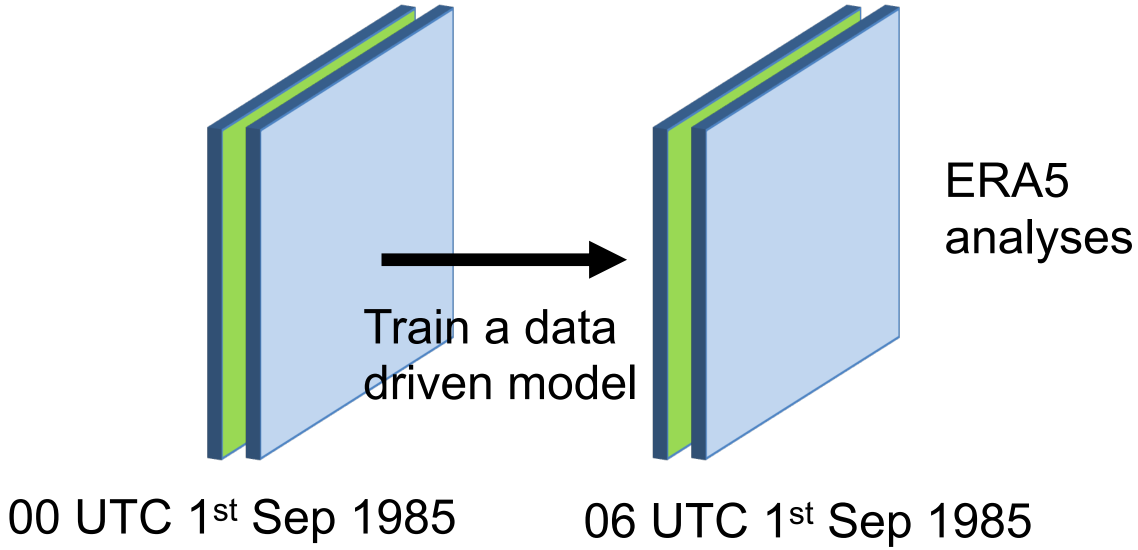
Data assimilation and machine learning

Alan Geer
European Centre for Medium-range Weather Forecasts
alan.geer@ecmwf.int

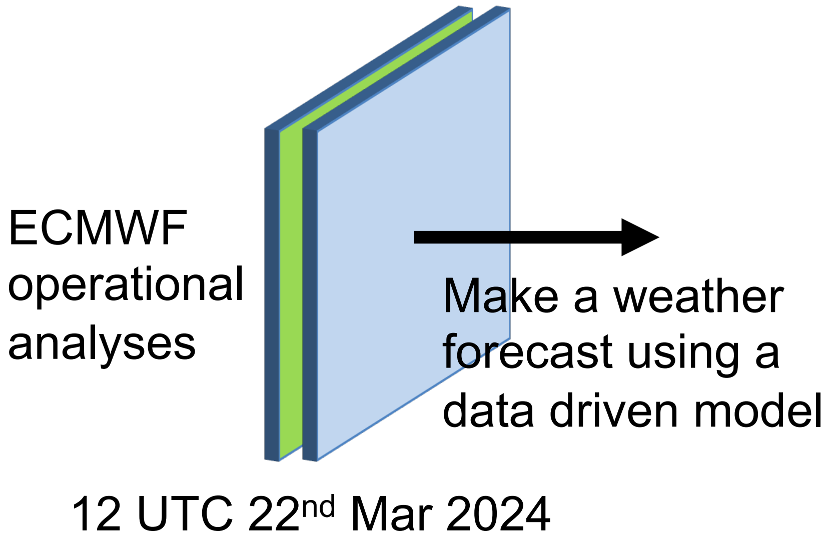
ECMWF machine learning training course, March 18 – 22, 2024

The importance of data assimilation in data-driven weather forecasting

- Training data
  - Sequences of gridded atmospheric and surface state variables
  - E.g. Graphcast was trained on ERA5 from 1978 to 2018

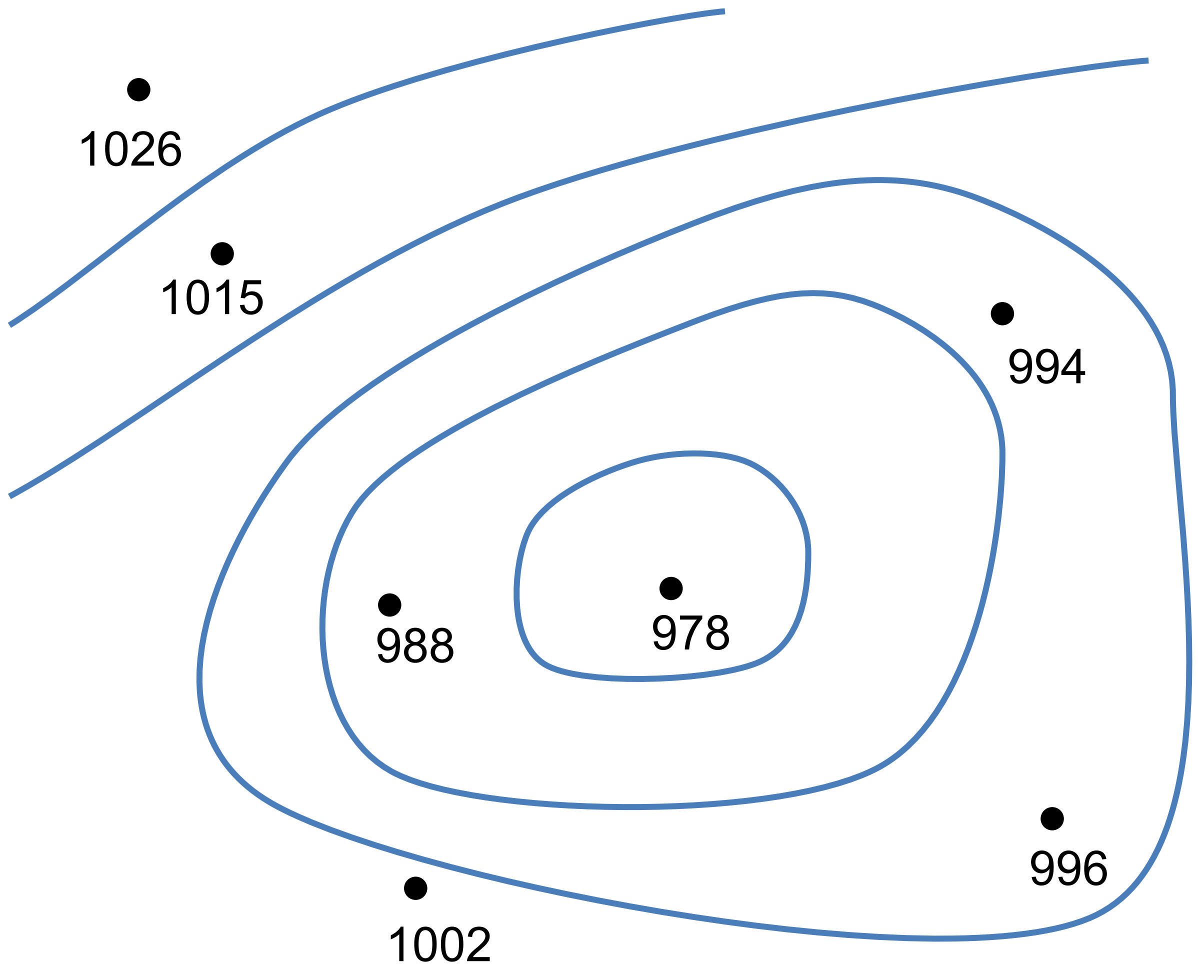


- Initial conditions
  - The current state of the atmosphere, from which we can start a forecast
  - E.g. to forecast the week ahead, we need to know the current state of the atmosphere and surface right now.



