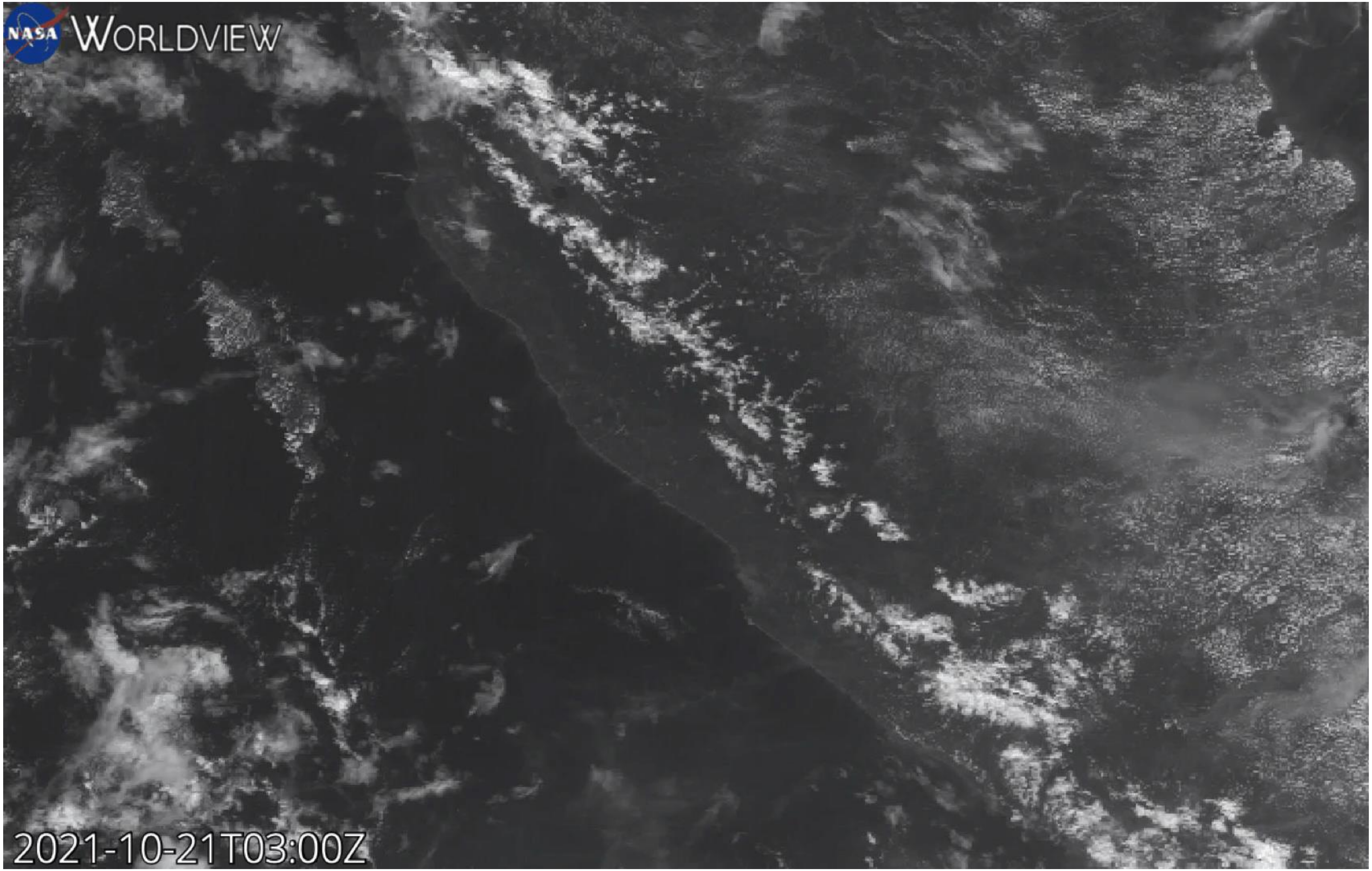
Combining data assimilation and machine learning to extract more information from earth observations

Alan Geer

Presentation at **ESA-ECMWF workshop**
Machine Learning for Earth System Observation and Prediction
15-18 November, 2021



