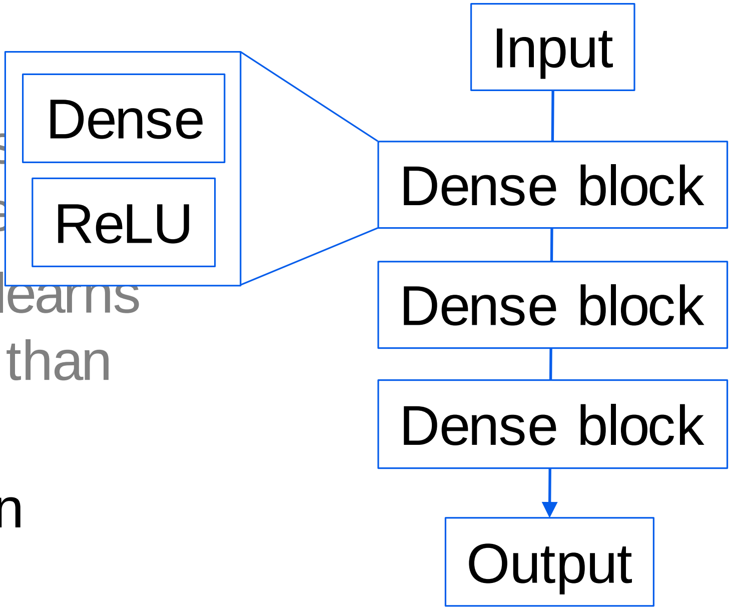
Neural networks

- Simplest neural networks repeat dense affine transformations and nonlinearities
- *Deep neural networks* use many layers in sequence
  - Each trainable layer learns higher-level features than the previous one



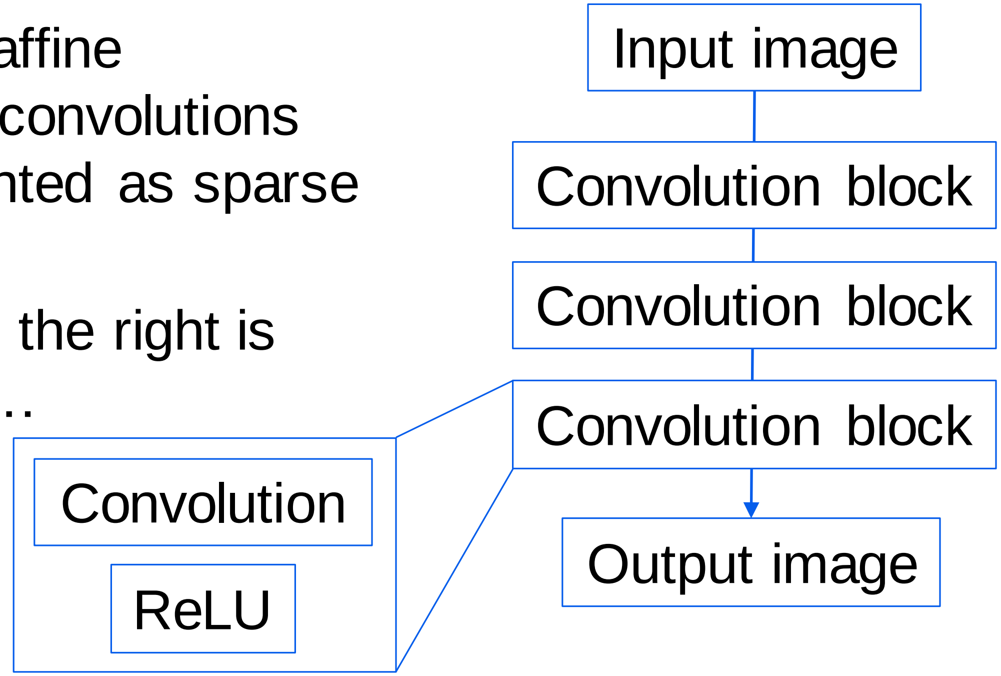
Neural networks

- Simplest neural networks repeat dense affine transformations and nonlinearities
- *Deep neural networks* use many layers in sequence
  - Each trainable layer learns higher-level features than the previous one
- *Blocks* of layers are often repeated

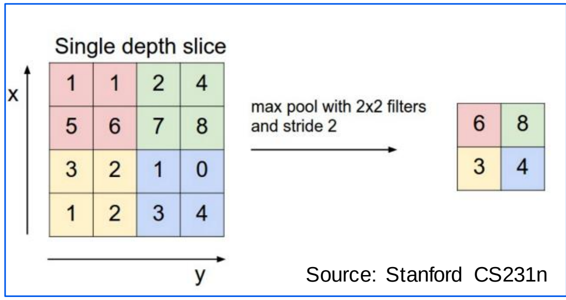


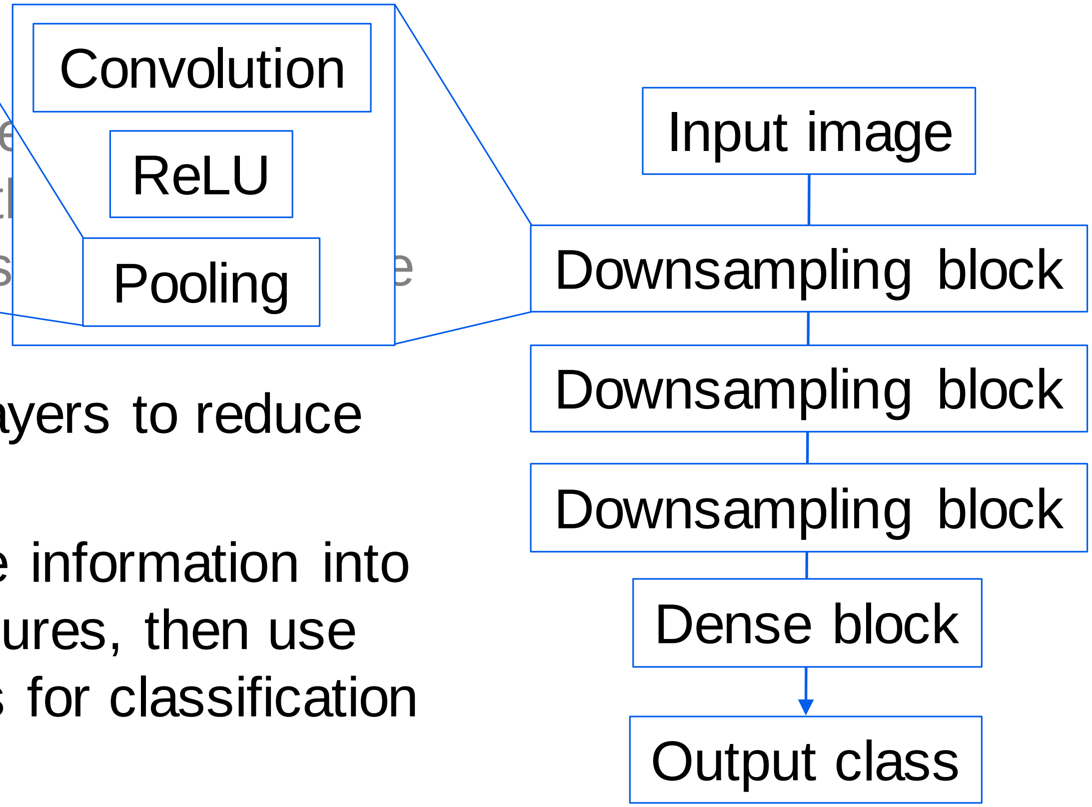
Convolutional networks

- Replace dense affine transforms with convolutions (can be represented as sparse matrices)
- The example on the right is image-to-image…



Convolutional networks



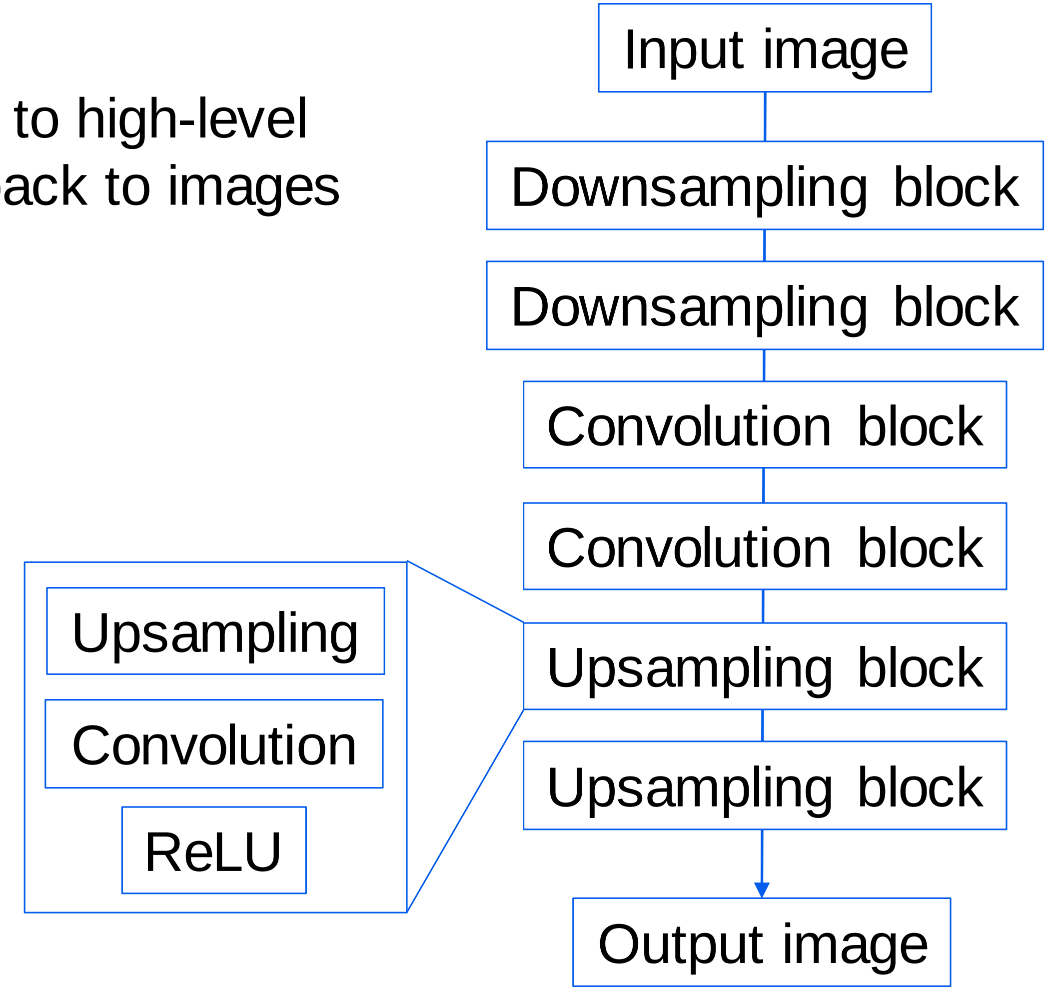


matrices)