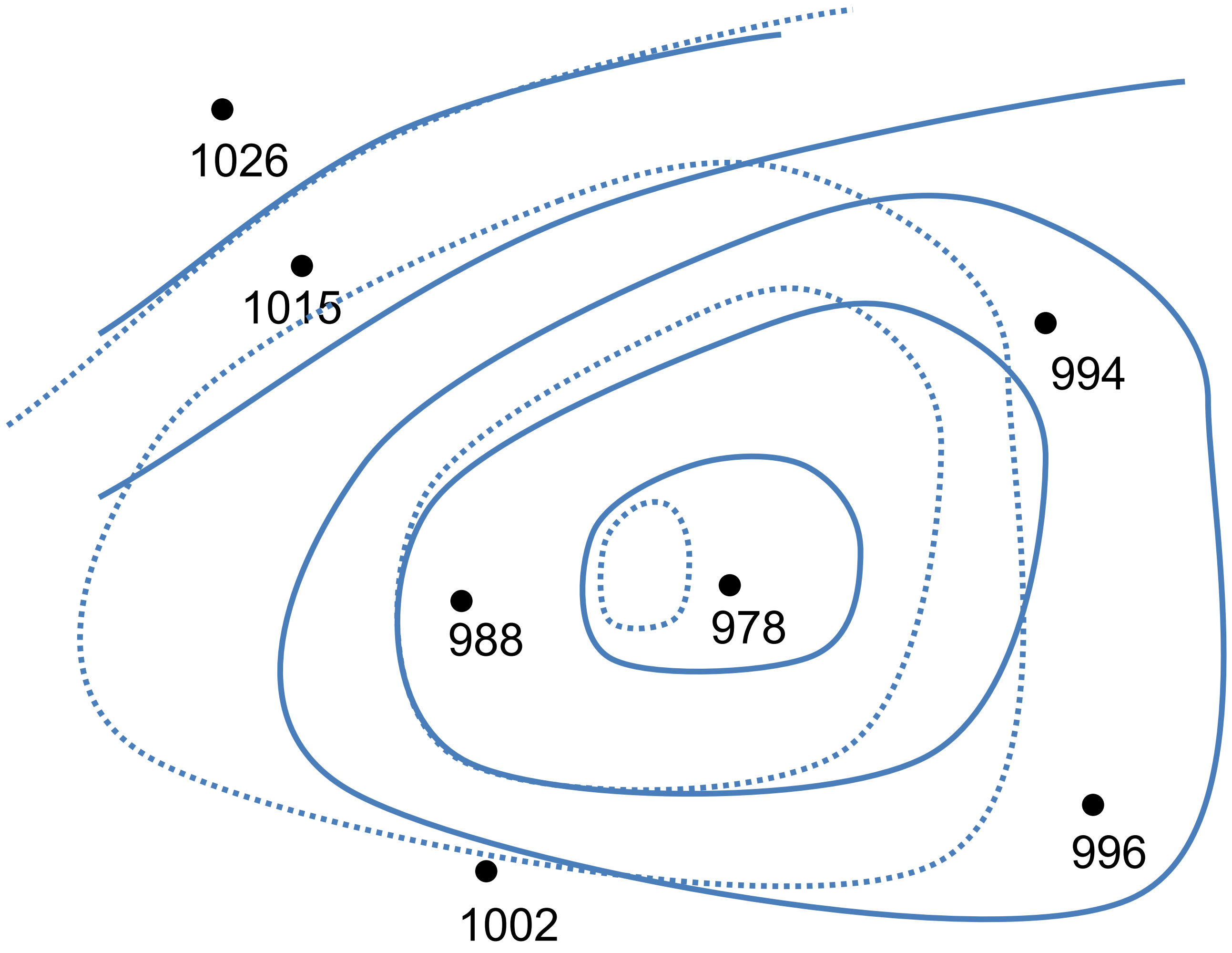
Why data assimilation?

Making a map of the atmosphere from irregularly spaced observations



- What can we use to help?
  - Physics: the known equations of the atmosphere
  - Prior knowledge: the previous forecast for today
  - Statistics: the relative size of errors in the observations and the previous forecast

Making a map of the atmosphere from irregularly spaced observations



Background forecast (prior guess for what the state will be)

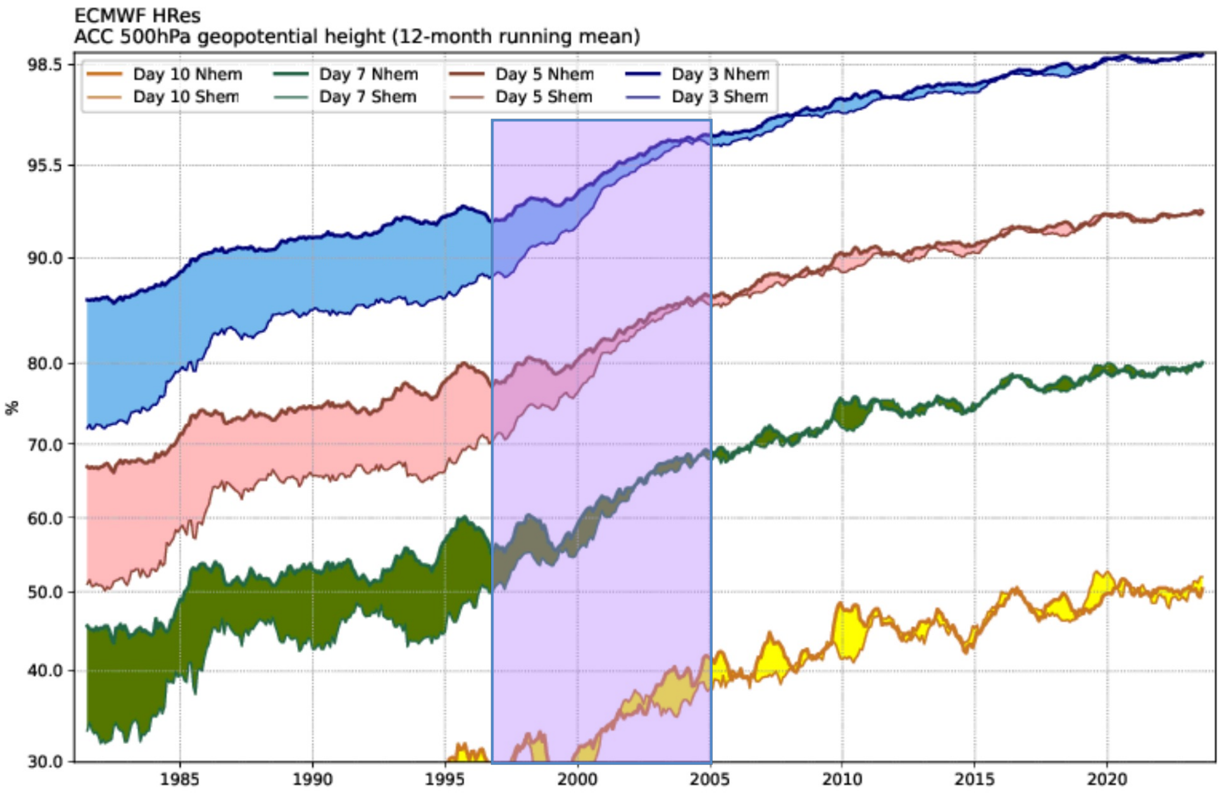
+

New observations

=

Analysis (best estimate of the current state of the atmosphere)