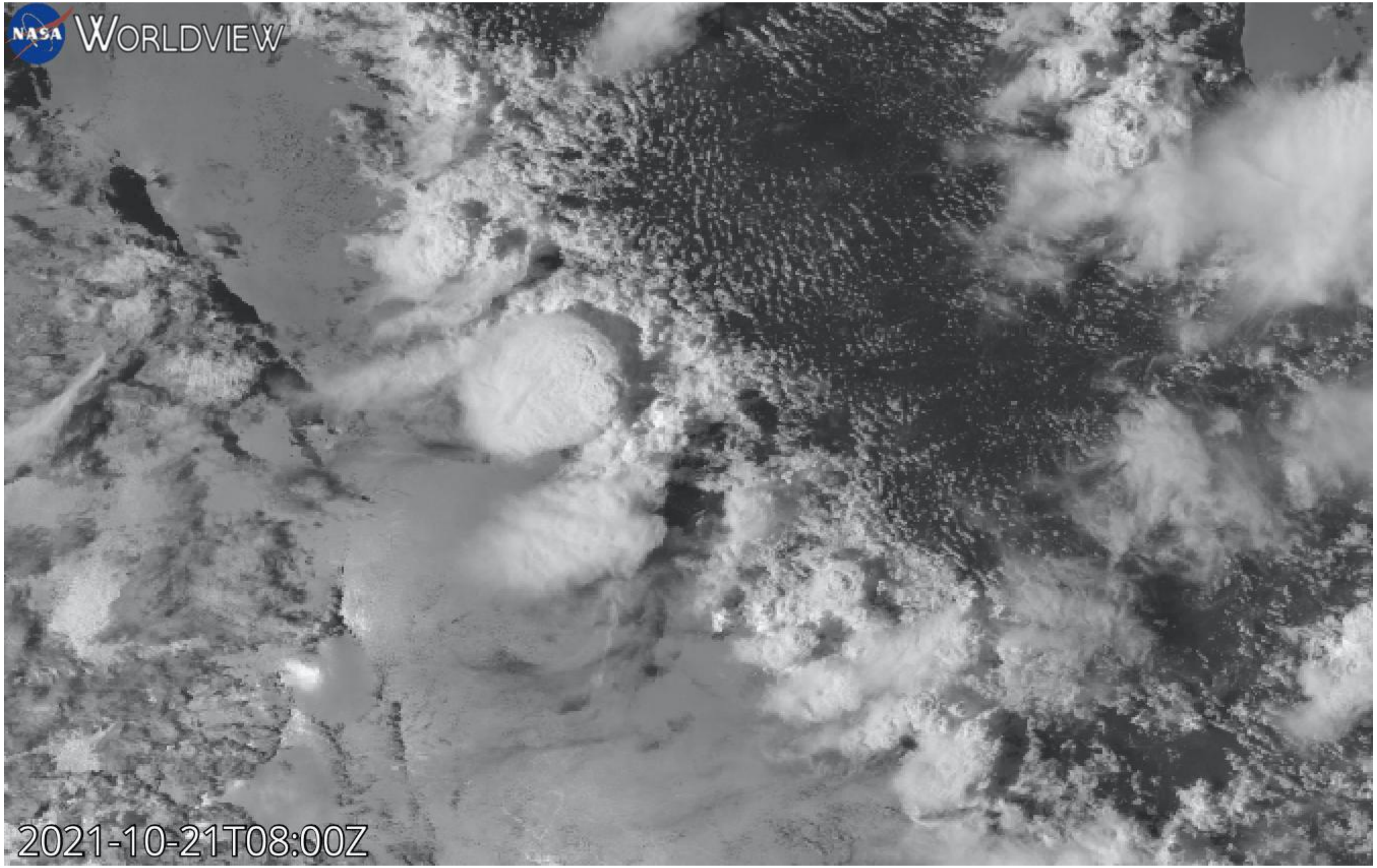
Southern Sumatra 11-17 local solar time

JAXA Himawari 8 observations

Resolution:
10 minute
0.5 km

Visible band 3, generated with NASA worldview

Challenge of using observations



Approx.
450km

Southern Sumatra 16:00 local solar time

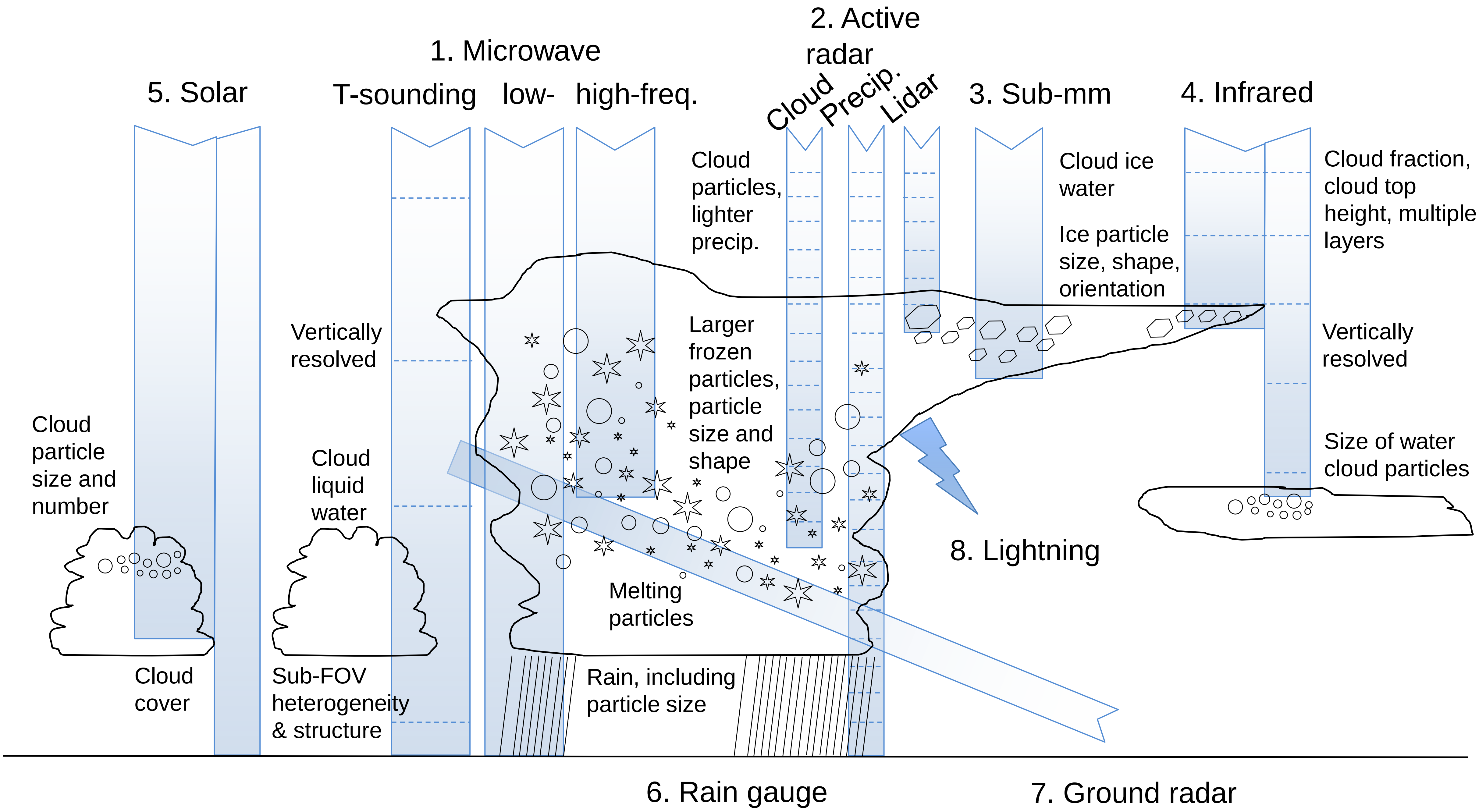
JAXA Himawari 8 observations

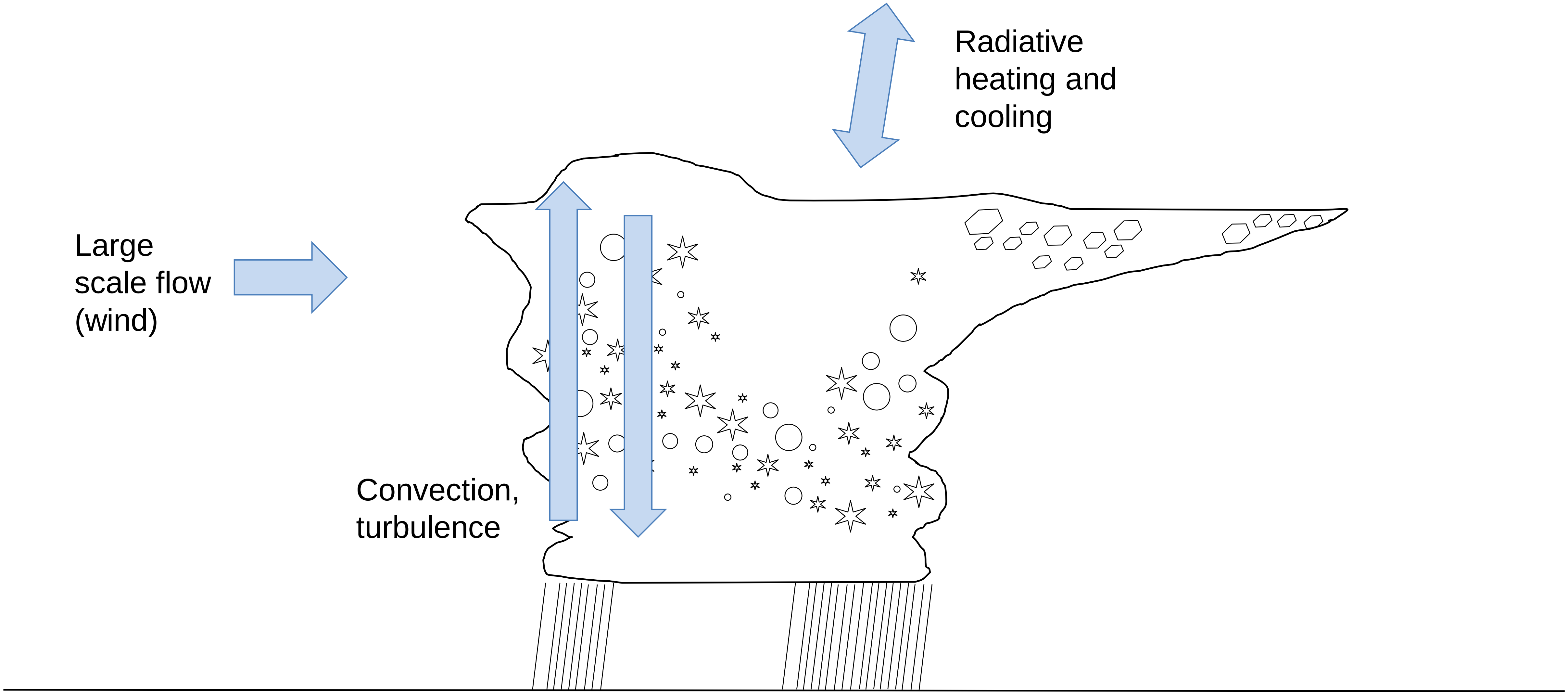
Resolution:

10 minute

0.5 km

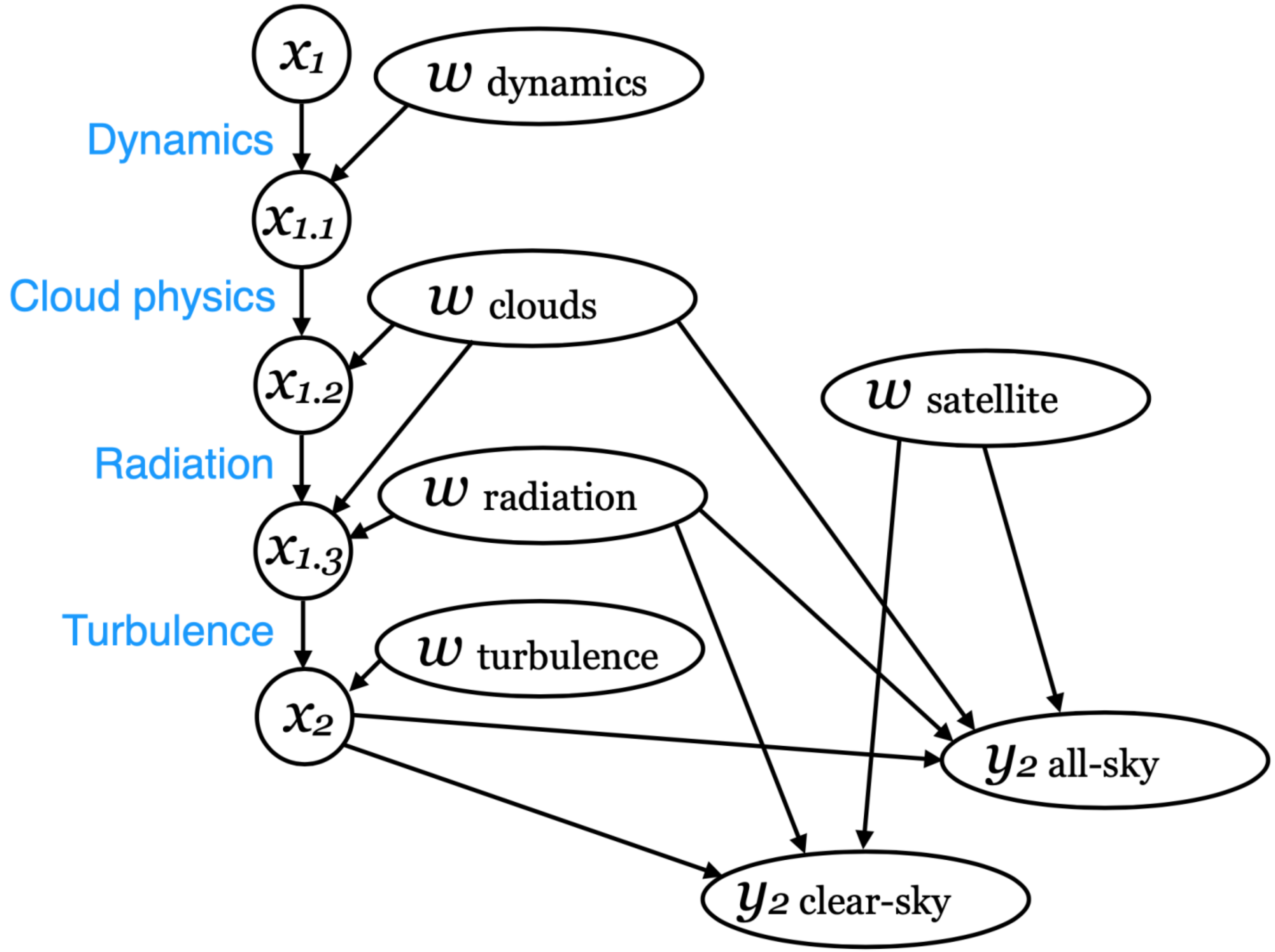
Visible band 3, generated with NASA worldview


1. Microwave
T-sounding
low-
high-freq.
2. Active radar
Cloud
Precip.
Lidar
3. Sub-mm
4. Infrared
5. Solar
Cloud particles, lighter precip.
Cloud ice water
Ice particle size, shape, orientation
Cloud fraction, cloud top height, multiple layers
Vertically resolved
Larger frozen particles, particle size and shape
Vertically resolved
Size of water cloud particles
Cloud particle size and number
Cloud liquid water
8. Lightning
Melting particles
Cloud cover
Sub-FOV heterogeneity & structure
Rain, including particle size
6. Rain gauge
7. Ground radar


Radiative heating and cooling
Large scale flow (wind)
Convection, turbulence

Physical forecast models in a data assimilation framework

One model timestep x_1 to x_2



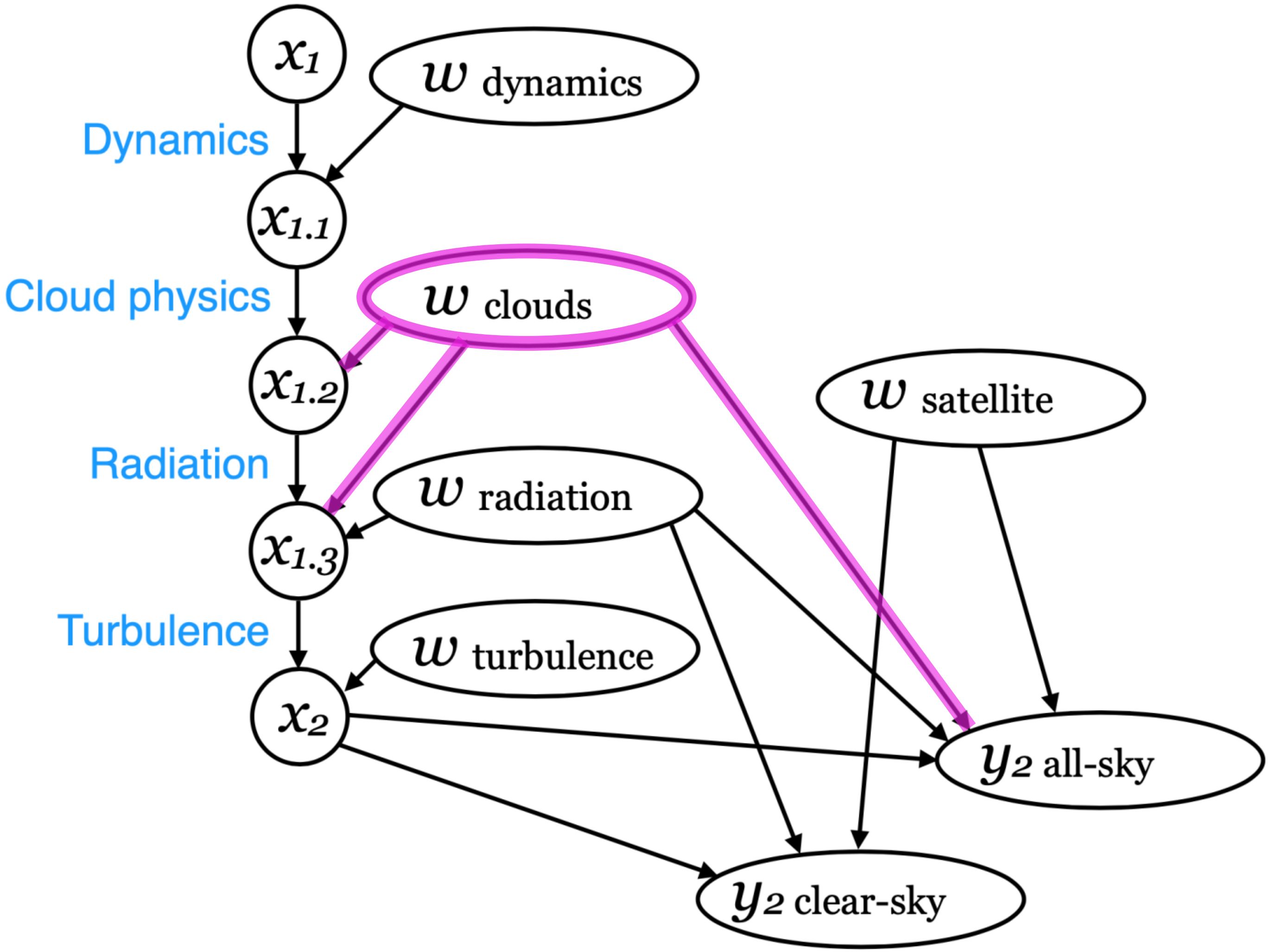
x geophysical state

w parameters and structure of models

y observations

Train a new cloud model inside a data assimilation system?