- Add *pooling* layers to reduce image size
- Encode image information into high-level features, then use these features for classification

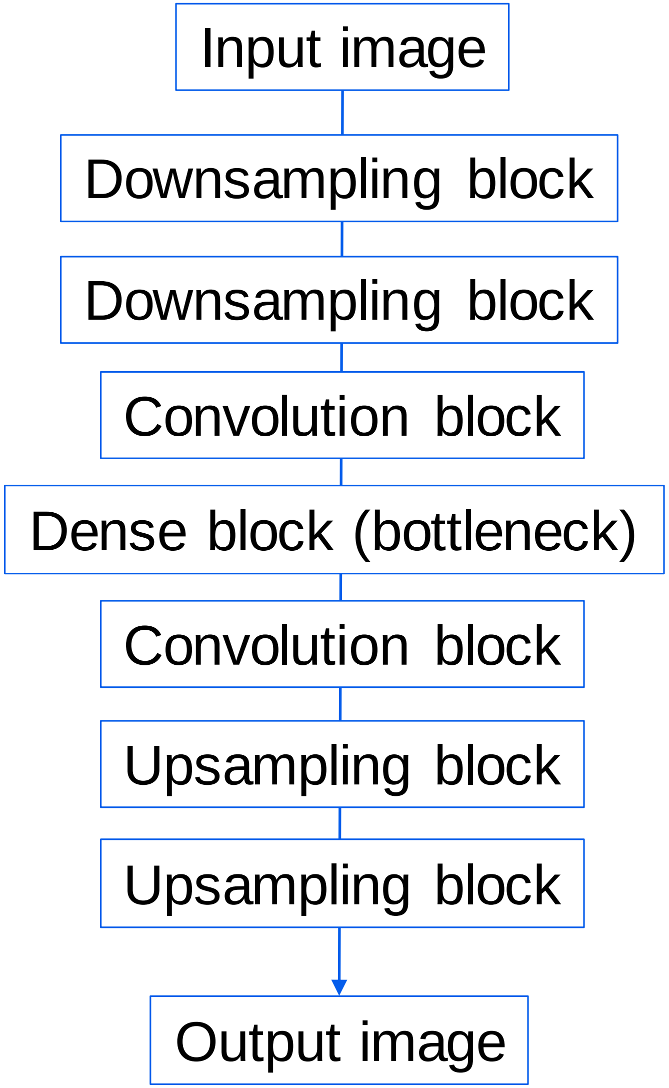
Encoder-decoder architectures

- Encode images to high-level features, then back to images



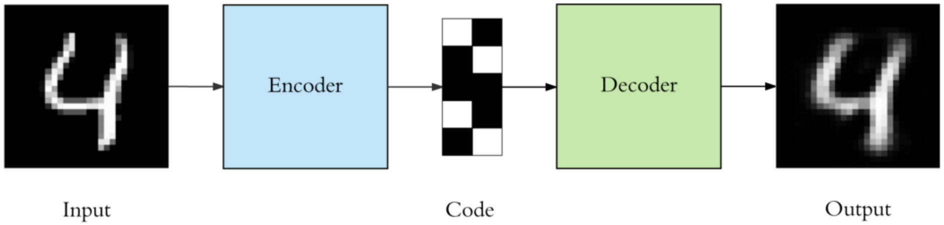
Autoencoders

- Optimize input and output to be similar
- A “bottleneck” in the middle of the network encodes the essential features of the data

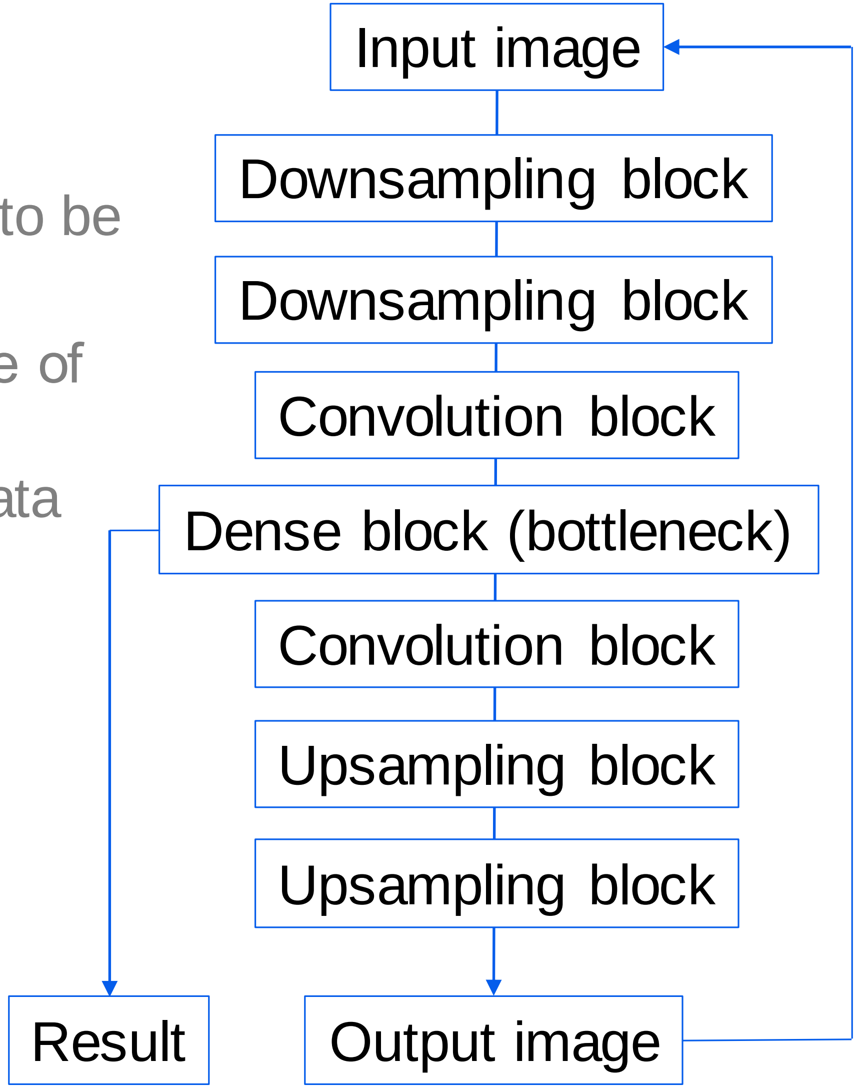


Autoencoders

- Optimize input and output to be similar
- A "bottleneck" in the middle of the network encodes the essential features of the data
- Use contents of the dense block as features
  - ➢ Unsupervised learning

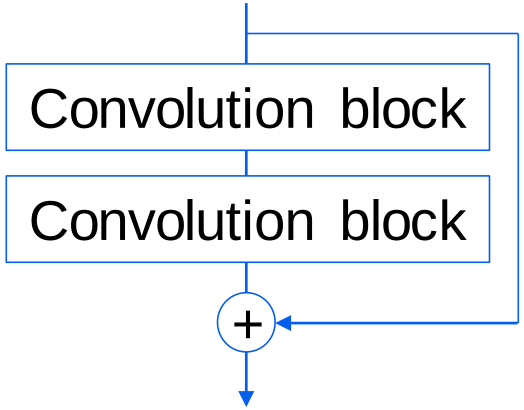


Source: Arden Dertat, 2017



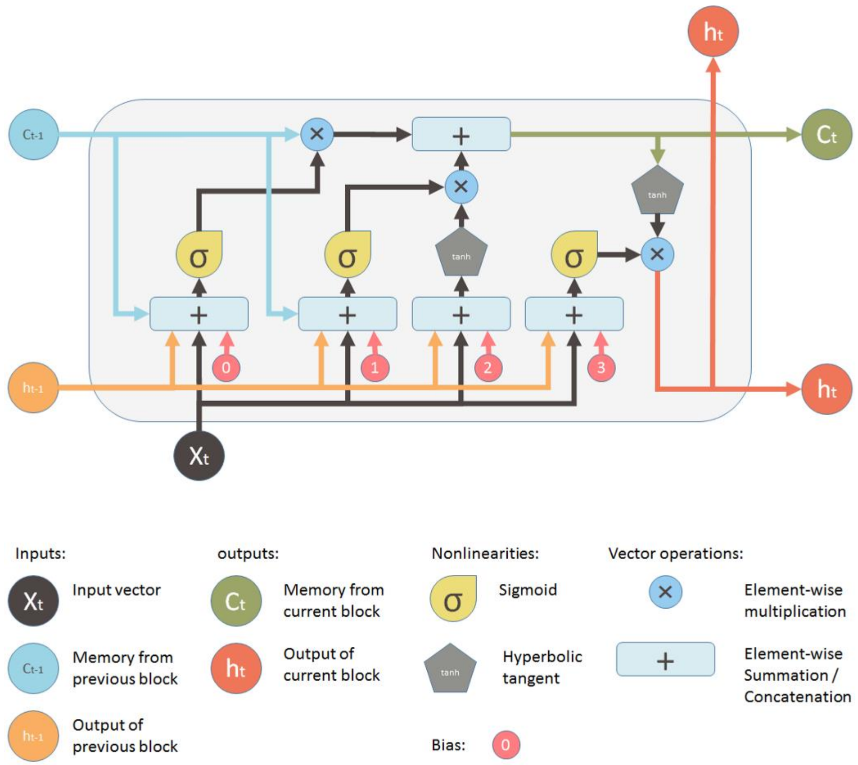
Residual blocks

- Include a *skip connection* in the network
  - Learn the residual of the previous block
  - Network can pass data through unused layers
  - Optimization gradients are better preserved



Recurrent units

- Used to model time-variable fields
- Learn update rules between time steps, encoded as trainable parameters
- Popular implementations include LSTM, GRU
- Typically used for time series and natural language processing, but implementations exist also for images evolving in time (ConvLSTM, ConvGRU)



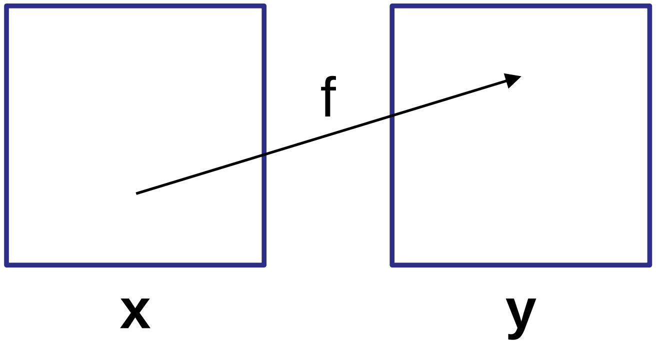
Source: Shi Yan, 2015

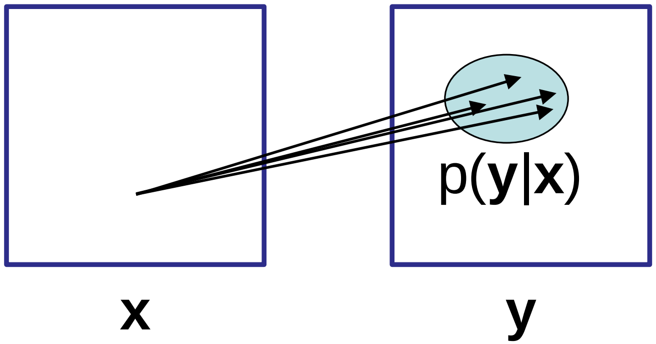
Generative models

- Typical predictive model:
  predict **y** = f(**x**)
  - One answer per input

- Generative model:
  generate samples from p(**x**)