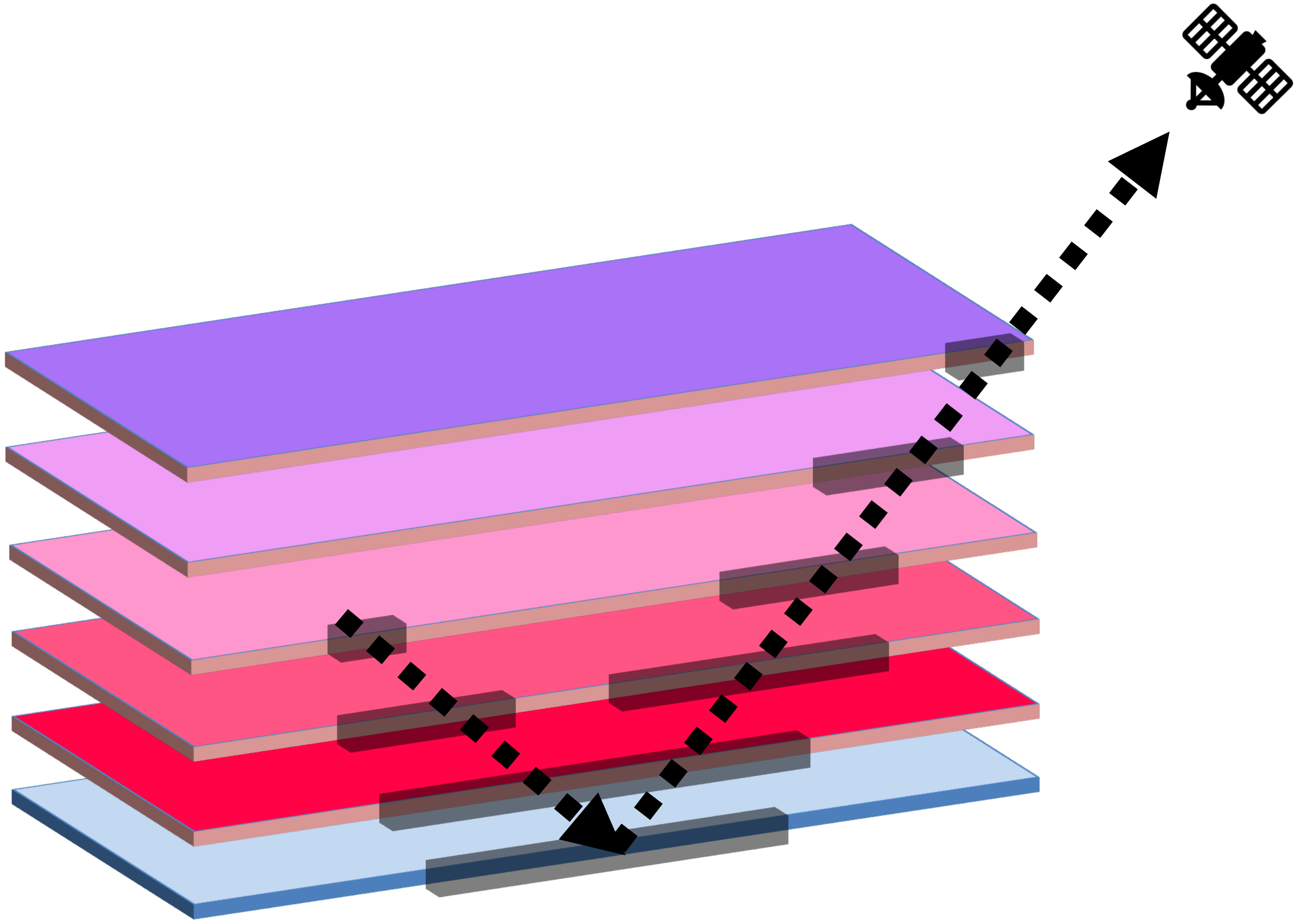
Lead time of anomaly correlation coefficient (ACC) reaching multiple thresholds (High resolution (HRES) 500 hPa height forecasts)



The closest to a step-change in ECMWF forecast skill: a few years following the introduction of 3D and 4D variational data assimilation (4D-Var)

- The ability to assimilate satellite radiances directly as radiances (rather than retrievals)
- The use of a physical forecast model trajectory as part of the data assimilation process

The inverse problem: using indirect observations of the state



At its simplest – a satellite radiance observation is a weighted average of the temperature in different atmospheric and surface temperature layers

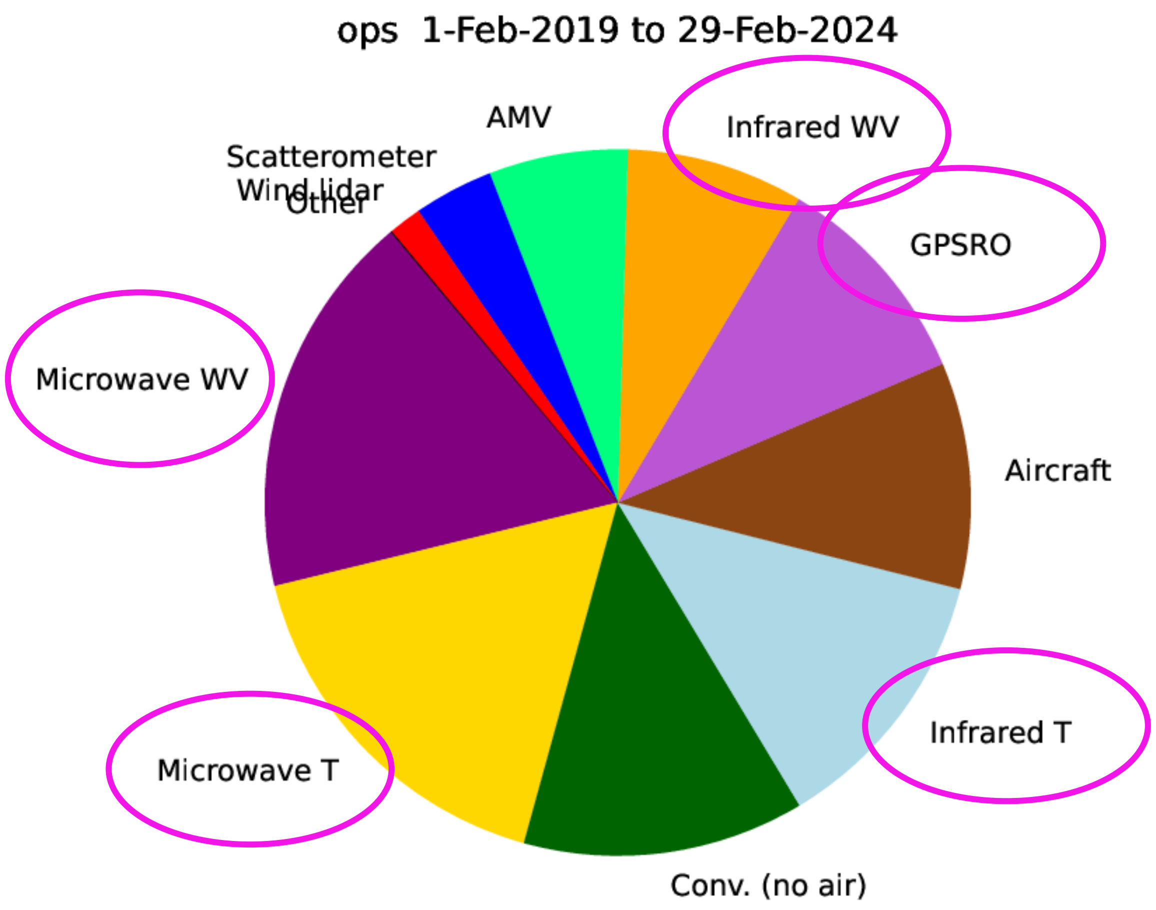
The inverse problem

One observation

$$y = h\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_{\dots} \end{pmatrix}$$