One model timestep x_1 to x_2



x geophysical state

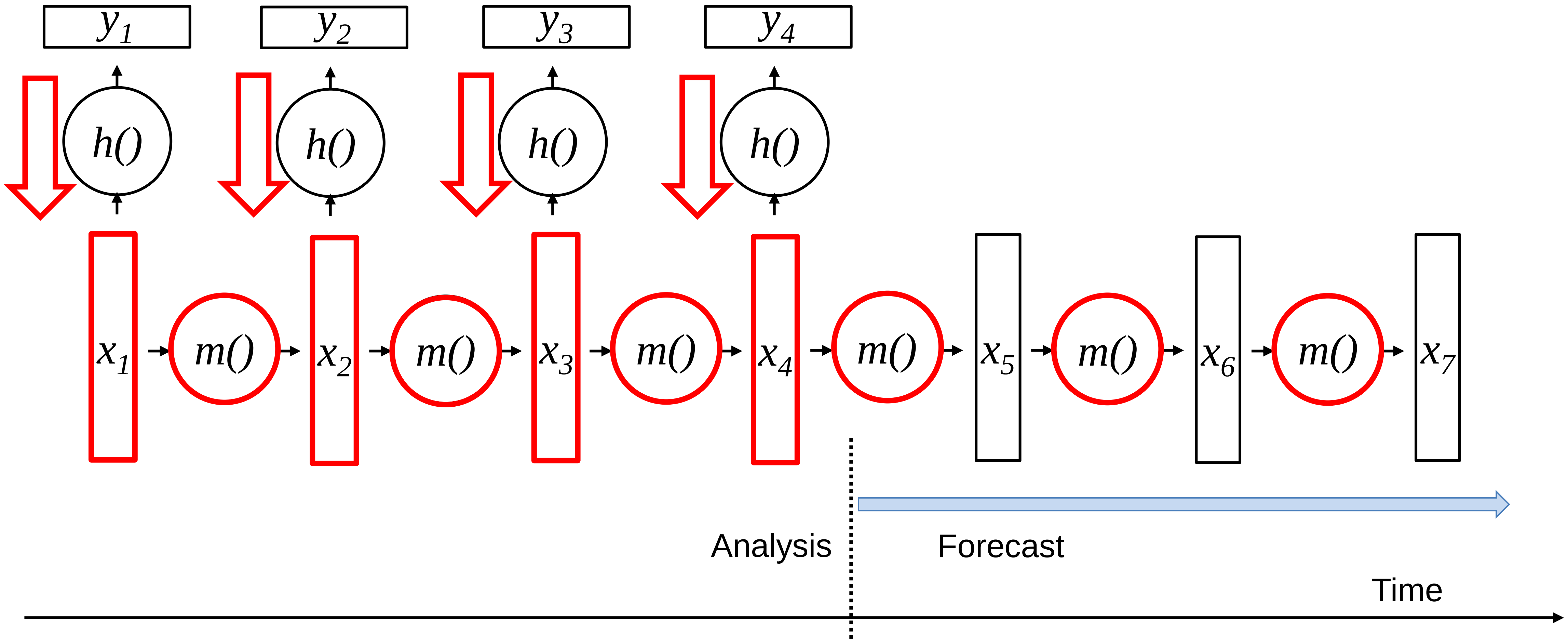
w parameters and structure of models

y observations

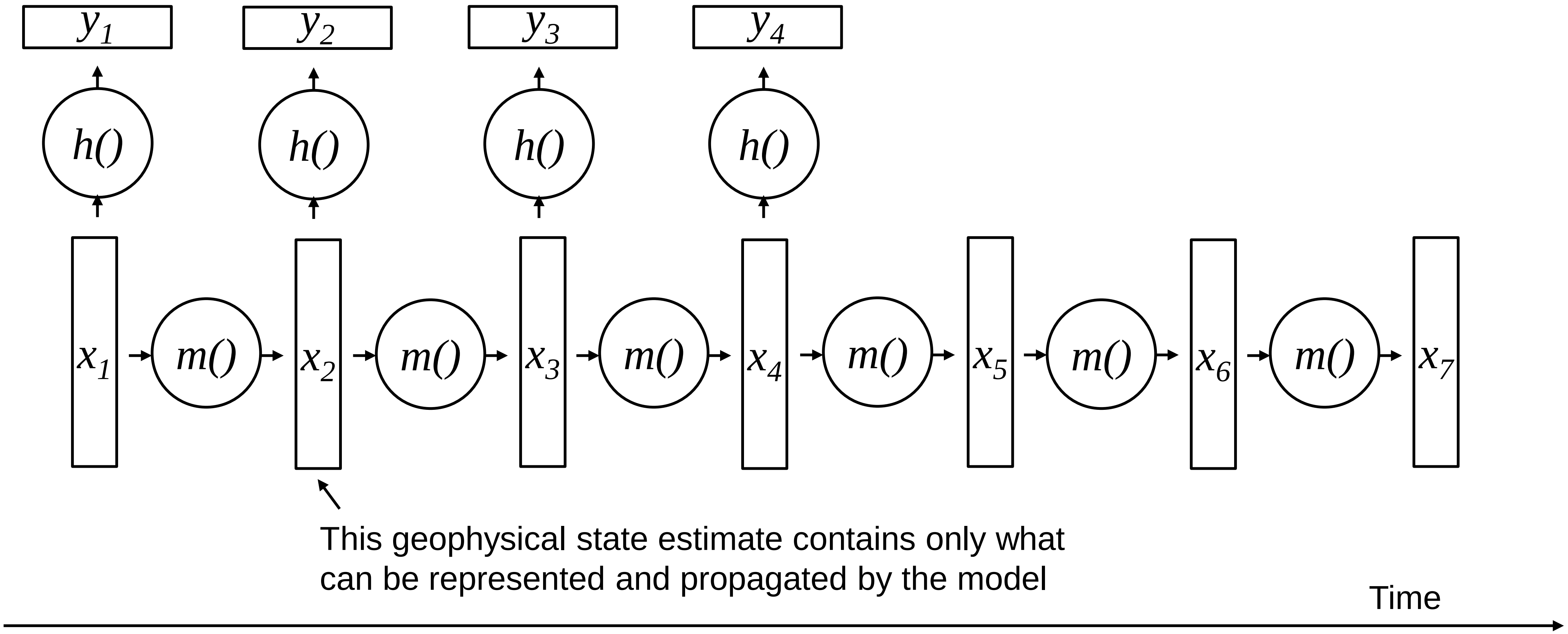
Data assimilation

h() observation operator

m() geophysical model



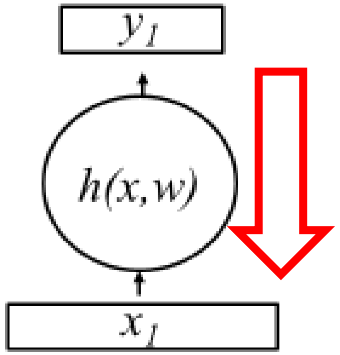
Data assimilation: importance of the model



Cost function for variational DA

Start from Bayes theorem
Assume Gaussian errors (error standard deviation σ)
and for clarity here simplify to scalar variables
and ignore any covariance between observation, model or state error

$$y = h(x, w)$$



$$J(x,w) = \underbrace{\frac{(y - h(x,w))^2}{(\sigma^y)^2}}_{J^y} + \underbrace{\frac{(x^b - x)^2}{(\sigma^x)^2}}_{J^x} + \underbrace{\frac{(w^b - w)^2}{(\sigma^w)^2}}_{J^w}$$

DA	Cost function	Observation term	Prior knowledge of state	Prior knowledge of model

Cost / loss function equivalence of ML and variational DA

Start from Bayes theorem
Assume Gaussian errors (error standard deviation σ)
and for clarity here simplify to scalar variables
and ignore any covariance between observation, model or state error

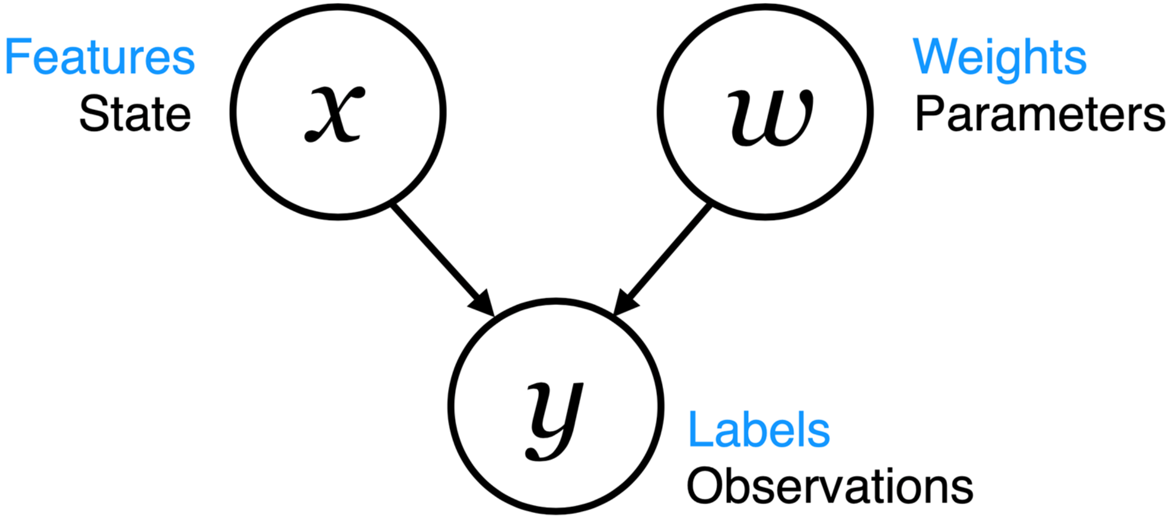
ML	Loss function	Basic loss function	Feature error?	Weights regularisation

$$J(x,w) = \underbrace{\frac{(y - h(x,w))^2}{(\sigma^y)^2}}_{J^y} + \underbrace{\frac{(x^b - x)^2}{(\sigma^x)^2}}_{J^x} + \underbrace{\frac{(w^b - w)^2}{(\sigma^w)^2}}_{J^w}$$

DA	Cost function	Observation term	Prior knowledge of state	Prior knowledge of model