- Conditional generative model:
  generate samples from p(**y**|**x**)
  - Multiple answers per input,
    uncertainty modeled



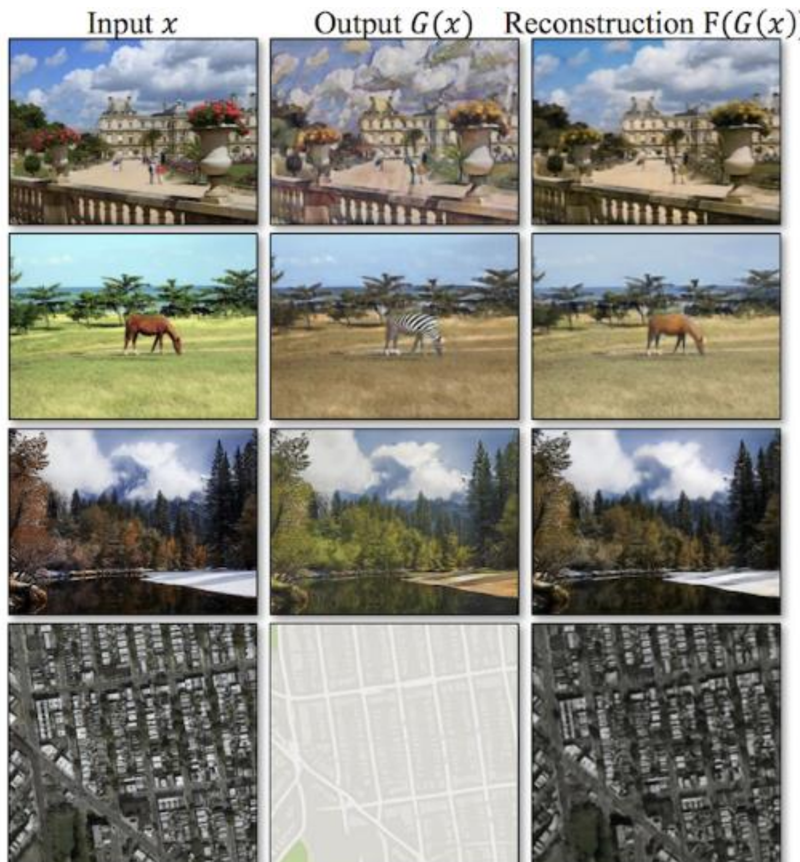


Generative adversarial networks (GANs)

Example applications:

Image generation

Domain translation

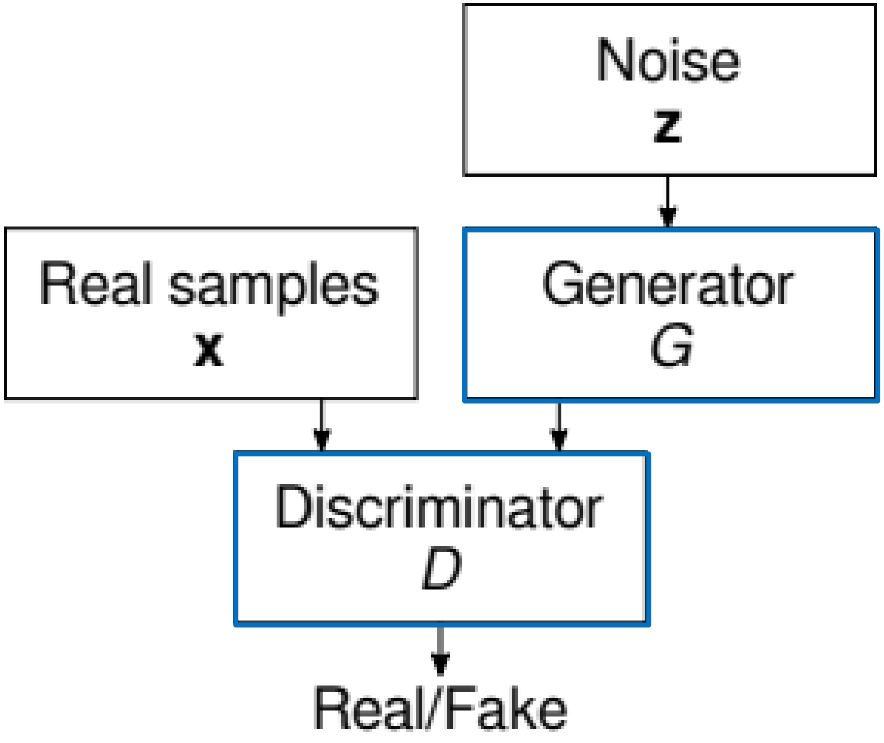


Infilling

Generative adversarial networks (GANs)

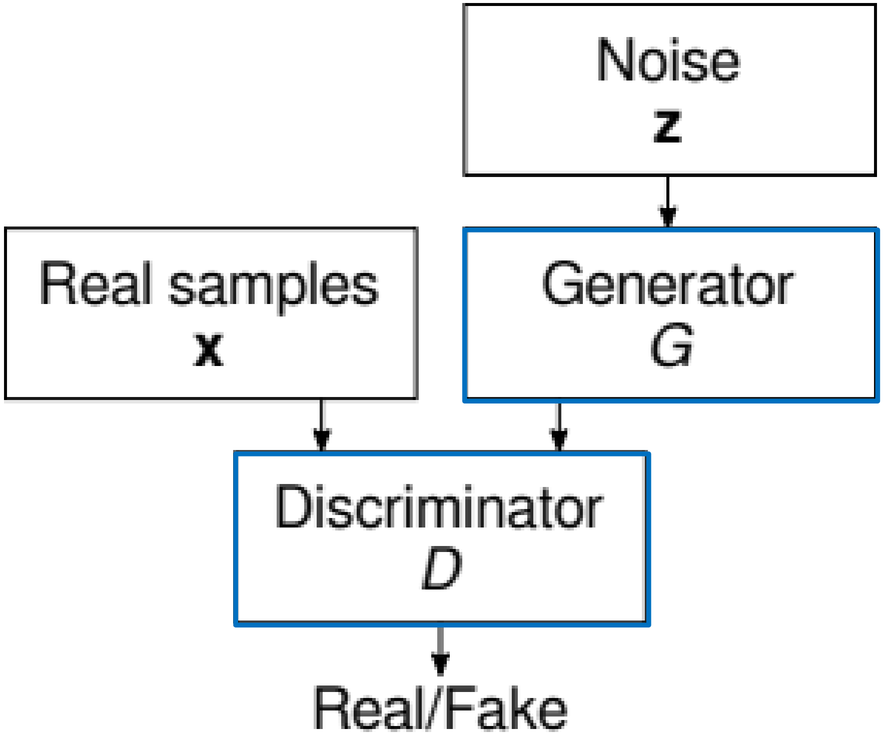
Two competing (usually convolutional) neural networks:

- Discriminator tries to distinguish real samples from generated ones
- Generator tries to output samples that discriminator considers real
  - ➢ Leans to generate realistic samples



Generative adversarial networks (GANs)

GAN generator: map
$p(\mathbf{z}) \rightarrow p(\mathbf{x})$

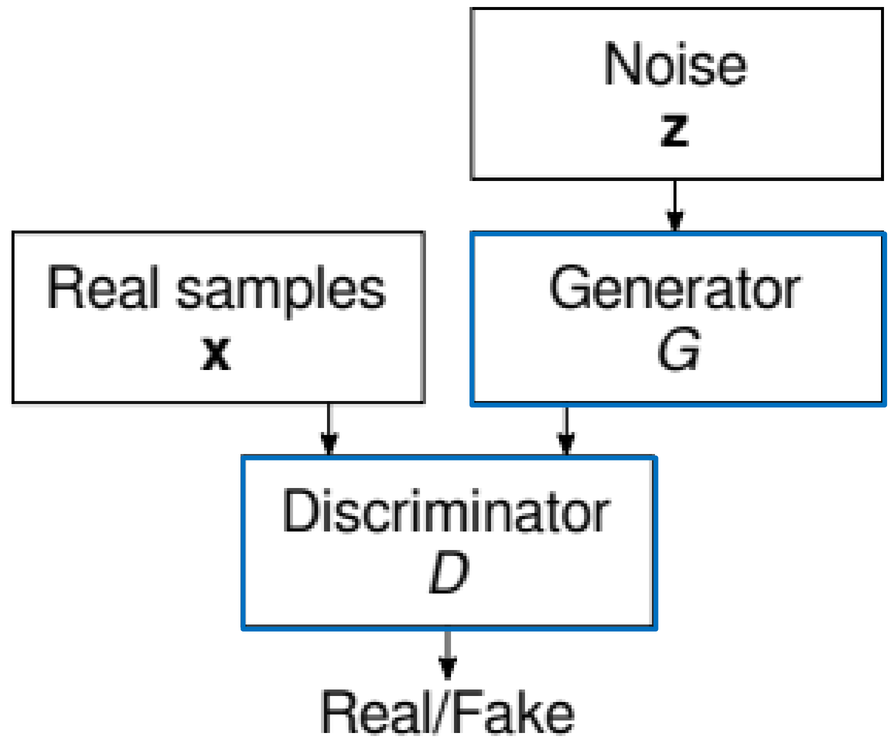


Generative adversarial networks (GANs)

GAN generator: map
p(**z**) → p(**x**)