Depends on many state variables of the earth system

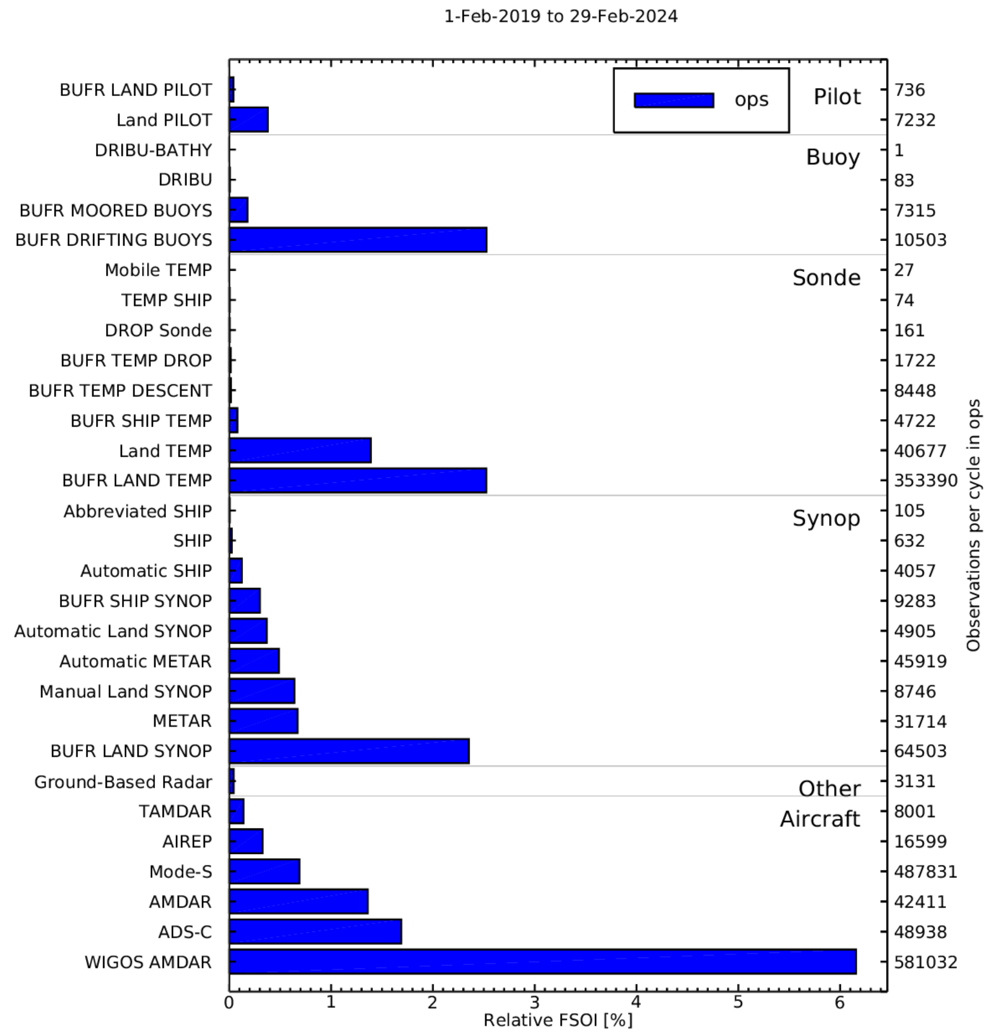
Observations used in operational data assimilation – by relative impact on the 24 hour forecast quality (FSOI)



Indirect observations provide the majority of the forecast impact

Number and diversity of observations

Surface based ("conventional") observation types – balloons, ground stations, aircraft, ships…

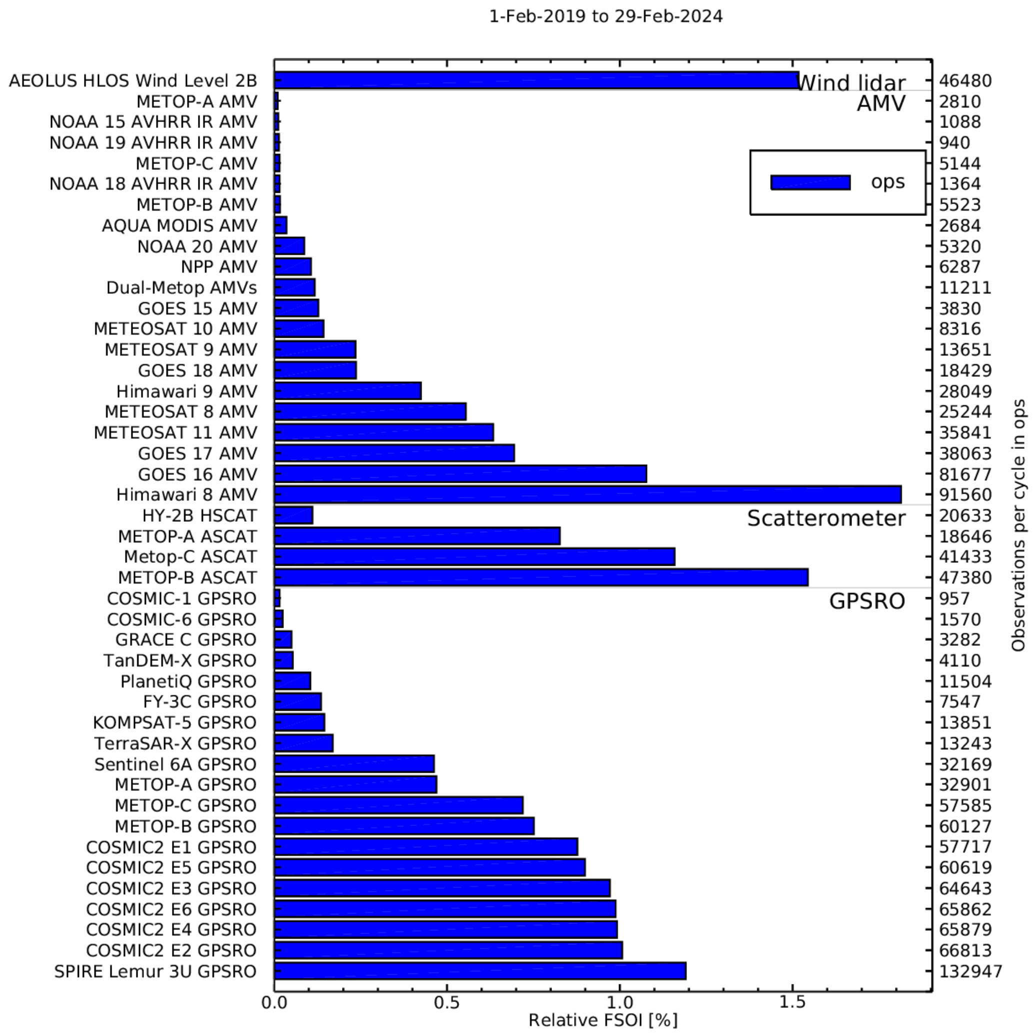


Number of new observations used every 12 hours

Number and diversity of observations

Satellites part I: winds from lidar, image tracking (AMV) and ocean surface scatterometry,