Machine learning (e.g. NN)		Variational data assimilation	
Labels	y	Observations	$\mathrm{y^o}$
Features	x	State	x
Neural network or other learned models	$\mathrm{y}' = W(\mathrm{x})$	Physical forward model	$\mathrm{y} = H(\mathrm{x})$
Objective or loss function	$(\mathrm{y} - \mathrm{y}')^2$	Cost function	$J = J^b + \left(\mathrm{y^o} - H(\mathrm{x})\right)^T \mathrm{R}^{-1} \left(\mathrm{y^o} - H(\mathrm{x})\right)$
Regularisation	$\lVert \mathrm{w} \rVert$	Background term	$J^b = \left(\mathrm{x} - \mathrm{x^b}\right)^T \mathrm{B}^{-1} \left(\mathrm{x} - \mathrm{x^b}\right)$
Iterative gradient descent		Conjugate gradient method (e.g.)	
Back propagation		Adjoint model	$\frac{\partial J}{\partial \mathrm{x}} = \mathrm{H}^T \frac{\partial J}{\partial \mathrm{y}}$
Train model and then apply it		Optimise state in an update-forecast cycle	

Equivalence of ML and DA

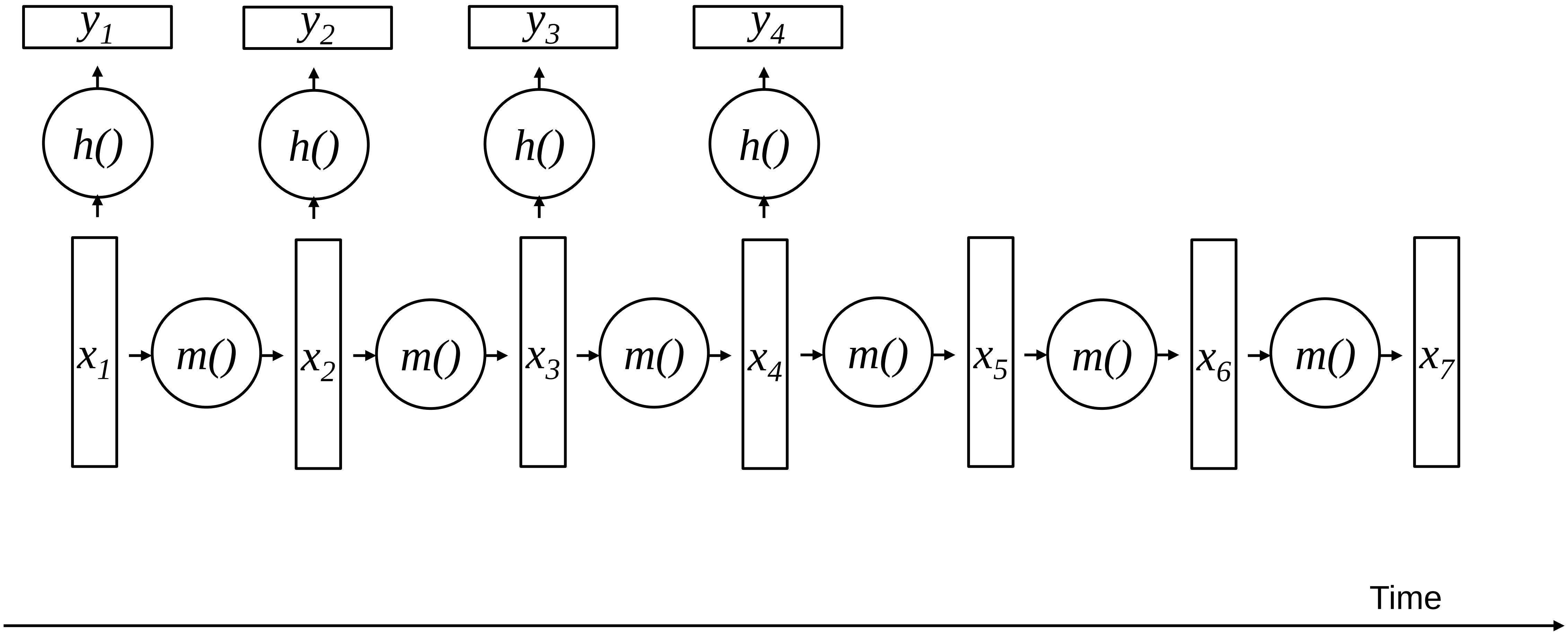


$$y = h(x, w)$$

As a Bayesian network

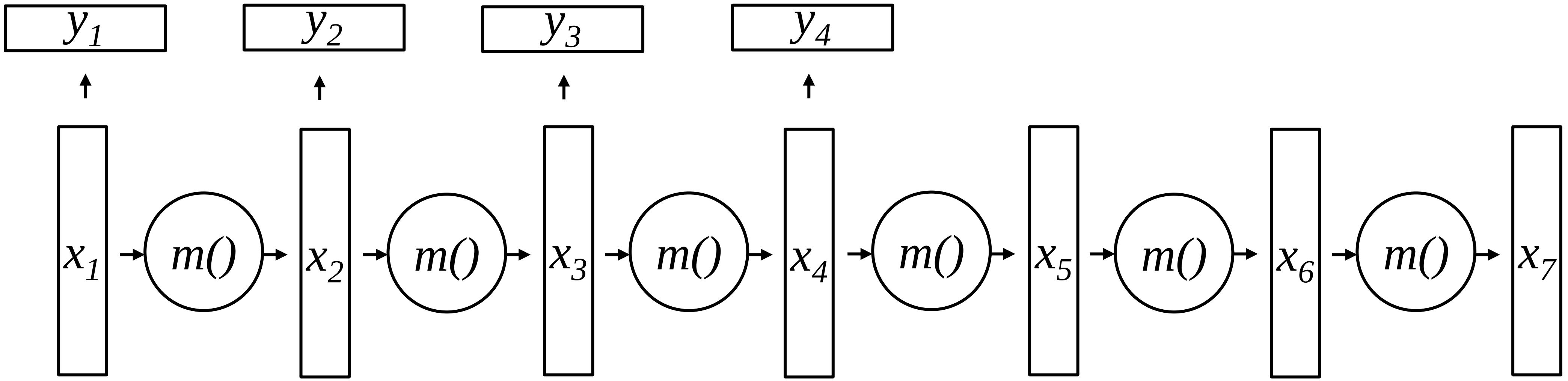
- Hsieh and Tang (1998) https://doi.org/10.1175/1520-0477(1998)079%3C1855:ANNMTP%3E2.0.CO;2
- Abarbanel et al. (2018) https://doi.org/10.1162/neco_a_01094
- Bocquet et al. (2020) https://arxiv.org/abs/2001.06270
- Geer (2021) https://doi.org/10.21957/7fyj2811r
- Bayesian basis of ML: Goodfellow et al. (2016) https://www.deeplearningbook.org
- ML – DA merger: see Rosella Arcucci's talk this workshop

Data assimilation

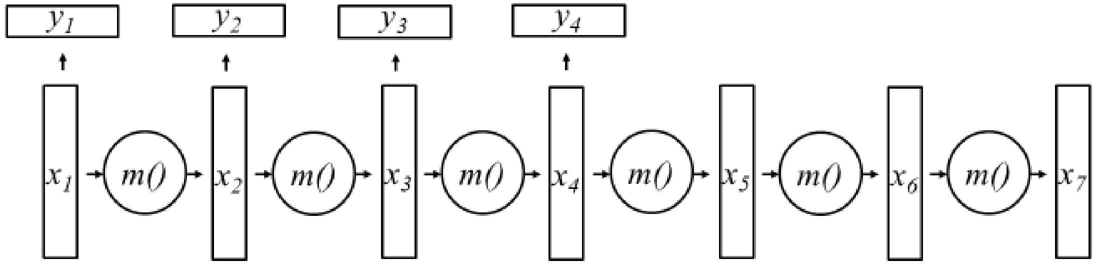


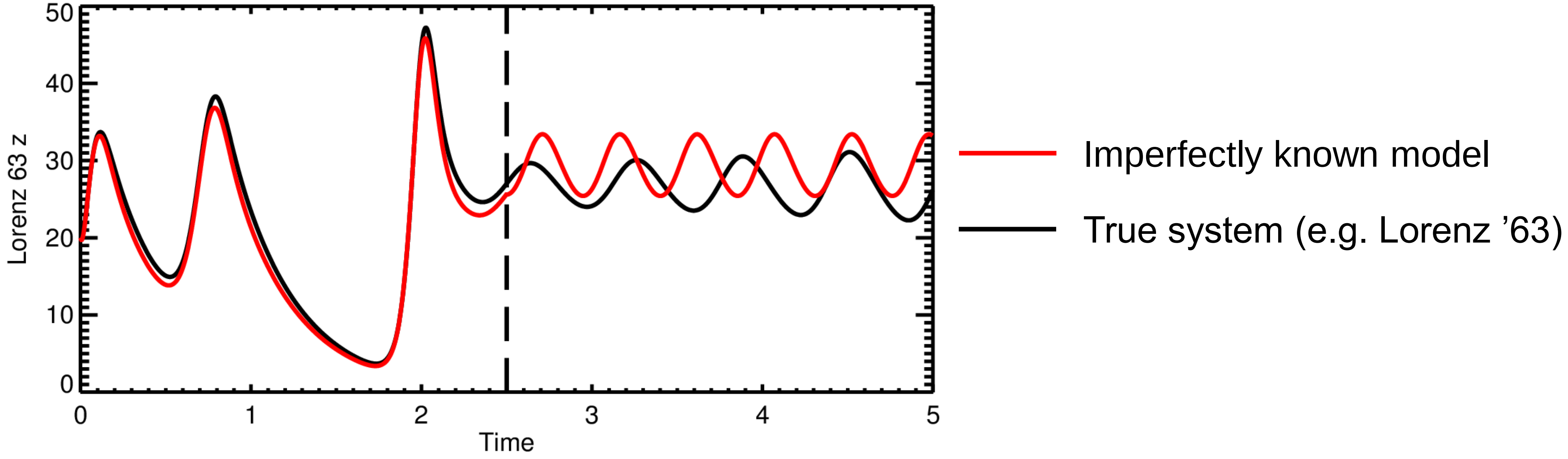
Data assimilation – ignoring the complexity of observations

Assume for the moment that ingesting information from real observations is easy!



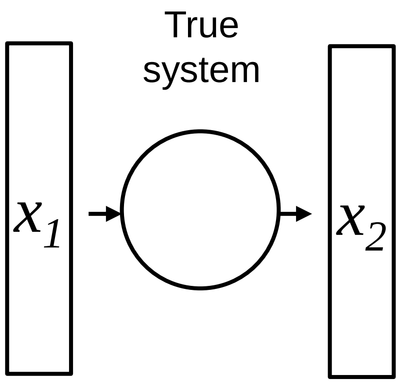
Data assimilation ↔ dynamical systems, recurrent neural networks, etc.

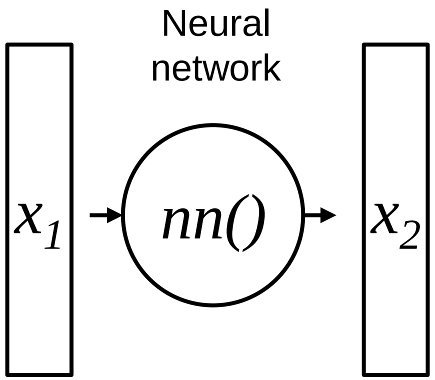




synchronise (e.g. DA) predict

How can machine learning help? No need for a physical model?