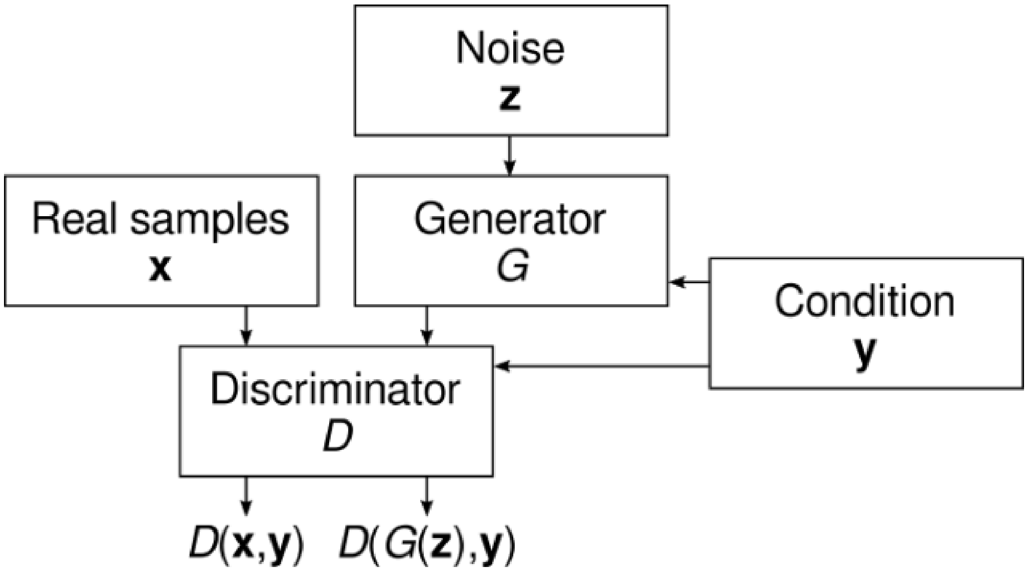
Simple

Complex,
spatially
correlated



Conditional GANs

CGAN generator: map $p(\mathbf{z},\mathbf{y}) \rightarrow p(\mathbf{x}|\mathbf{y})$



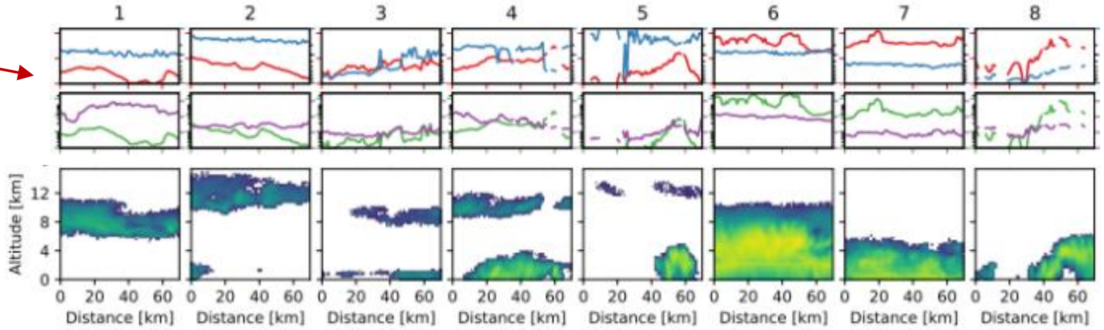
Conditional probability problems

- Ubiquitous in Climate Science
- Examples: inferring...
  - p(Quantity x | quantities y)
  - p(Quantity x | measurements y)
  - p(Future state | current and/or past state)
  - p(High resolution field | low resolution field)
  - p(Complete data | incomplete data)
- Underdetermined problems, CGANs can learn to generate the conditional *distribution* of solutions

Generating cloud profiles with CGAN

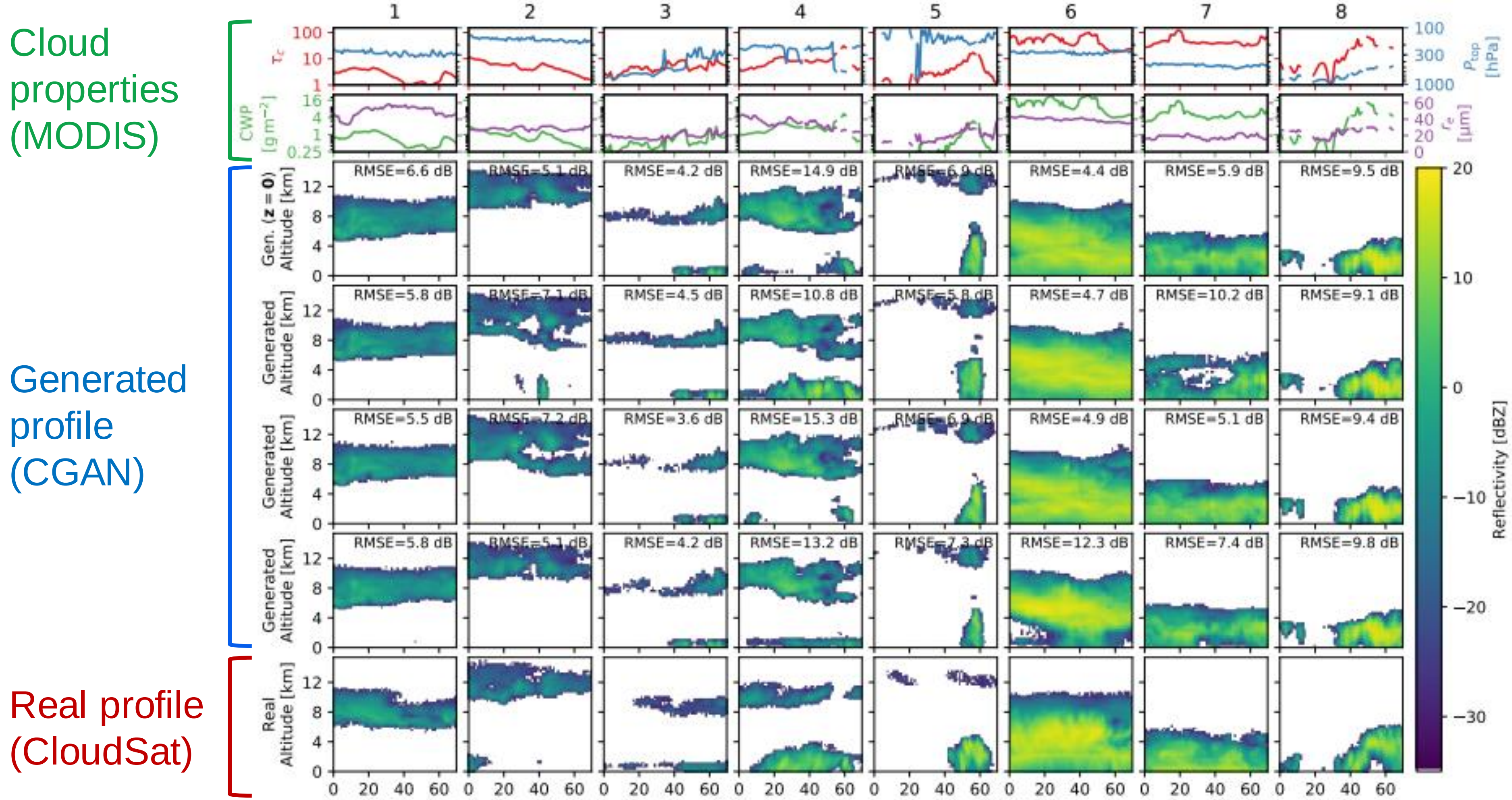
Dataset of collocated cloud observations from:

- MODIS spectrometer (1D, 4 variables)
- CloudSat radar (2D, 1 variable)



Can we train a CGAN to generate the CloudSat vertical profiles based on the MODIS data?

Generating cloud profiles with CGAN

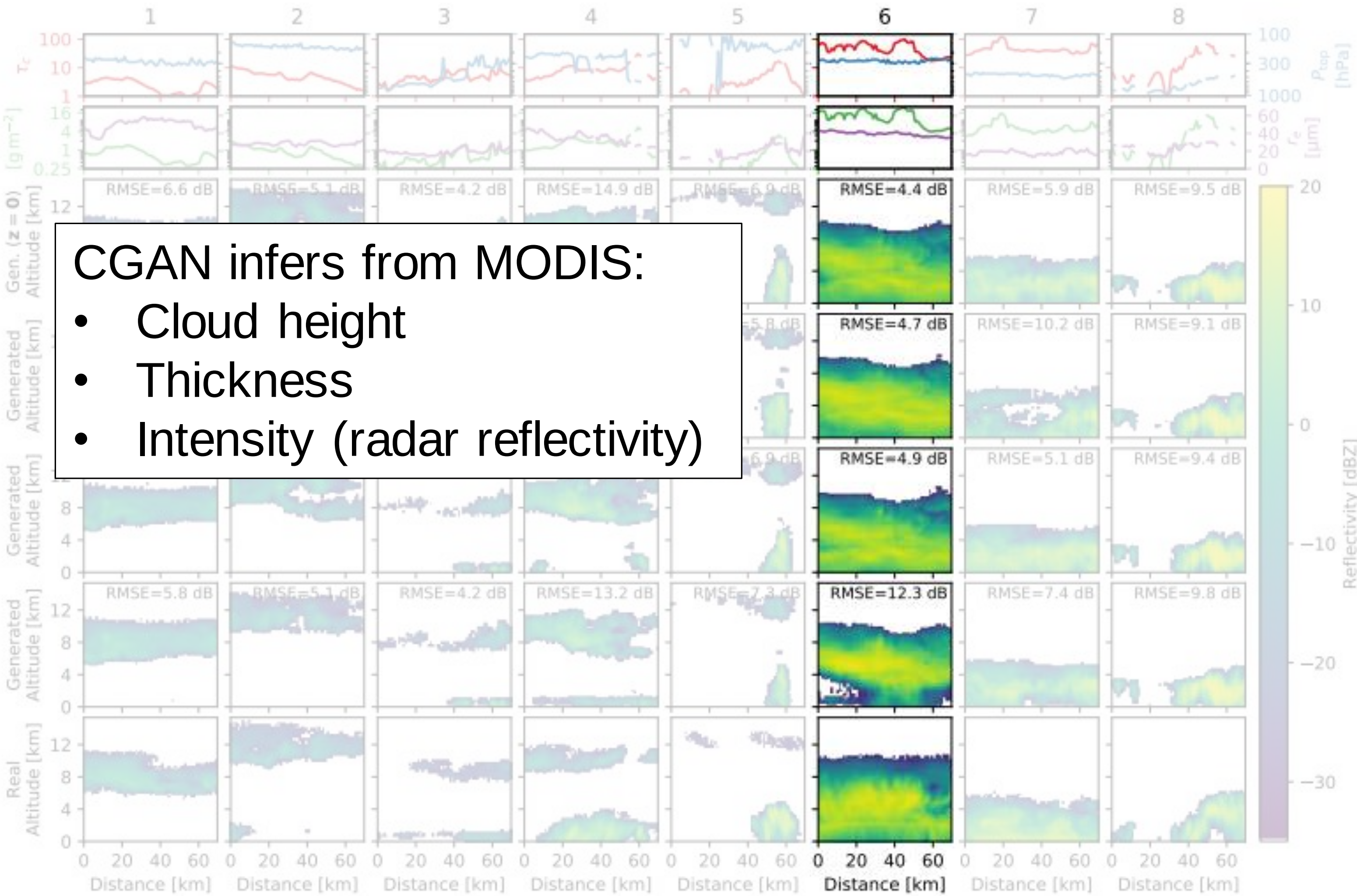


Generating cloud profiles with CGAN

Cloud properties (MODIS)

Generated profile (CGAN)