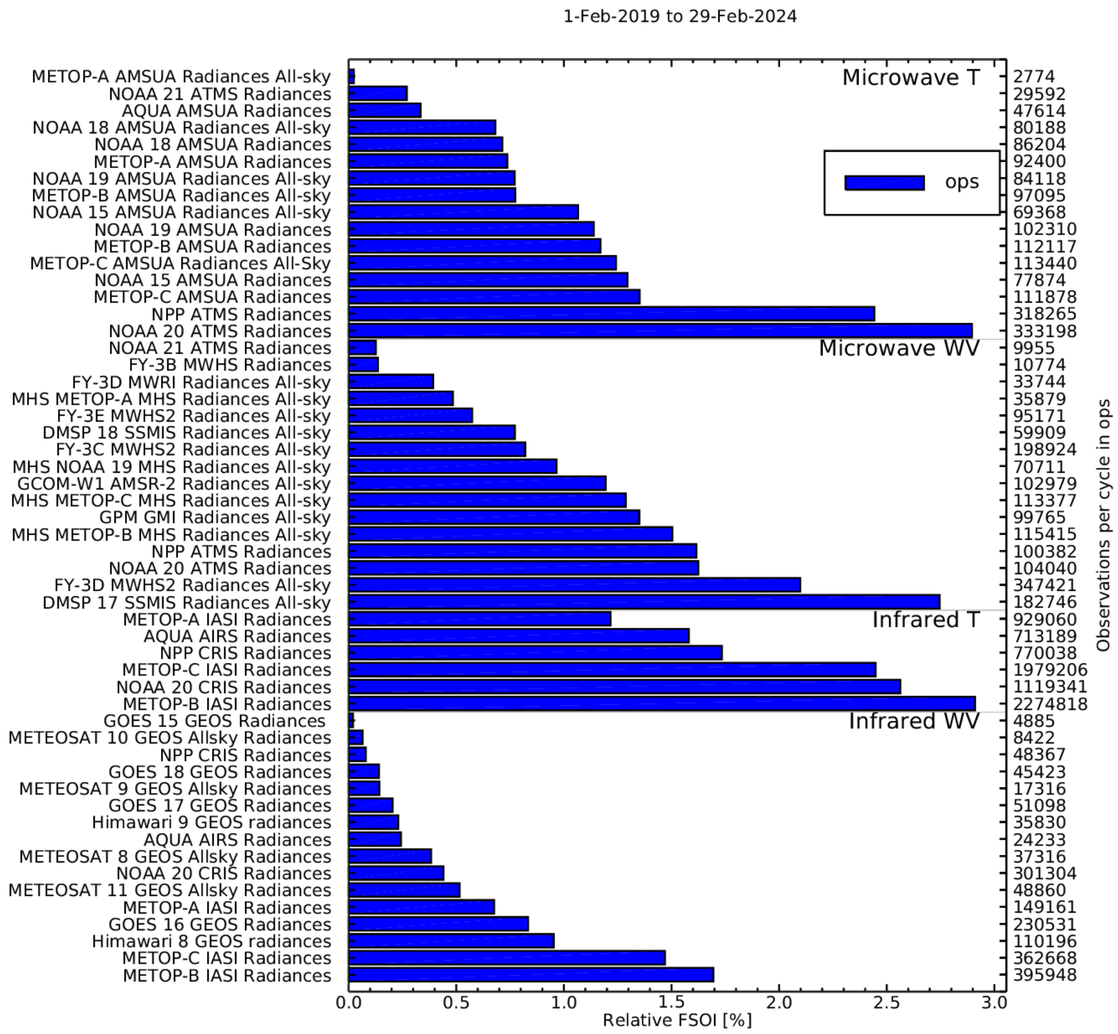
Radio occultation (GPSRO)



Number of new observations used every 12 hours

Number and diversity of observations

Satellites part 2: Radiances at microwave and infrared wavelengths



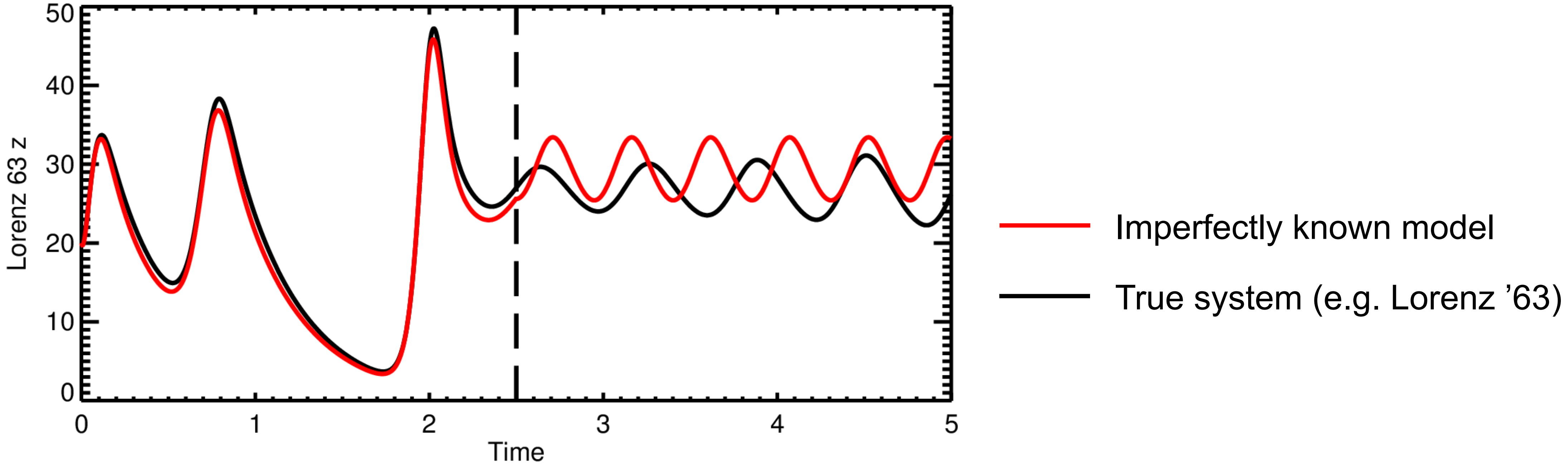
Number of new observations used every 12 hours

… approximately 16 million across all observation types

… about **10 days** of observations are needed to make the best analysis: approximately **300 million observations**

What is data assimilation?

Data assimilation ↔ dynamical systems, control theory, statistical physics etc.