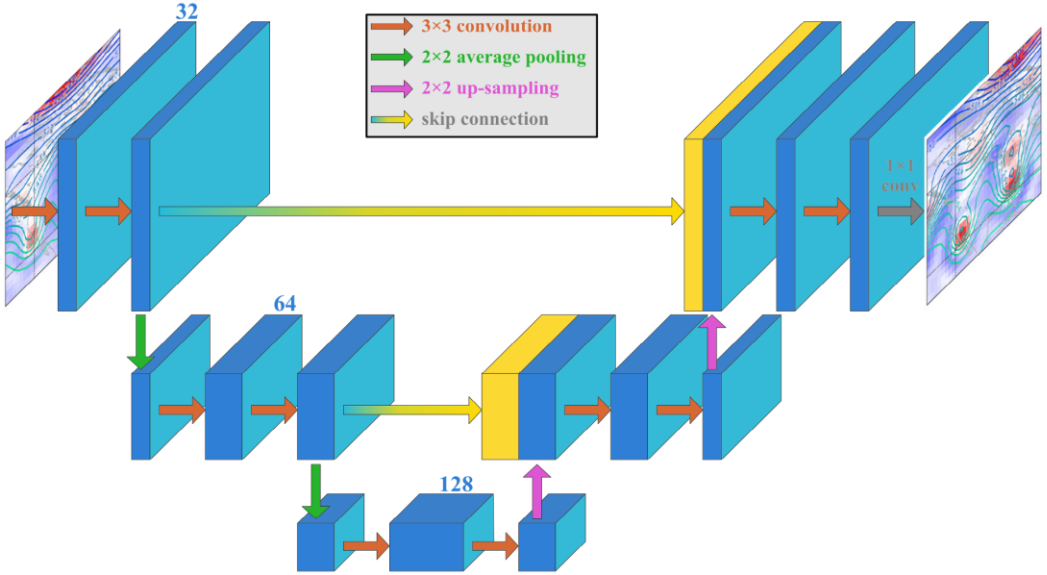




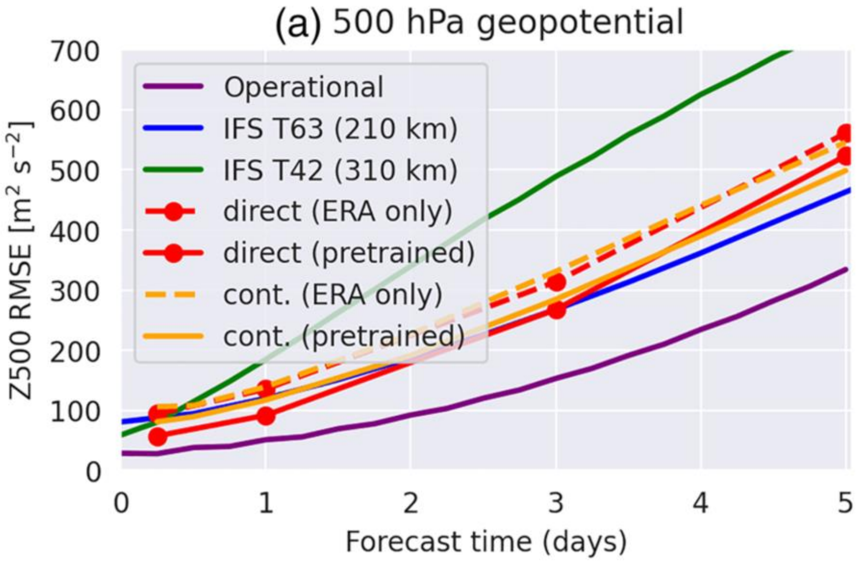
Techniques applied to generally simpler dynamical systems

E.g. recurrent neural networks (RNN, e.g. echo state networks, reservoir computing – e.g. Pathak et al., 2018, https:/doi.org/10.1103/PhysRevLett.120.024102),

Low-resolution data-driven weather forecasting: Weatherbench challenge



E.g. U-Net convolutional neural networks CNNs - Weyn et al. (2020, https:// doi.org/10.1029/2020MS002109)



E.g. resnet approach – Rasp and Thuerey (2021, https://doi. org/10.1029/2020MS002405)

Combine physical and empirical models: Physically constrained ML

```
def net_u(self, x, t):
    u = self.neural_net(tf.concat([x,t],1), self.weights, self.biases)
    return u

def net_f(self, x,t):
    u = self.net_u(x,t)
    u_t = tf.gradients(u, t)[0]
    u_x = tf.gradients(u, x)[0]
    u_xx = tf.gradients(u_x, x)[0]
    f = u_t + u*u_x - self.nu*u_xx

    return f

self.loss = tf.reduce_mean(tf.square(self.u_tf - self.u_pred)) + \
            tf.reduce_mean(tf.square(self.f_pred))
```

Neural network

Gradients of the network

Burger's equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0$$

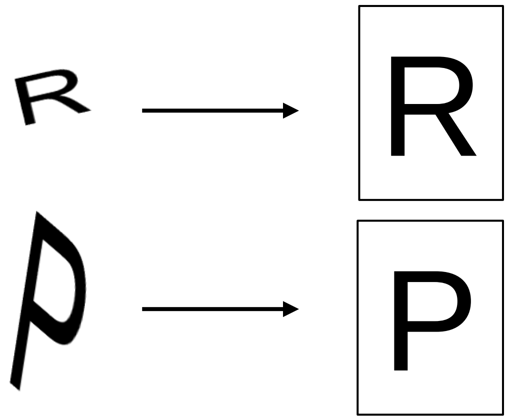
Custom loss function

https://github.com/maziarraissi/PINNs

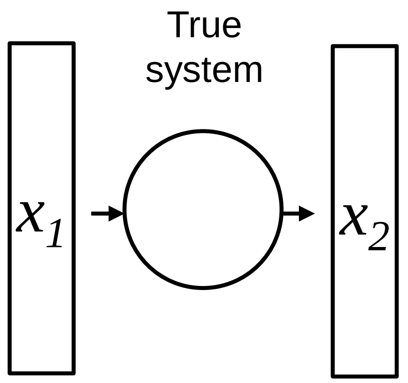
Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations." arXiv preprint arXiv:1711.10561 (2017)

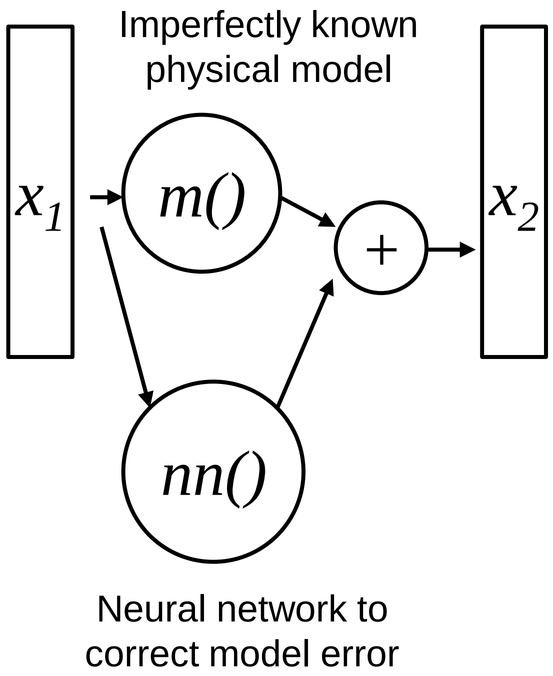
Combine physical and empirical models: semi-physical components in empirical models

- E.g. spatial transformers used in U-Net in Weatherbench framework (Chattopadhyay et al. , 2021, GMDD, https://doi.org/10.5194/gmd-2021-71)
  - Apply a transformation matrix and an interpolation that allows e.g. rotation and scaling (of latent space)
  - Original work on spatial transformers in image processing, e.g. character recognition (Jaderberg et al., 2015, https://arxiv.org/abs/1506.02025)
  - See also fluid dynamics example: Wang et al. (2021, Incorporating symmetry into deep dynamics models for improved generalization, https://arxiv.org/abs/2002.03061)

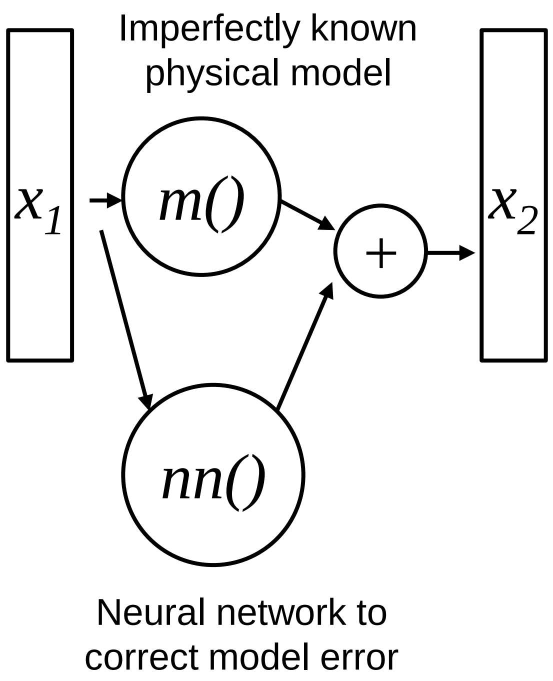


Combine physical and empirical models: error correction

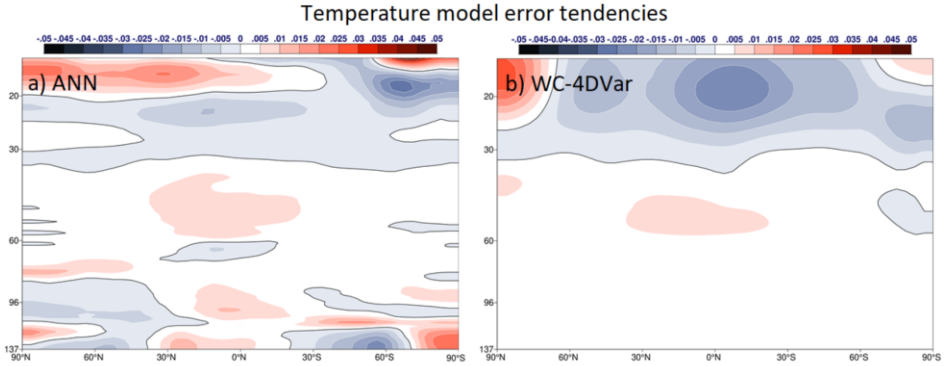




Combine physical and empirical models: error correction



- Simpler models: e.g. Lorenz ’63, ’96, QG:
  - Pathak et al. (2018, https://doi.org/10.1063/1.5028373)
  - Use iterative cycles of data assimilation followed by neural network training (Brajard et al., 2020, https://doi.org/10.1016/j.jocs.2020.101171)
- Applied to an operational NWP model:



Bonavita and Laloyaux, 2020, https://doi.org/10.1029/2020MS002232

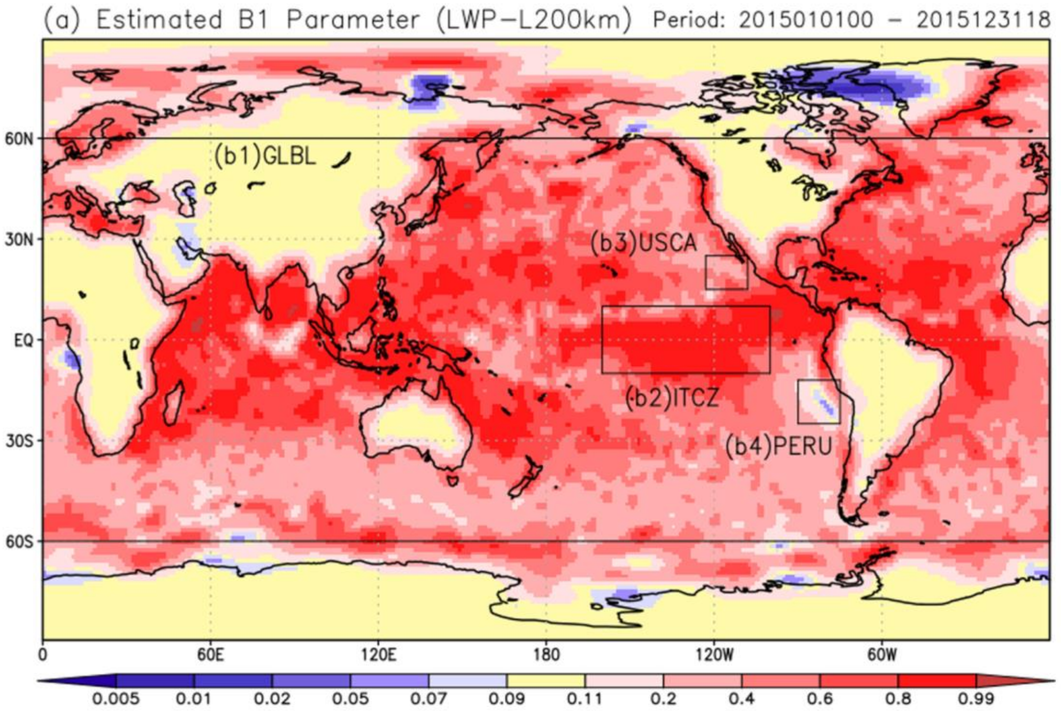
- See talks by Alban Farchi and Marcin Chrust at this workshop

Combine physical and empirical models: parameter estimation

- Parameter estimation in data assimilation
  - E.g. Kotsuki et al. (2020, https://doi.org/10.1029/2019JD031304) estimation of autoconversion parameter in atmospheric GCM
  - E.g. Tijana Janjic presentation in this workshop

$$J(x,w) = \underbrace{\frac{(y-h(x,w))^2}{(\sigma^y)^2}}_{J^y} + \underbrace{\frac{(x^b-x)^2}{(\sigma^x)^2}}_{J^x} + \underbrace{\frac{(w^b-w)^2}{(\sigma^w)^2}}_{J^w}$$

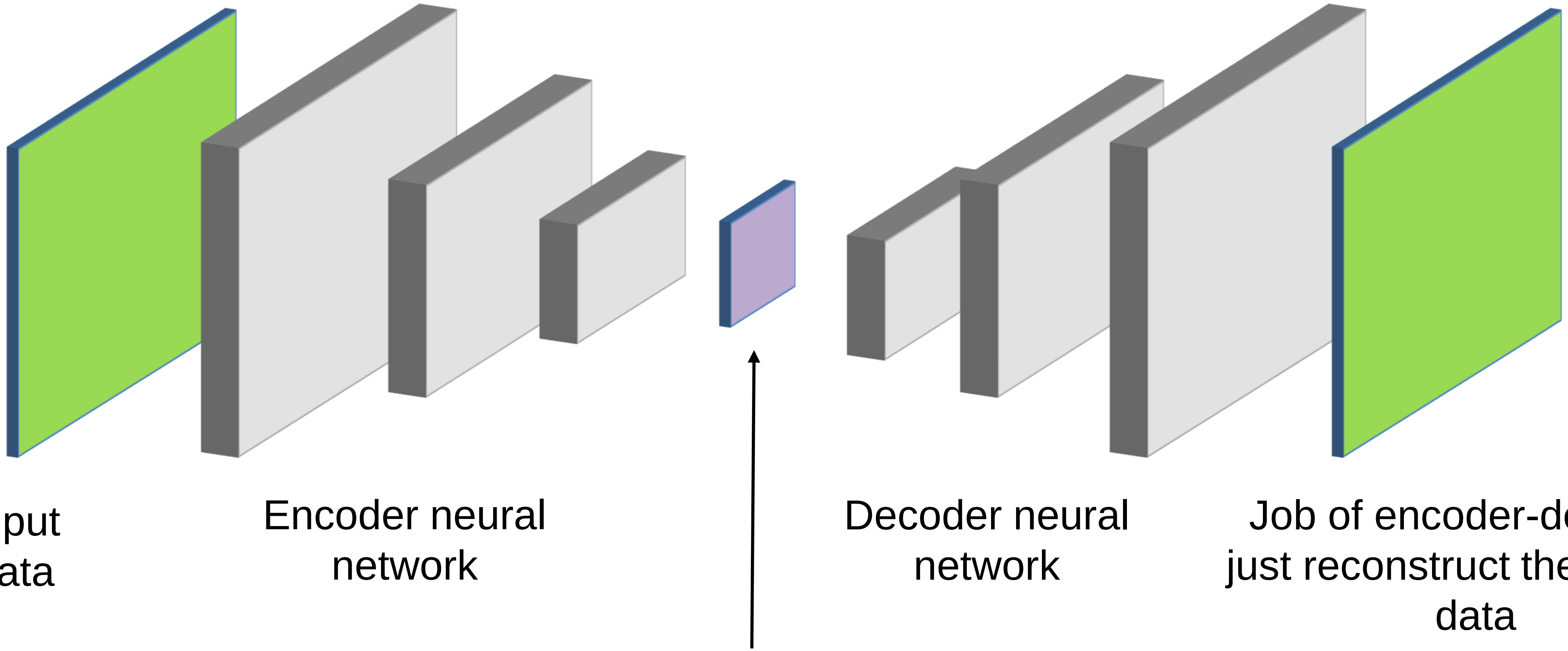
DA | Cost function | Observation term | Prior knowledge of state | Prior knowledge of model



Using ML to extend data assimilation capabilities

- In variational data assimilation:
  - Use machine learning emulators as an alternative numerical differentiation method to create tangent-linear (TL) and adjoint (AD) operators
    - e.g. Hatfield et al., 2021, https://doi.org/10.1029/2021MS002521, emulate a gravity wave drag scheme for use in TL and AD only
- In ensemble data assimilation
  - Use machine learning emulators to generate very large ensembles
    - E.g. Chattopadhyay et al. , 2021, GMDD, https://doi.org/10.5194/gmd-2021-71, generate a 1000-member ensemble
- Data assimilation in the latent space of an encoder-decoder
  - E.g. Amendola et al., 2020, Data assimilation in the latent space of a neural network, https://arxiv.org/abs/2012.12056
  - E.g. Peyron et al., 2021, Latent space data assimilation by using deep learning https://arxiv.org/abs/2104.00430
  - See talks by Rosella Arcucci and Sibo Cheng, this workshop

Latent space of the neural network – e.g. encoder - decoder



Input data

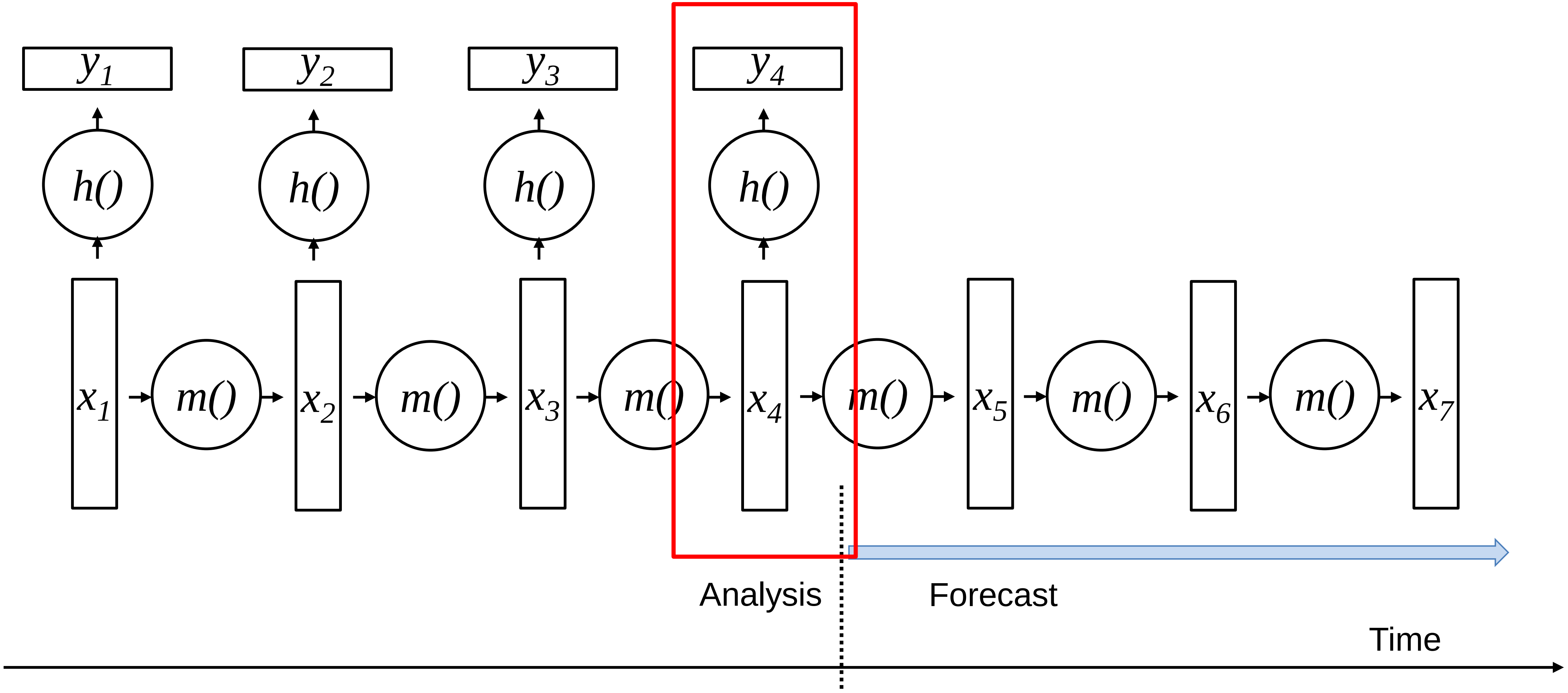
Encoder neural network

Decoder neural network

Job of encoder-decoder: just reconstruct the original data

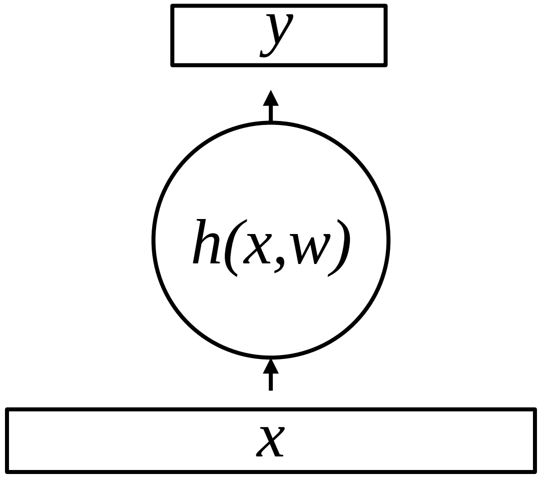
Hopefully, a compact representation of the input data