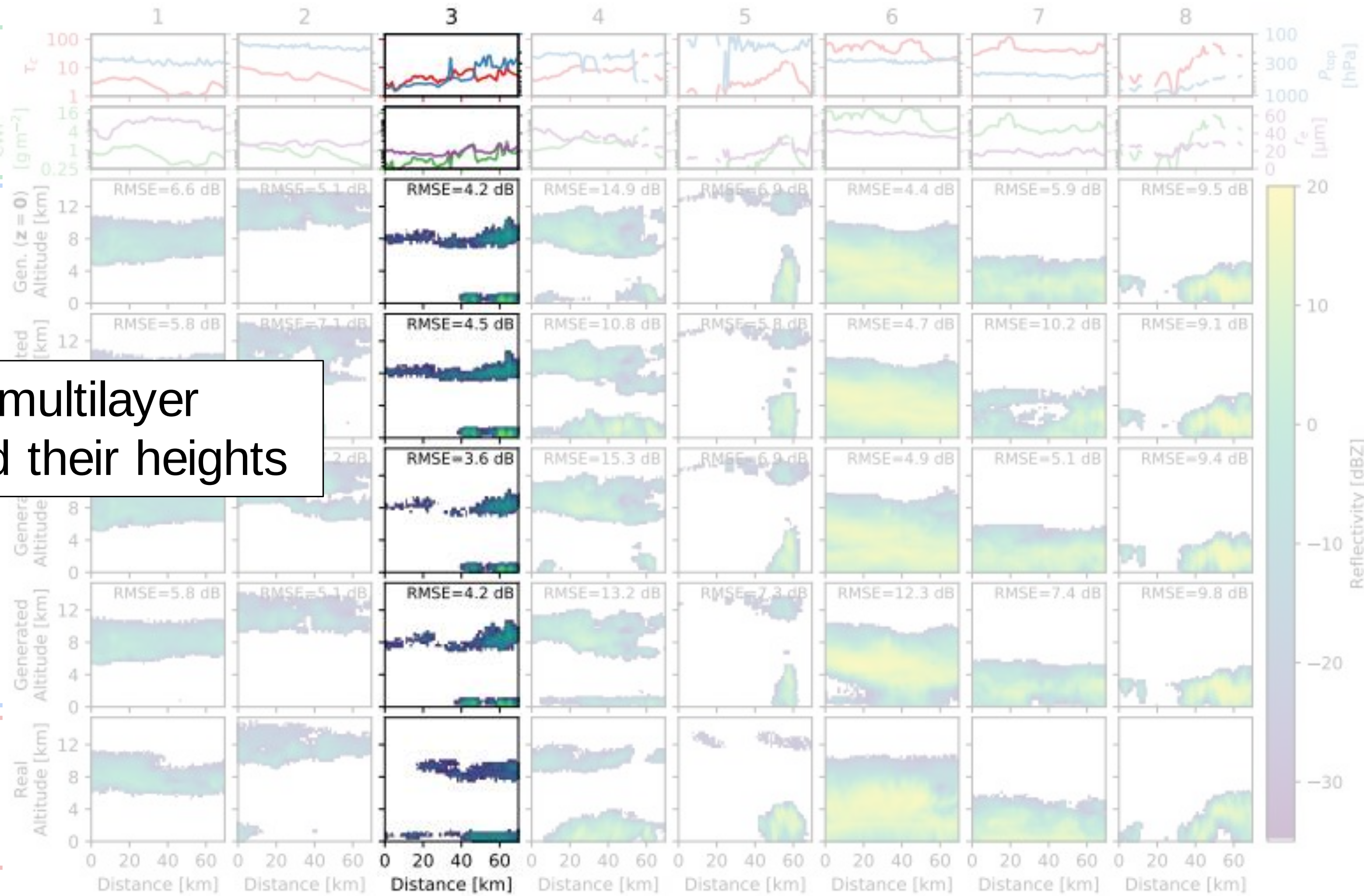
Real profile (CloudSat)



CGAN infers from MODIS:
- Cloud height
- Thickness
- Intensity (radar reflectivity)

Generating cloud profiles with CGAN

Cloud properties (MODIS)



Detecting multilayer clouds and their heights

(CGAN)

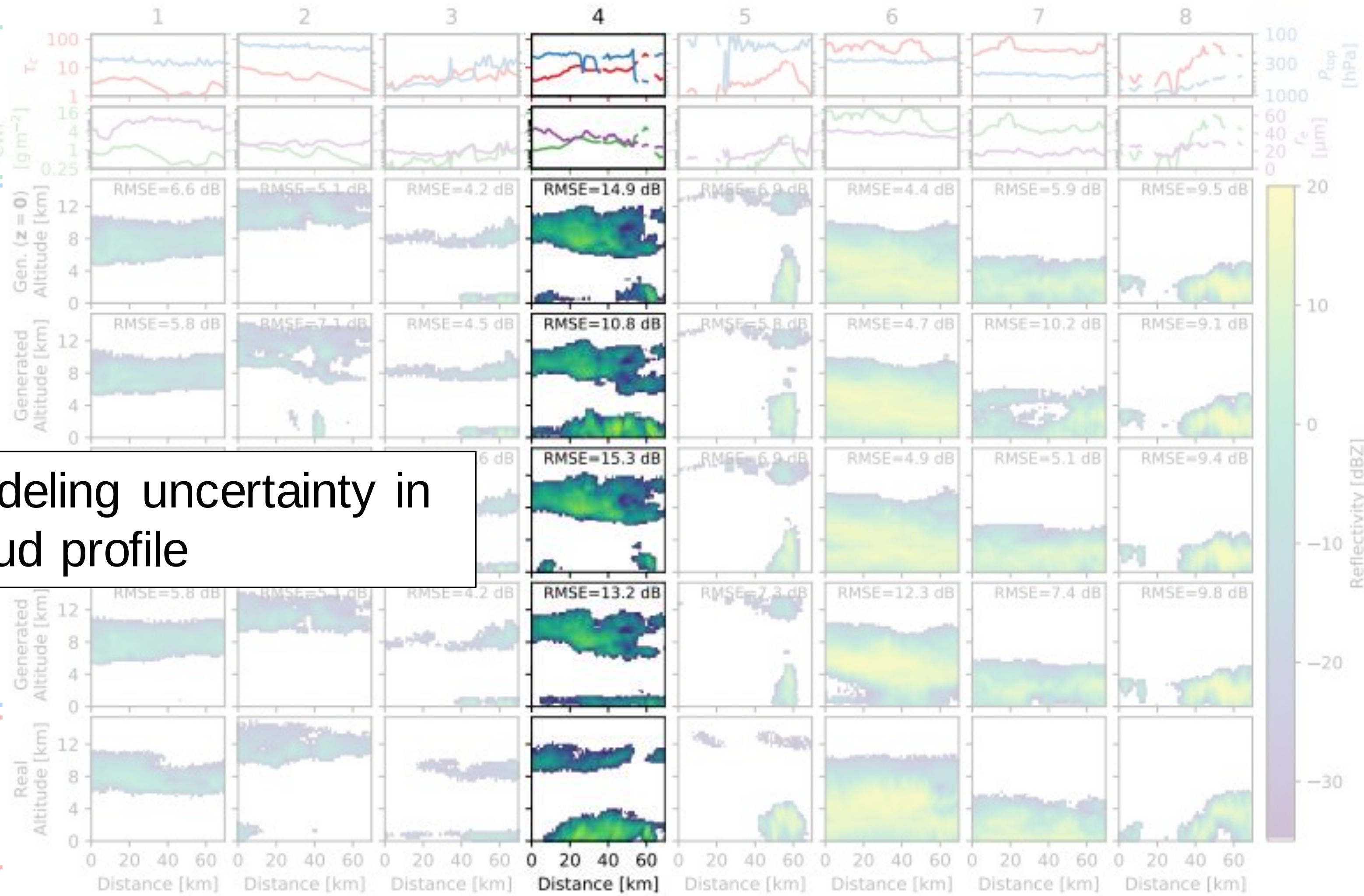
Real profile (CloudSat)

Generating cloud profiles with CGAN

Cloud properties (MODIS)

Generated profile (CGAN)

Real profile (CloudSat)