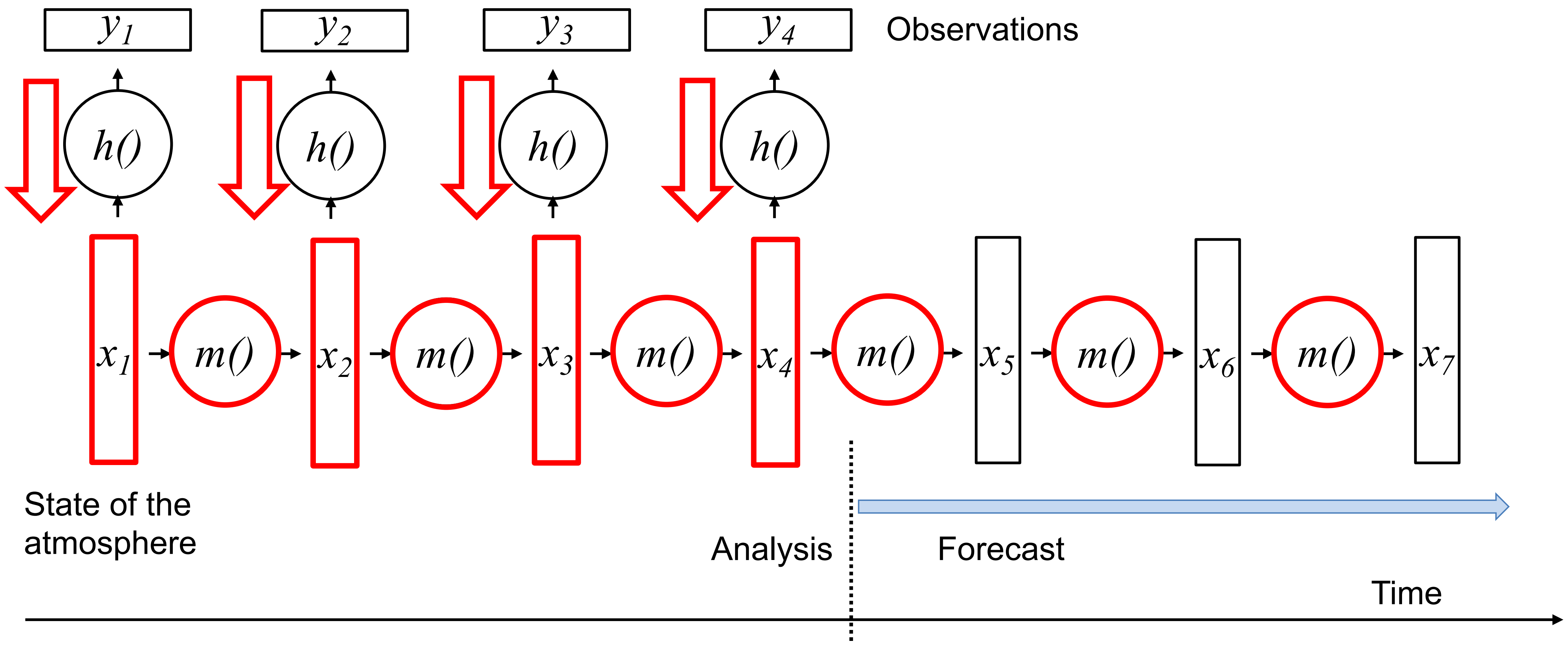


synchronise (e.g. DA) predict

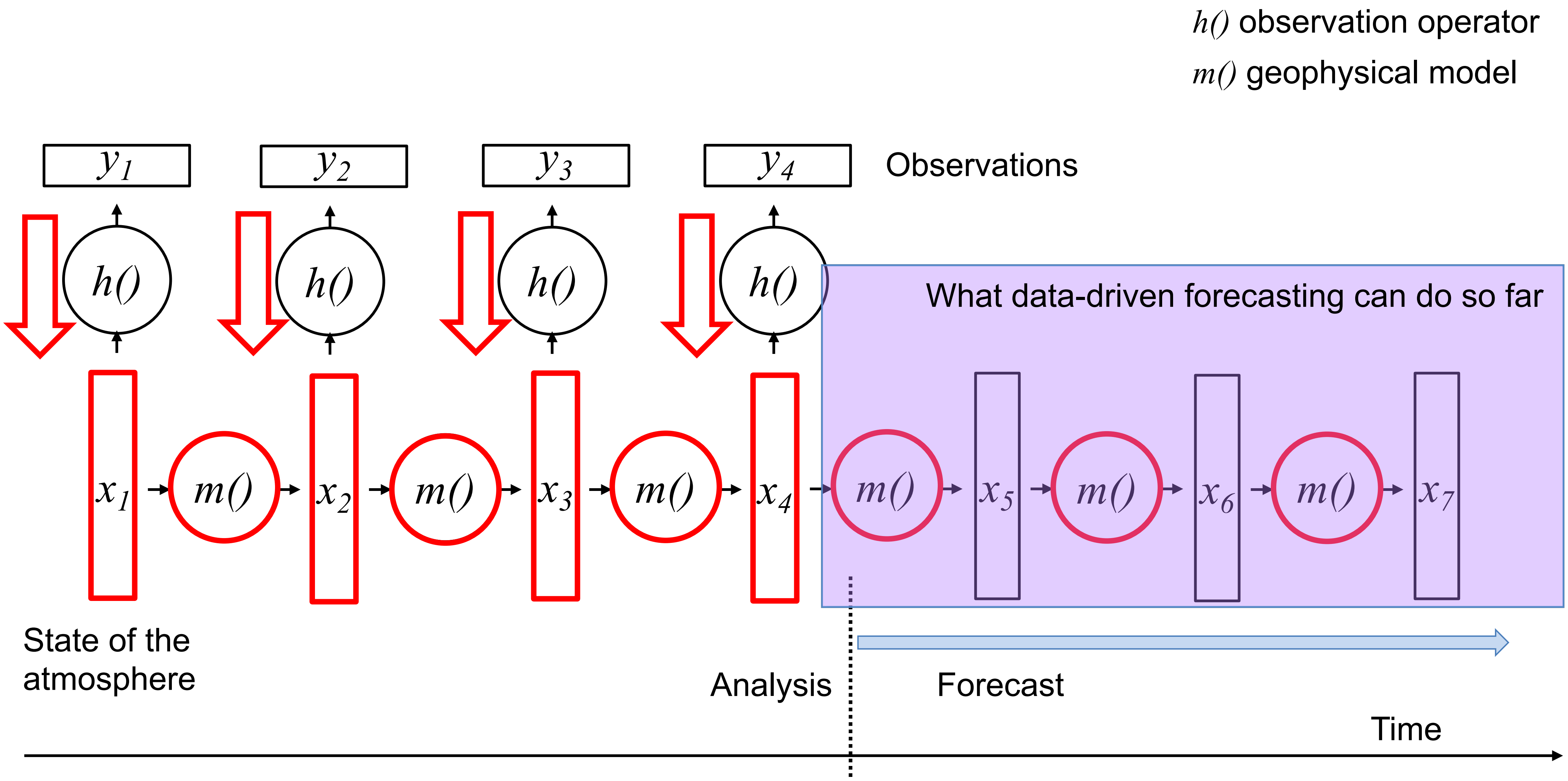
Data assimilation

$h()$ observation operator

$m()$ geophysical model



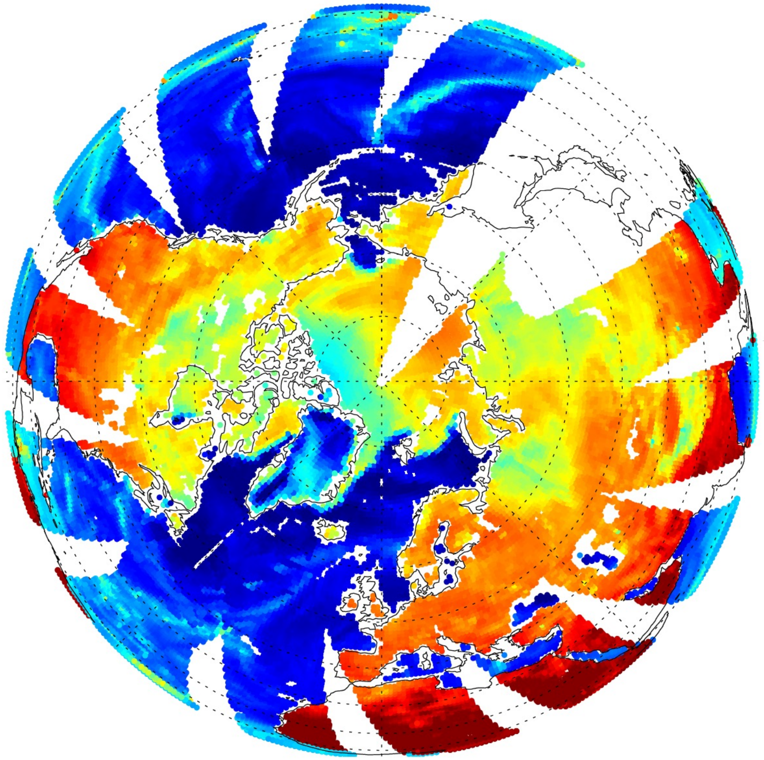
Data assimilation



The inverse problem

Physical forward model

Satellite observations



SSMIS F-17 channel 13 (19 GHz, v)
Microwave brightness temperatures
3rd December 2014

$$y = h\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_{\dots} \end{pmatrix}$$