Data assimilation: now focusing on observations and geophysical variables



How can machine learning help? No physical model available

Observations



Well-constrained model variables from the DA system

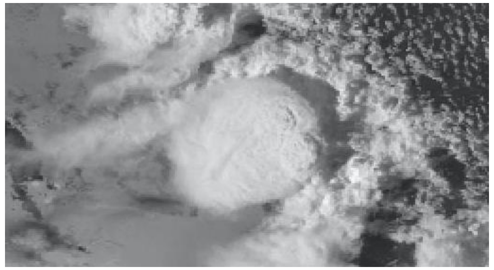
E.g. backscatter triplet from scatterometer

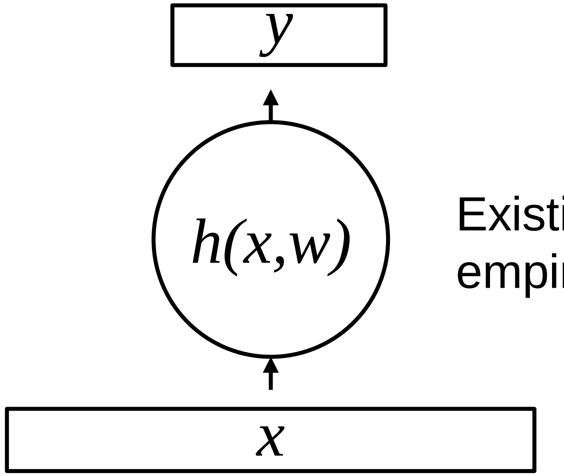
Train a neural network observation operator (w = weights) where a physical model is not available

E.g. ocean surface wind speed

- See Sean Healy's talk this workshop: a neural network scatterometer observation operator
- Example (in inverse direction) operationally used at ECMWF for soil moisture assimilatiuon from SMOS: Rodriguez-Fernandez et al., 2019, "SMOS Neural Network Soil Moisture Data Assimilation in a Land Surface Model and Atmospheric Impact", https://www.mdpi.com/2072-4292/11/11/1334

Machine learning within existing physical observation operators

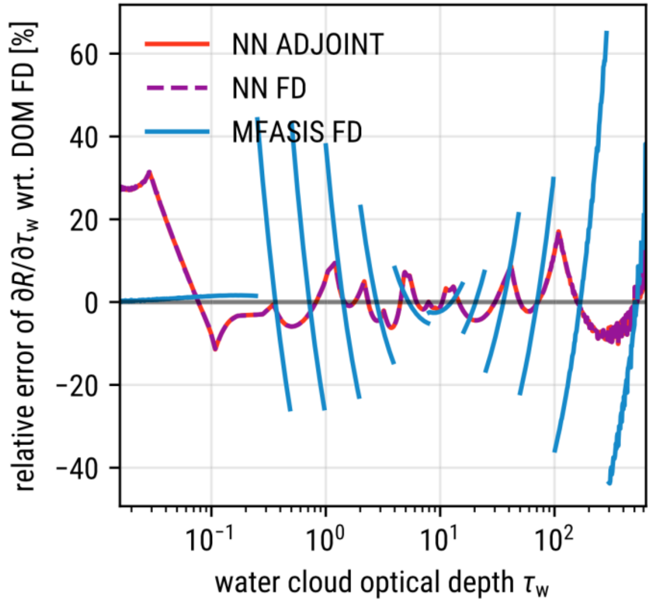




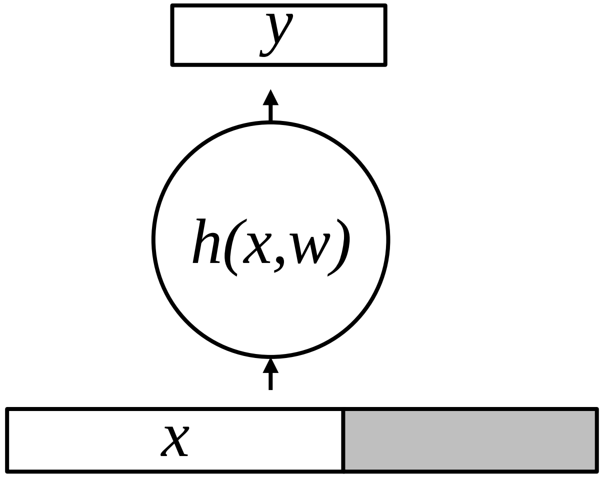
Existing physical-empirical model

Example: MFASIS cloudy visible reflectance observation operator within RTTOV radiative transfer model

- Scheck (2021, https://doi.org/10.1016/j.jqsrt.2021.107841)
- Replace 8 GB lookup table with 20 KB neural network
- Neural network gives much smoother gradients than the lookup table



What about poorly-known or unknown geophysical variables?

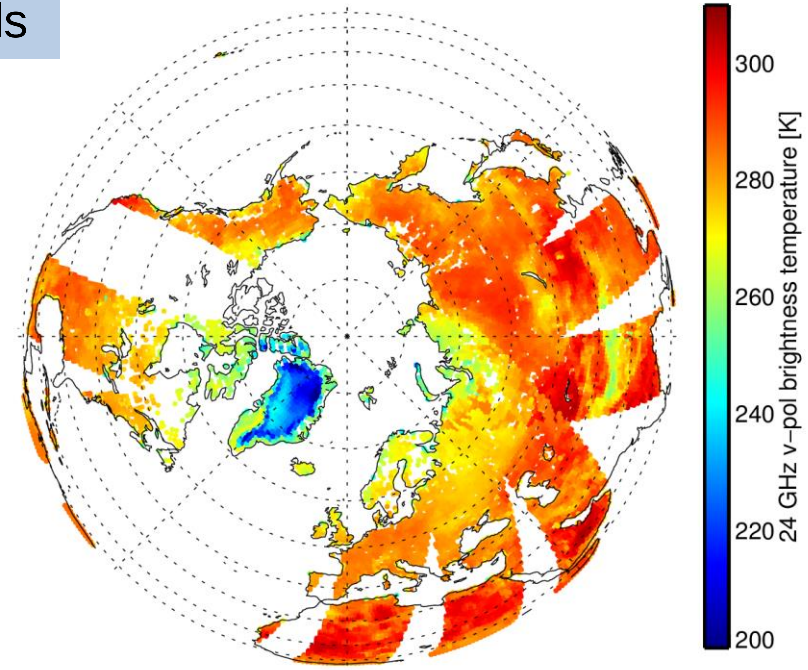


Variables that cannot easily be described or constrained by a physical model, but to which the observations are sensitive

Observation operator for microwave land surface radiative transfer

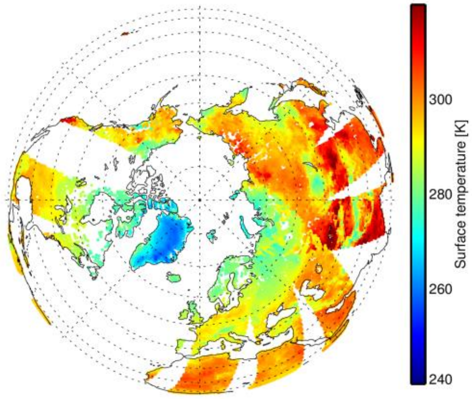
AMSR2 24GHz v-pol observations

Labels



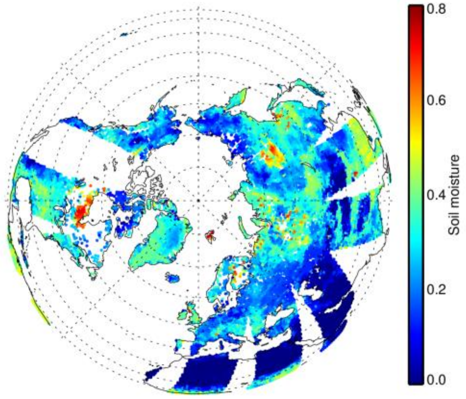
See also talks by Filipe Aires and Eulalie Boucher in this workshop

10 possible predictors for the brightness temperature

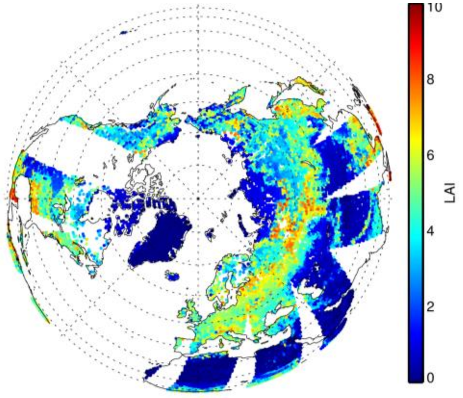


Skin temperature

Features



Soil moisture



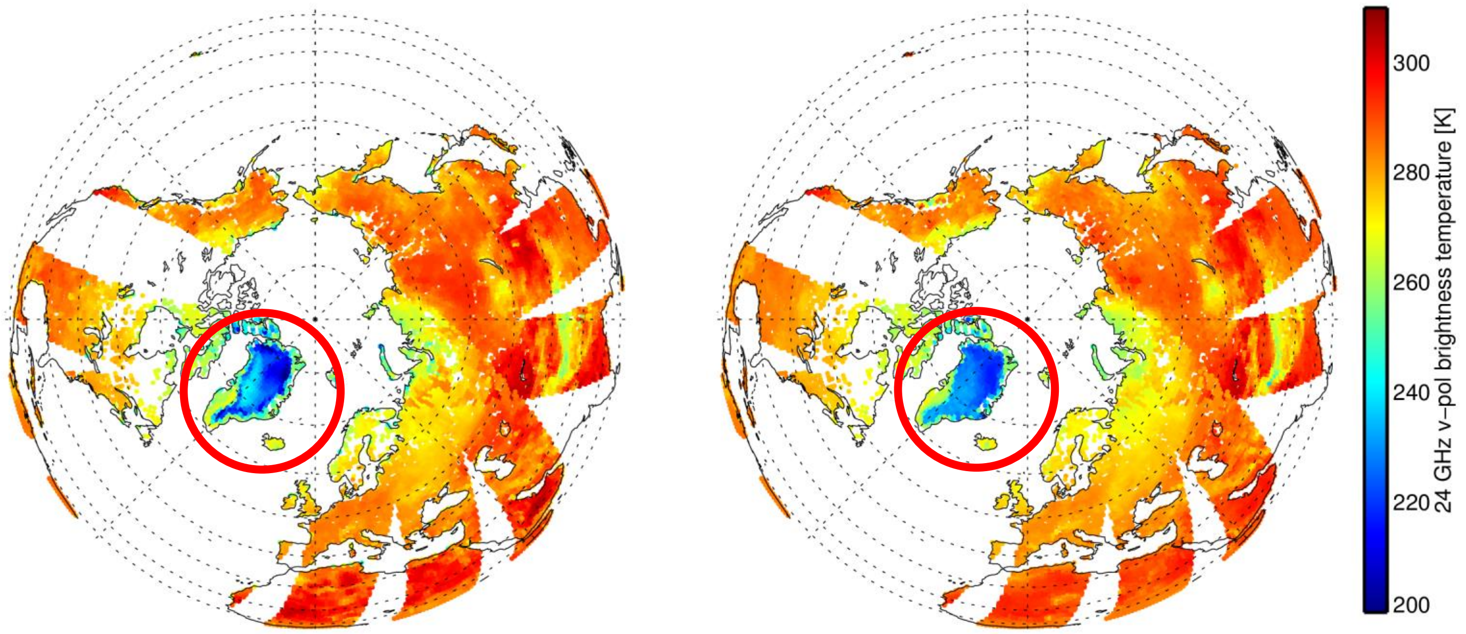
Leaf area index

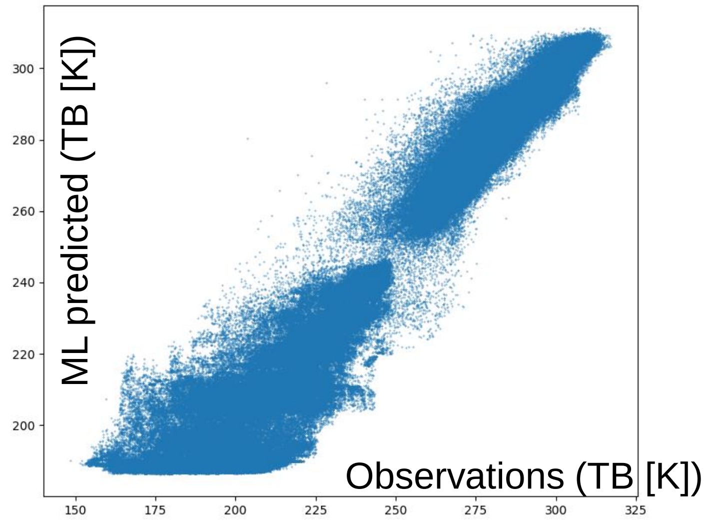
+ orography, snow depth, snow density, integrated water vapour, cloud, rain and snow water contents