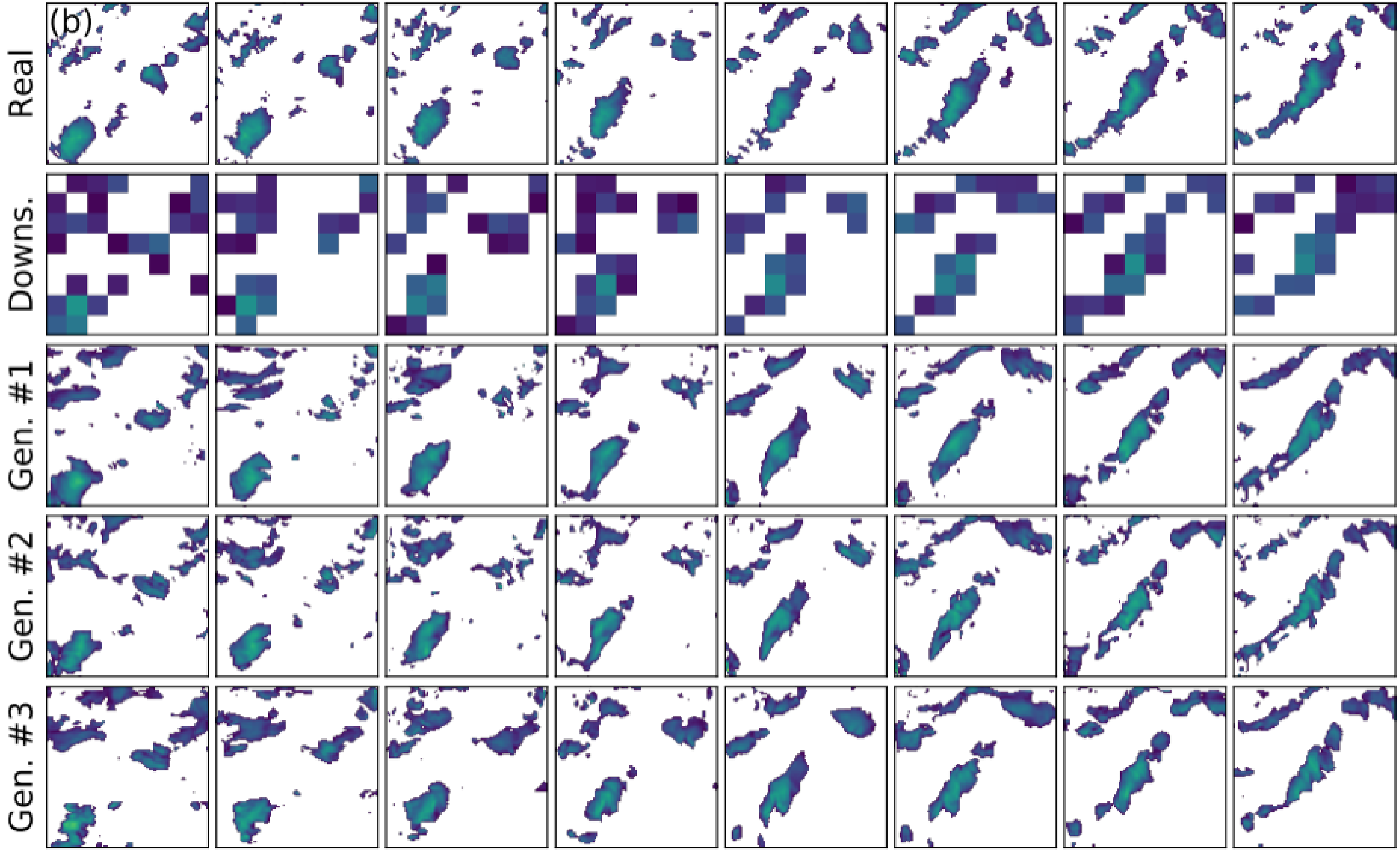
Modeling uncertainty in cloud profile

Stochastic downscaling of precipitation data

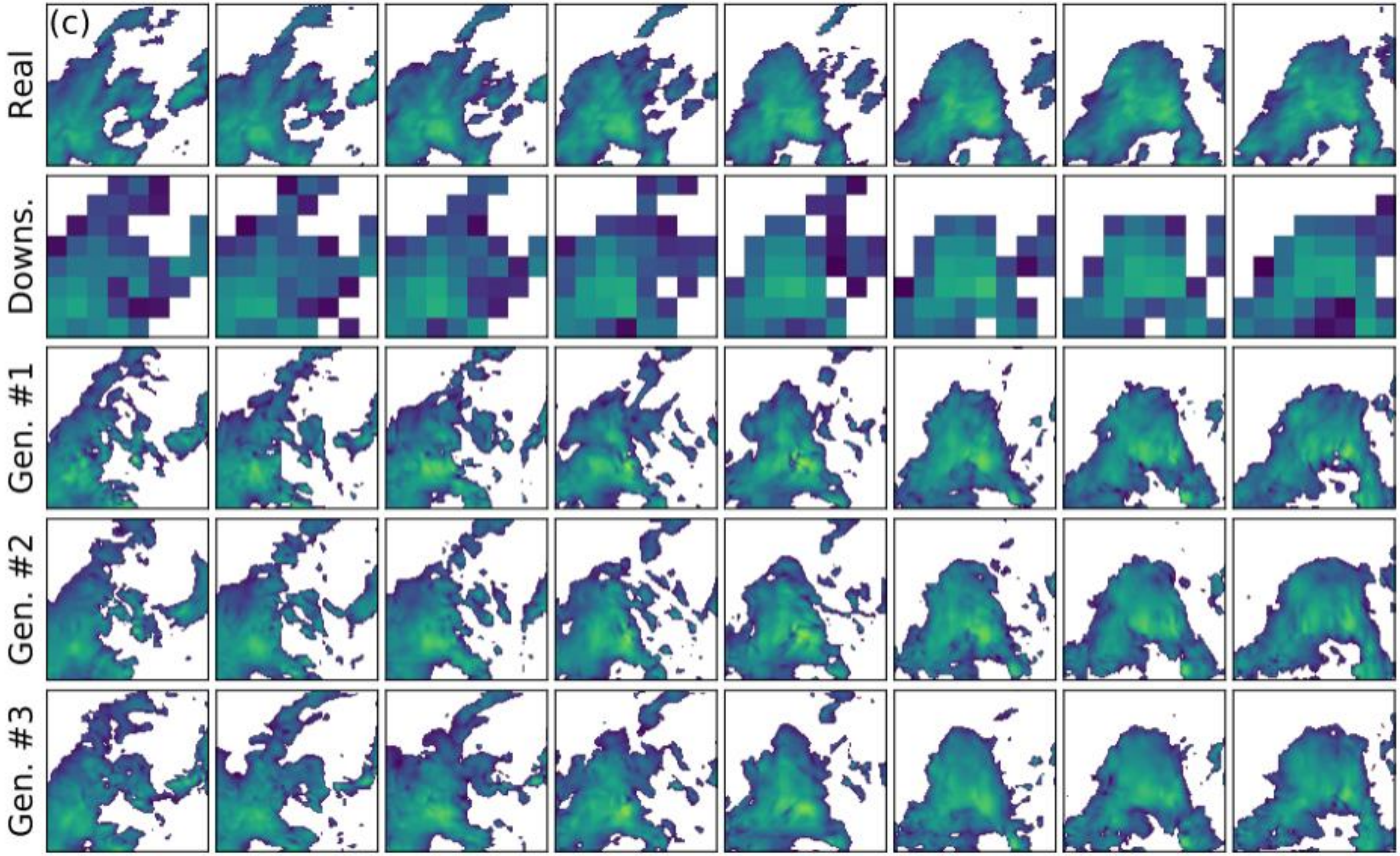
- Objectives:
  - Demonstrate stochastic downscaling with GANs (i.e., generate high-resolution fields from low-resolution inputs)
  - Generate realistic fields
  - Use the non-deterministic aspect of GANs to model the uncertainty
  - Model the time evolution of fields consistently
    - ➢ We need a recurrent generator

Stochastic downscaling of precipitation data

Leinonen et al. 2020

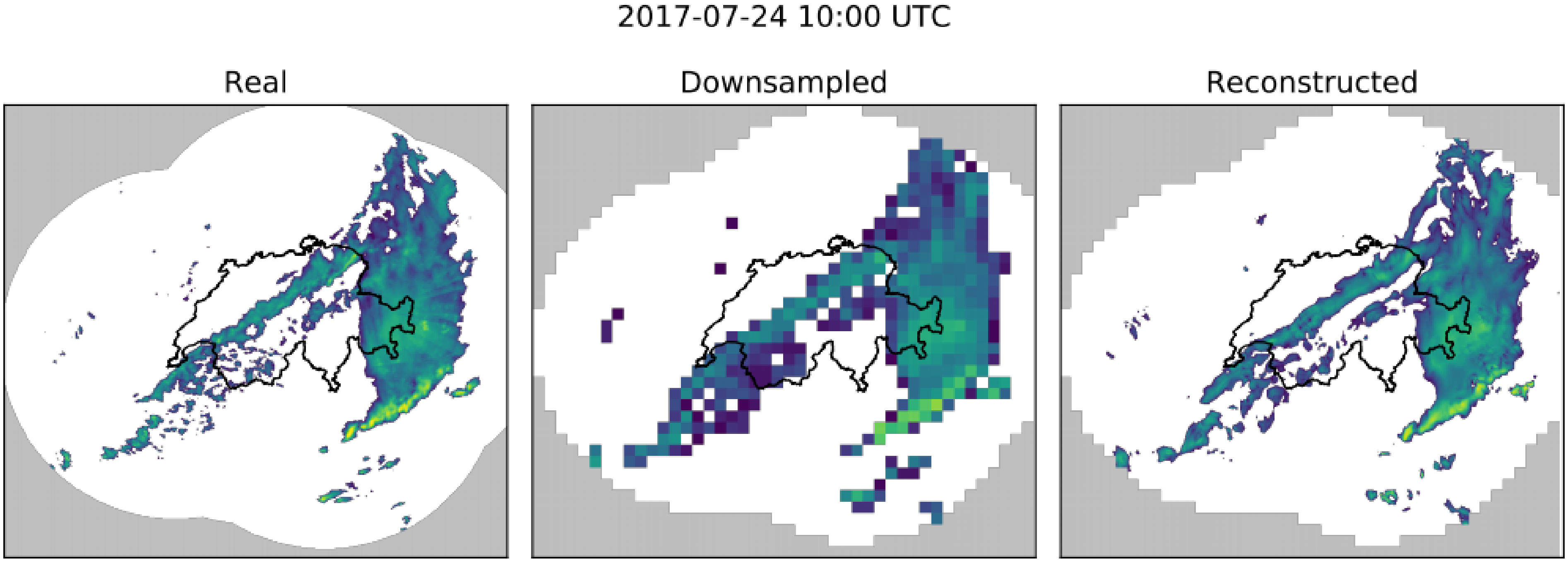


Stochastic downscaling of precipitation data



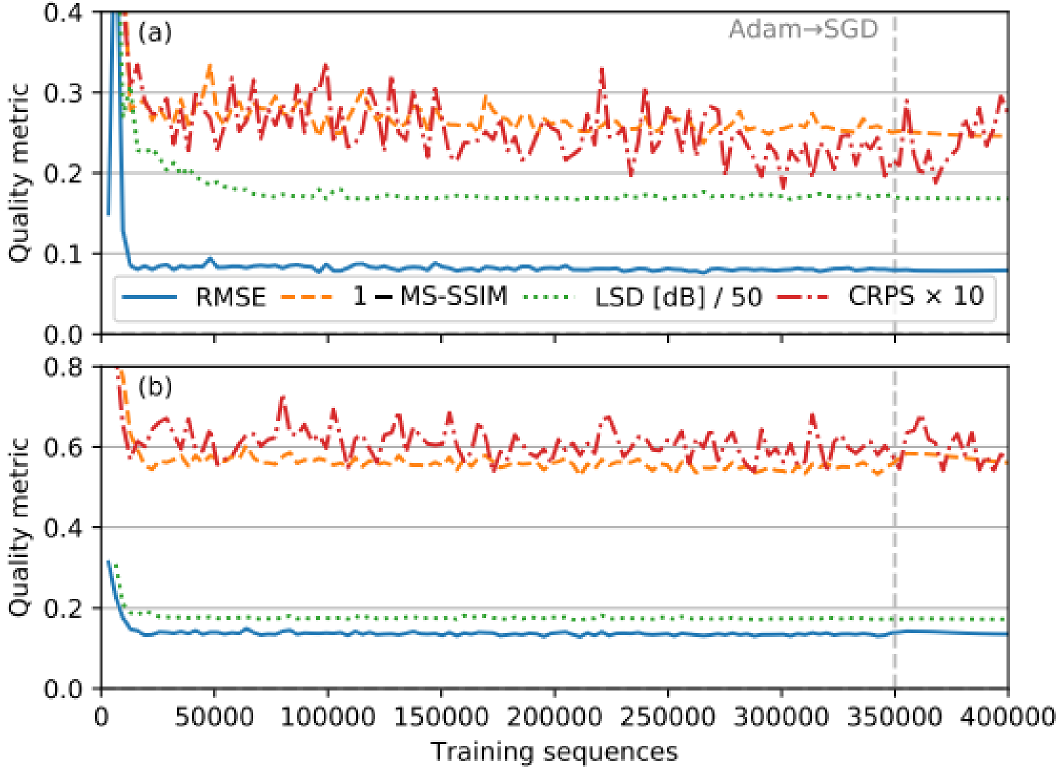
Stochastic downscaling of precipitation data