Forward function / observation operator / observation model

Geophysical variables

Atmospheric temperature, water vapour, wind, cloud, precipitation

Skin and substrate temperature and moisture

Ocean wind, waves, foam

Sea-ice

Snowpack

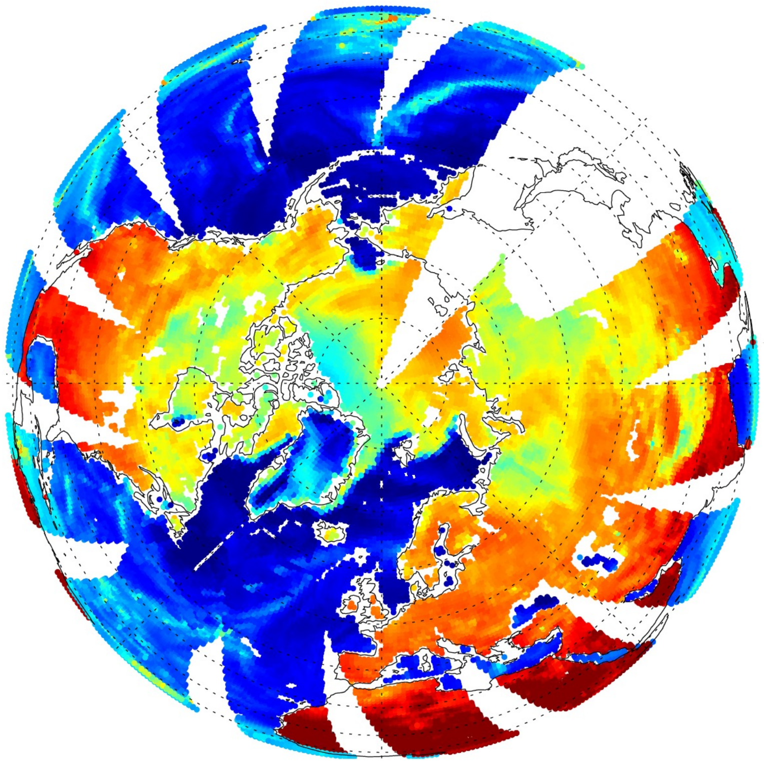
Ice

Vegetation

Soil

Physical forward model

Satellite observations



SSMIS F-17 channel 13 (19 GHz, v)
Microwave brightness temperatures
3rd December 2014

Geophysical variables

Equations & parameters – where sometimes knowledge is quite uncertain

$$y = h\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_{\dots} \end{bmatrix}, \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_{\dots} \end{bmatrix}\right)$$

Forward function

Gas spectroscopy

Scattering from hydrometeors

Cloud and precipitation micro and macro-structure

Snow / ice grain size and structure

…..

The forward and inverse problem

Forward model

Observations

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

Geophysical state

Model parameters

$$\cancel{x, w = h^{-1}(y)}$$

No unique solution: ill-posed

Inverse model

The best that observations can do is to provide a statistical improvement in our knowledge of x and w

The inverse problem solved by Bayes theorem

Observations

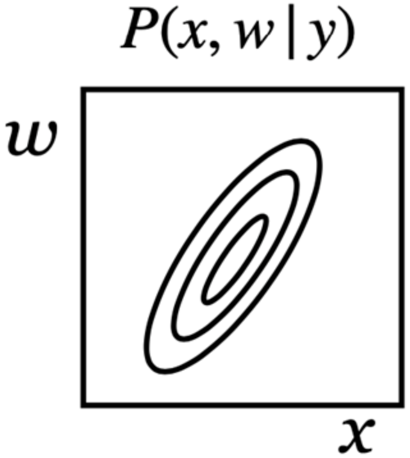
Geophysical state

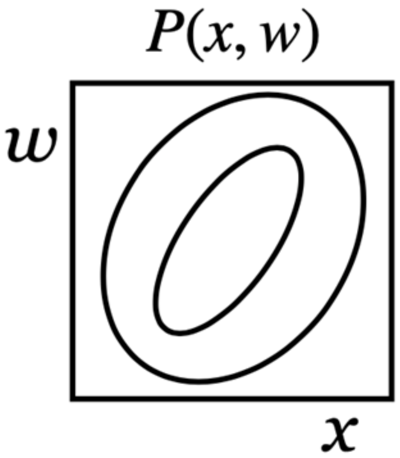
Model parameters

Bayes theorem

$$P(x,w|y) = K\,\left(y, P(y|x,w), P(x,w)\right)$$

Probabilistic equivalent of the forward model h()





(Posterior) Probability of x and w given y

Prior probability of x and w

Cost function for variational DA

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

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

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

Links to machine learning

Cost / loss function equivalence of ML and variational DA

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

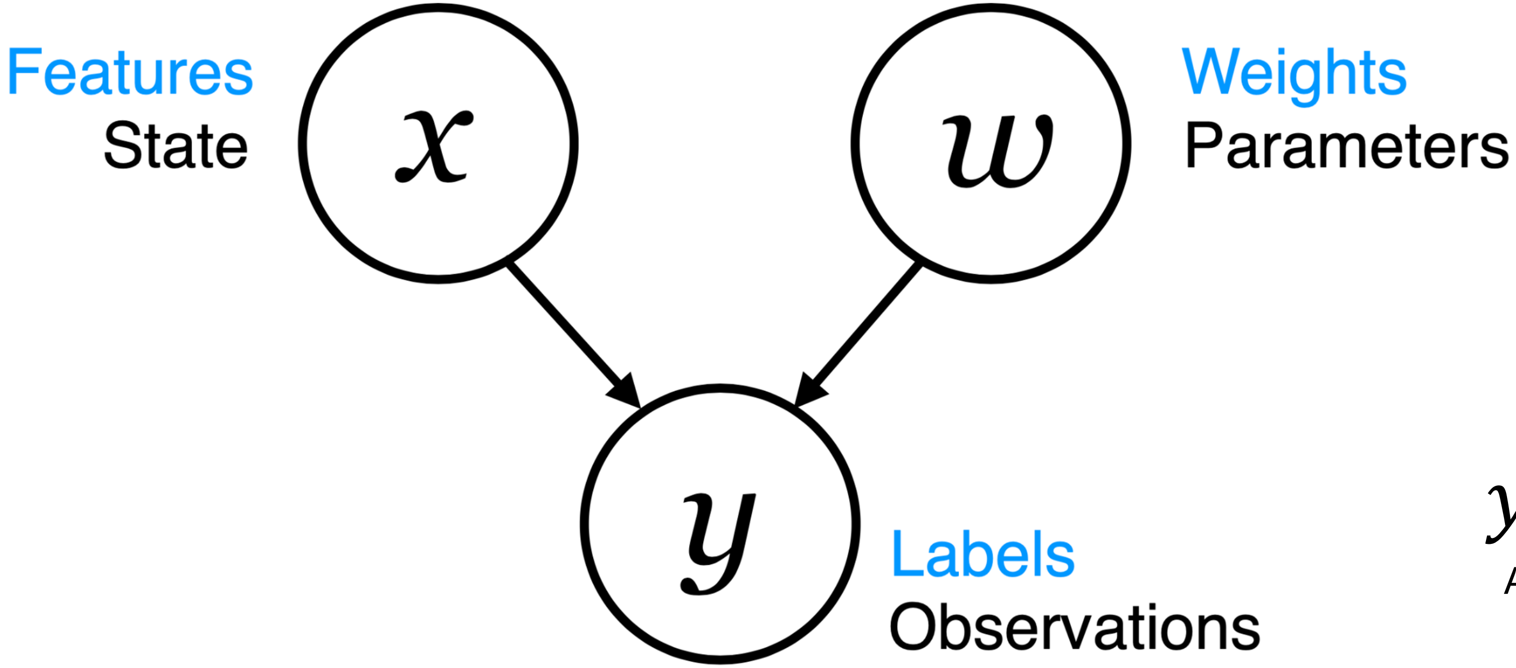
ML	Loss function	Basic loss function	Feature error?	Weights regularisation