Results (ability to fit training dataset)

Observations

ML predicted





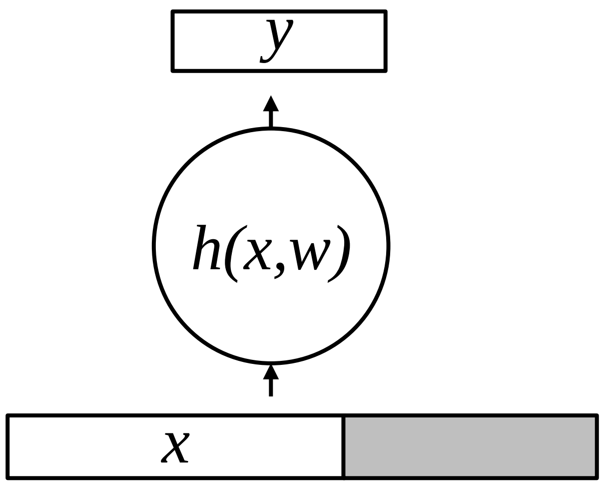
Minimum in observations: 150 K

Minimum in ML predictions: 185 K

The inputs do not contain enough information to drive outputs

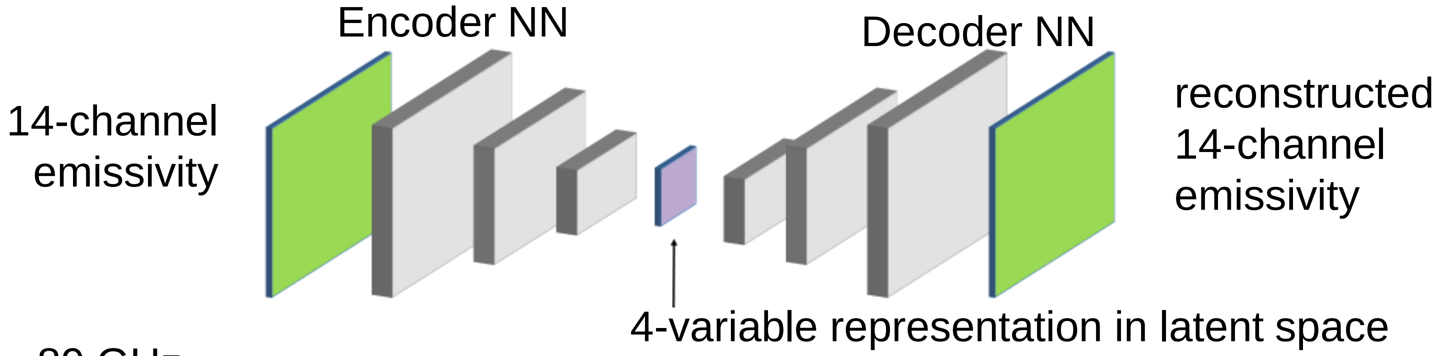
(in this case, no detailed knowledge of snow and ice microstructure)

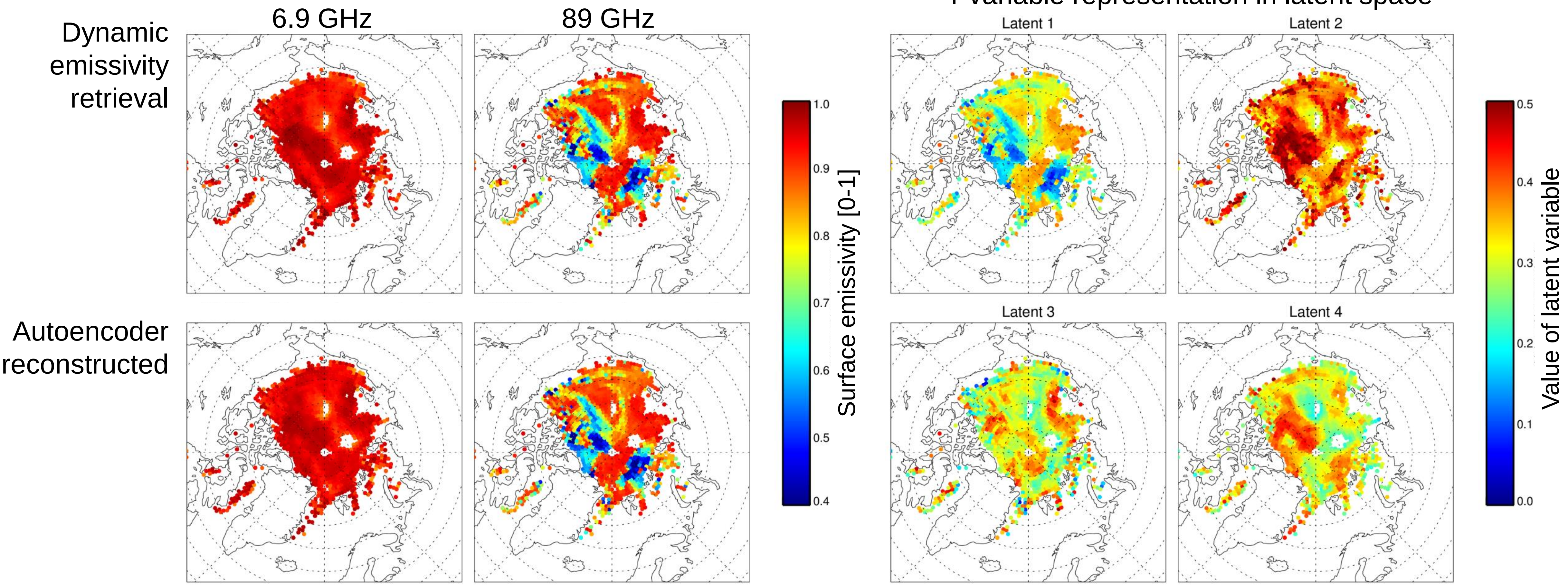
What about poorly-known or unknown geophysical variables?



Variables that cannot easily be described or constrained by a physical model, but to which the observations are sensitive

Machine learning for sea-ice emissivity: autoencoder approach

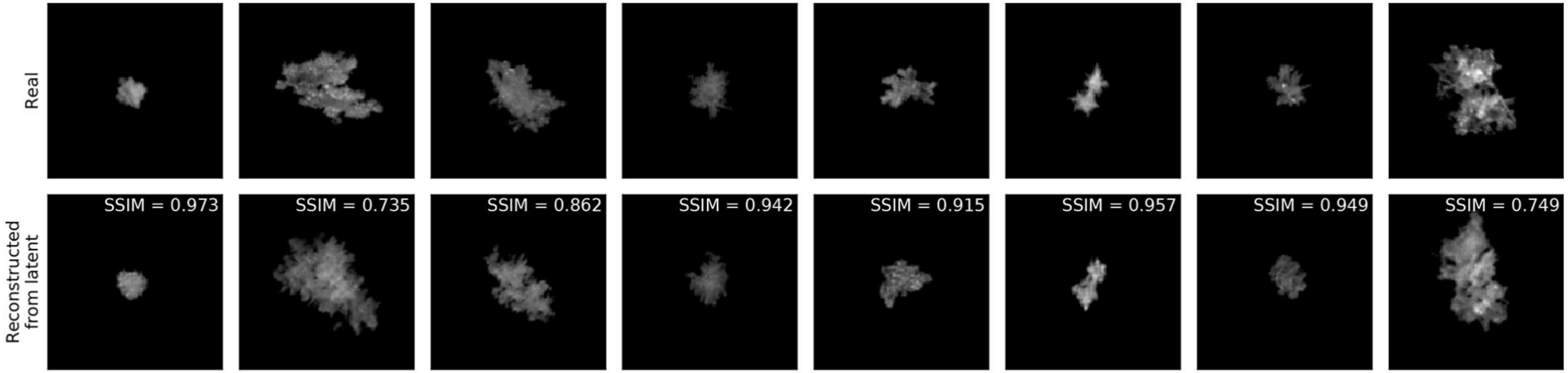




Generative learning: Generative-adversarial network (GAN) for snowflake pictures

Real snowflake image

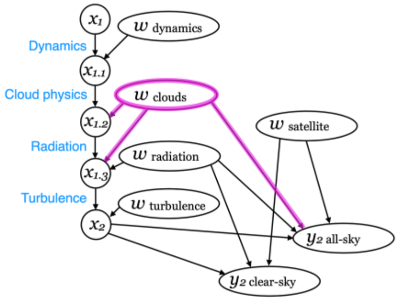
Reconstruction from latent space



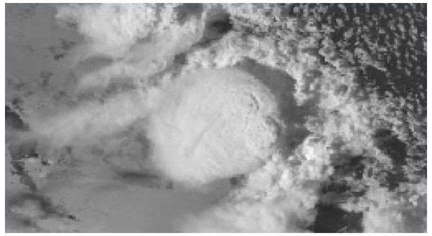
↑

- Leinonen and Berne (2020, Unsupervised classification of snowflake images using a generative adversarial network… , https://doi.org/10.5194/amt-13-2949-2020)
- Leinonen et al. (2021, Reconstruction of the mass and geometry of snowfall particles … , https://doi.org/10.5194/amt-14-6851-2021)
- See also Jussi Leinonen's talk this workshop (different work)

Some current challenges or hopes



- Train a neural network online within an operational-scale data assimilation system
  - E.g. Fortran-Keras bridge (e.g. Ott, 2020, https://doi.org/10.1155/2020/8888811)
- Very large-scale neural networks in the earth sciences
  - Take full advantage of cloud computing and supercomputing platforms
  - Compare to e.g. GPT-3 AI - 1.75 x10^11 parameters (https://arxiv.org/abs/2005.14165)
  - E.g. ECMWF operational weather model state vector 10^10 variables



- Learn an empirical model directly from observations that supersedes existing physical models
- Infer almost completely unknown variables, lacking reliable physical models or extensive observations to constrain them
  - Generative ML models, AI for physics discovery?
  - Data assimilation (impose physical models and observational constraints)?