Fully convolutional generator:
can be applied to larger images after training



Evaluation: Image quality metrics

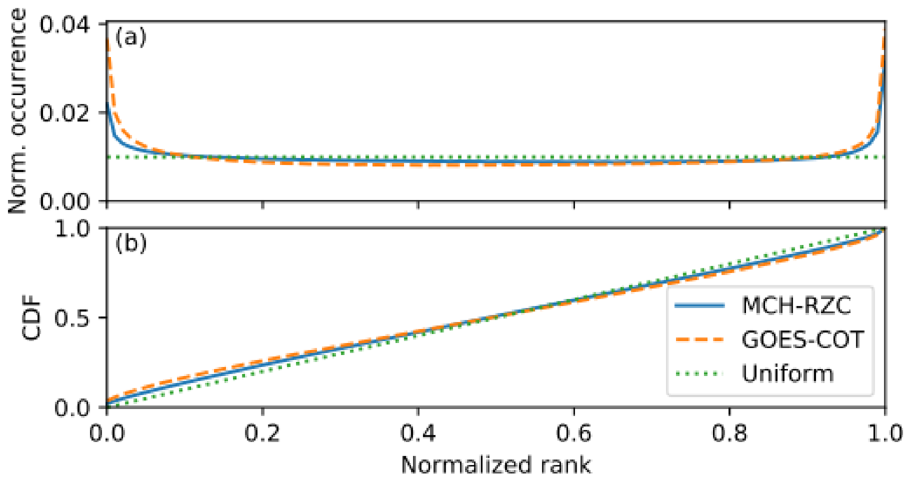
- Single-image quality metrics don't tell us very much
  - GAN isn't trained to optimize them
- CRPS is an ensemble metric that uses multiple predictions, works better



Rank statistics

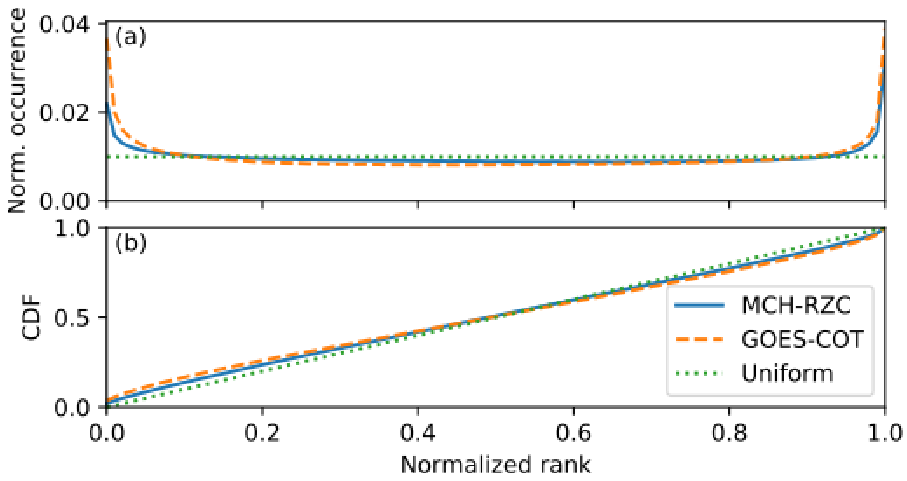
Does the distribution of values generated by the GAN match that of the observations?

- We don't know the true conditional distribution



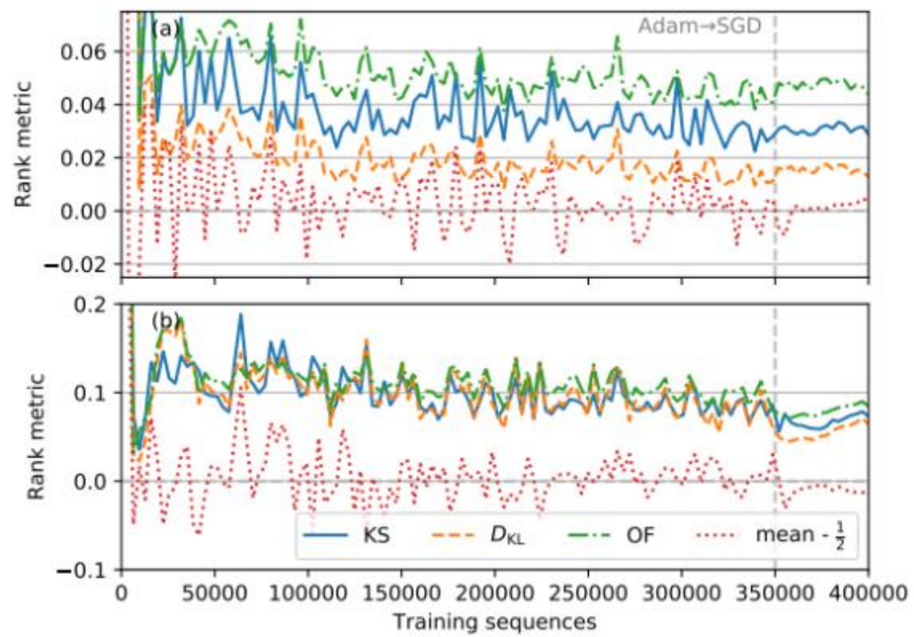
Rank statistics

- Compute the *rank* of the real observation in the ensemble, normalize to 0..1
- If uncertainty is modeled perfectly, the rank should be *uniformly distributed*
  - We can use metrics of similarity to the uniform distribution to evaluate whether the GAN is generating the right amount of uncertainty



Rank statistics

- Rank metrics converge for longer than image quality metrics

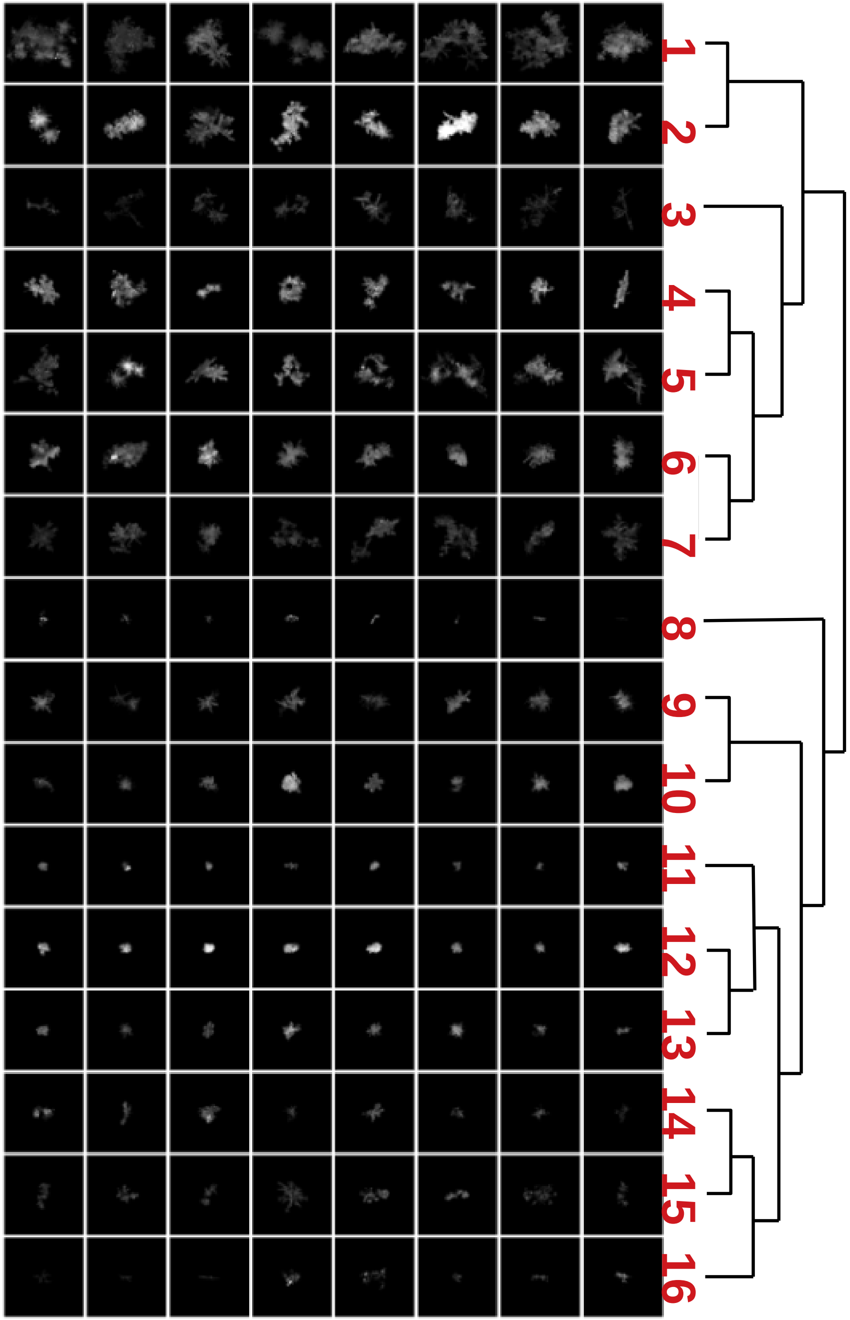


Other studies using GANs

- Snowflake classification
- Rainfall disaggregation
- Generating global climate data fields
- Downscaling of global climate model data

Other studies us

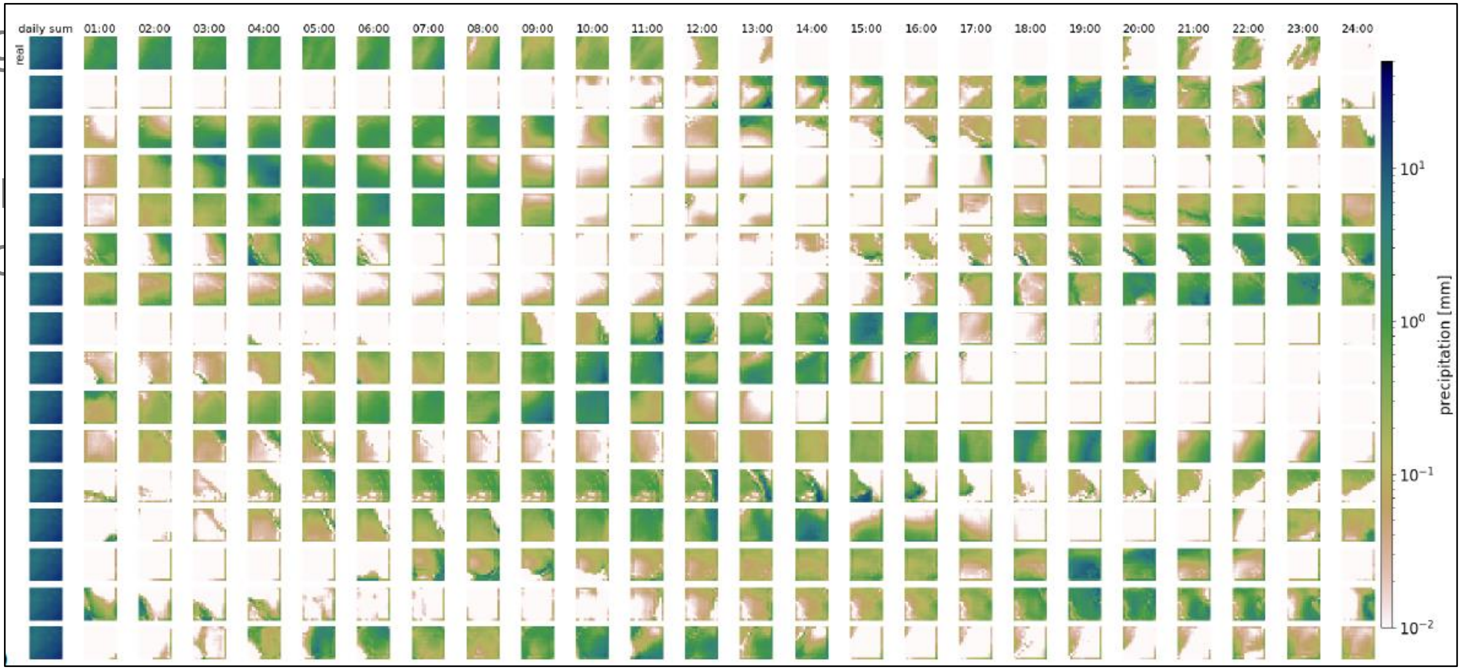
- Snowflake classification
- Rainfall disaggregation
- Generating global climate data fields
- Downscaling of global climate model data



Leinonen et al. 2020, AMT
https://doi.org/10.5194/amt-13-2949-2020

Other studies using GANs

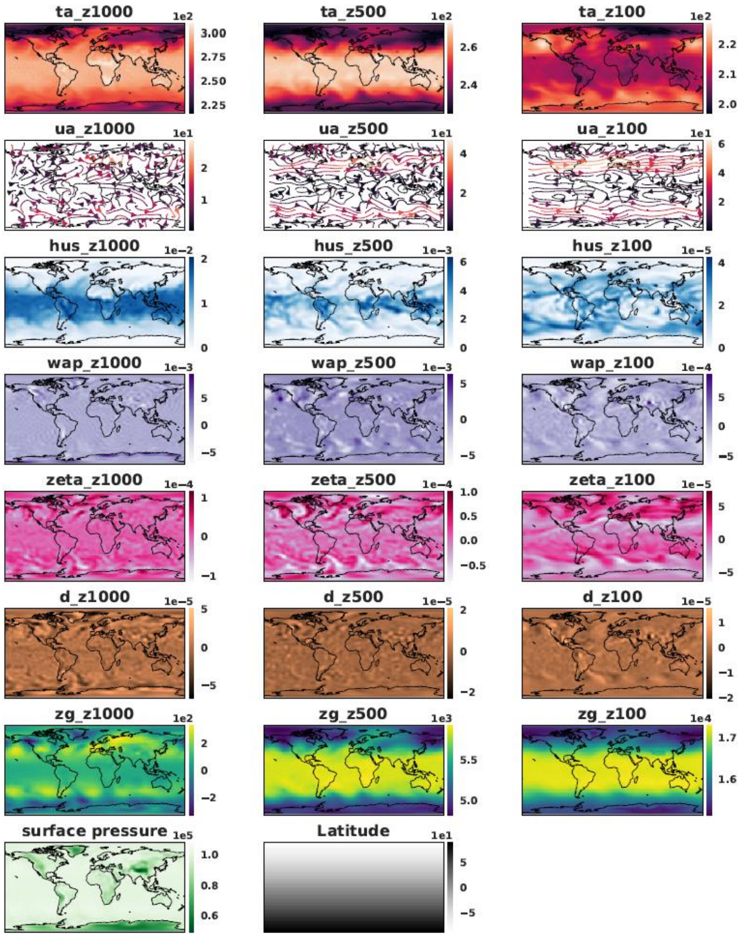
- Snowflake classification
- Rainfall disaggregation
- Generatin[g]
  data fields
- Downscali[ng]
  climate m[o]



Scher and Peßenteiner 2021, HESS
https://doi.org/10.5194/hess-2020-464

Other studies using GANs

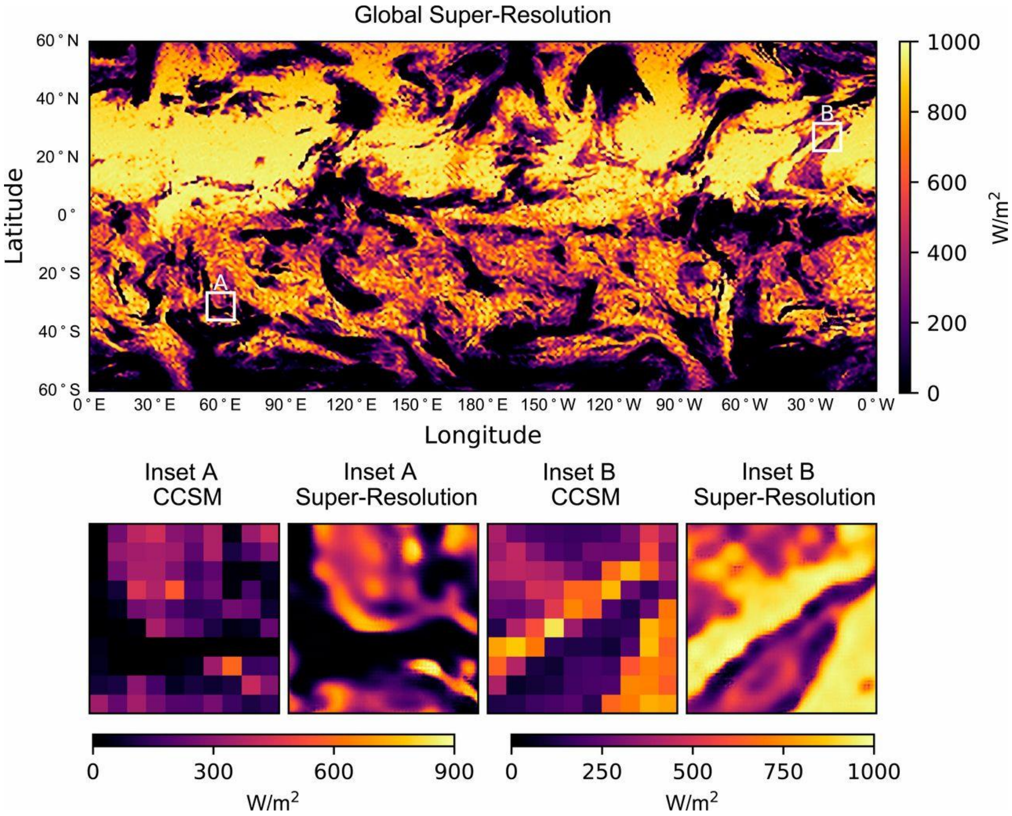
- Snowflake classification
- Rainfall disaggregation
- Generating global climate data fields
- Downscaling of global climate model data



Besombes et al. 2020, NPG
https://doi.org/10.5194/npg-2021-6

Other studies using GANs

- Snowflake classification
- Rainfall disaggregation
- Generating global climate data fields
- Downscaling of global climate model data



Stengel et al. 2020, PNAS
https://doi.org/10.1073/pnas.1918964117

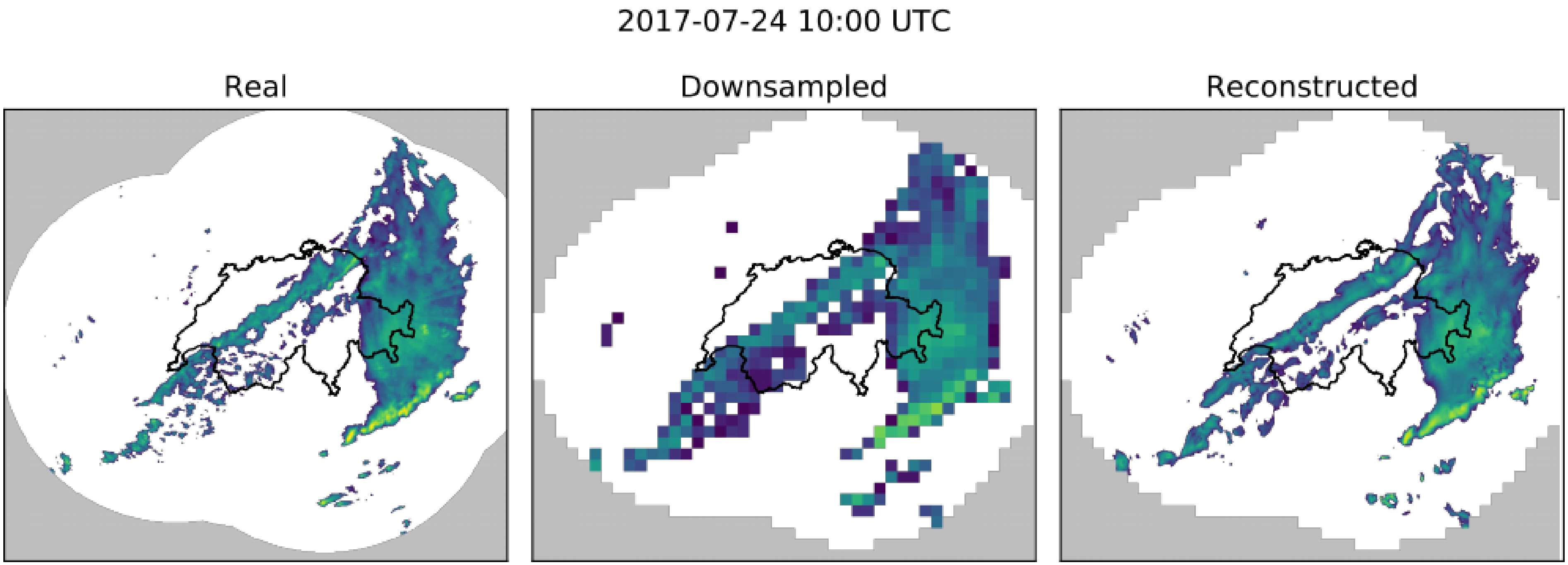
“Should I consider GANs for my project?”

- GANs (and CGANs in particular) seem a natural fit for many Earth science data problems
  - Consider CGANs if you need realistic spatial structure and/or uncertainty modeling
  - GANs can also do unsupervised data discovery
  - Many low-hanging fruits still available to pick!
  - But tricky to work with, needs cost-benefit evaluation
- GANs model uncertainty through sample diversity
  - Ensemble forecasters have the same mindset
  - Methods from ensemble forecasting can be applied to GANs

Questions?

Interested in discussing GANs in Weather/EO/Climate applications?

Email: jussi.leinonen@meteoswiss.ch, Twitter: @jsleinonen



https://www.youtube.com/watch?v=3OS6hz8gYC8