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

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

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

Bayesian equivalence of ML and DA



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

As a Bayesian network

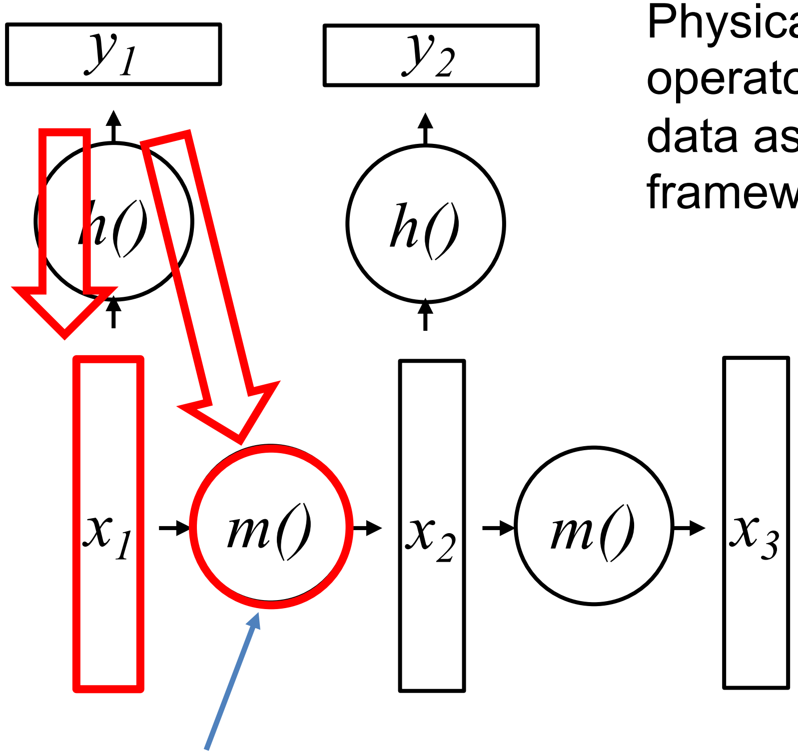
Geer (2021) https://doi.org/10.21957/7fyj2811r
Bocquet et al. (2020) https://arxiv.org/abs/2001.06270
Abarbanel et al. (2018) https://doi.org/10.1162/neco_a_01094
Hsieh and Tang (1998) https://doi.org/10.1175/1520-0477(1998)079%3C1855:ANNMTP%3E2.0.CO;2
Goodfellow et al. (2016) https://www.deeplearningbook.org

How can machine learning and data assimilation help each other?

Use 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

Use data assimilation to learn directly from observations



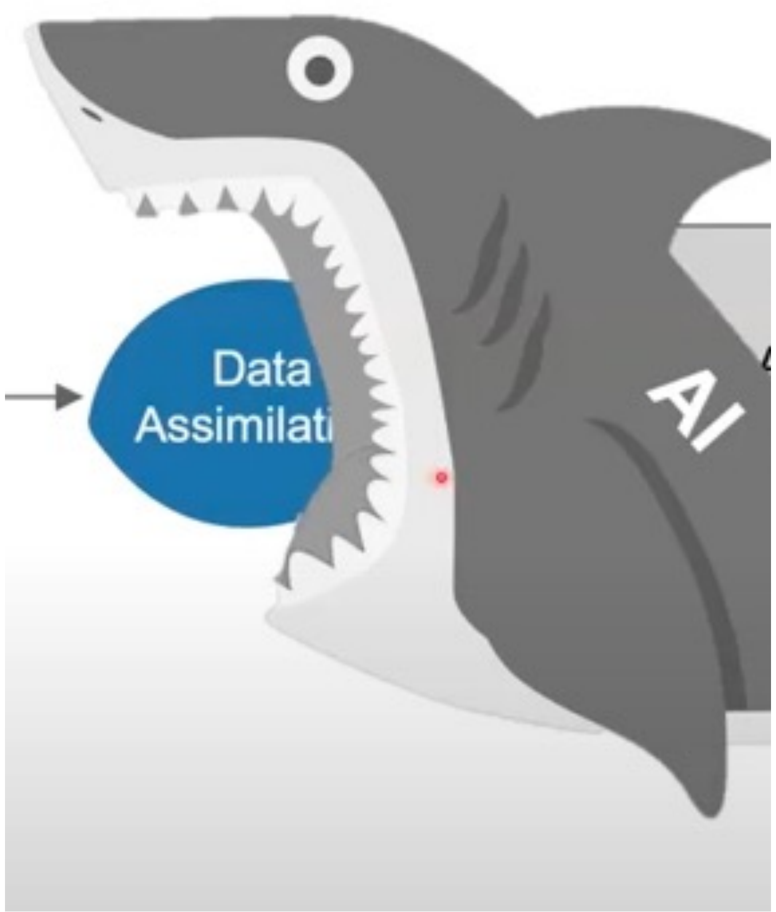
Physical observation operator and physical data assimilation framework

Data driven model for the atmosphere which is learned simultaneously with the atmospheric state

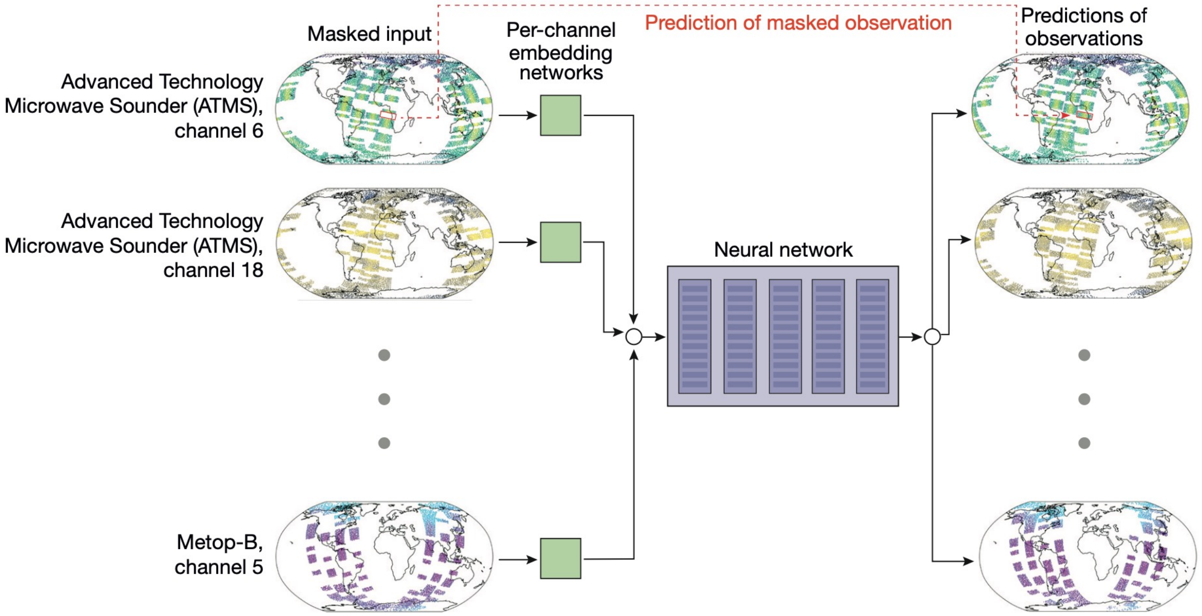
- Simultaneous estimation of the initial conditions, NN parameters and dynamical parameters of a model (e.g. Lorenz ’63) using data assimilation (Hsieh and Tang, 2001, https://doi.org/10.1175/1520-0493(2001)129<0818:CNNTID>2.0.CO;2)
- Use iterative cycles of data assimilation followed by neural network training (Brajard et al., 2020, https://doi.org/10.1016/j.jocs.2020.101171)

Use machine learning to replace data assimilation altogether

- Stephan Rasp's "big shark" at ISDA online - https://www.youtube.com/watch?v=CoiVfwJU4TY

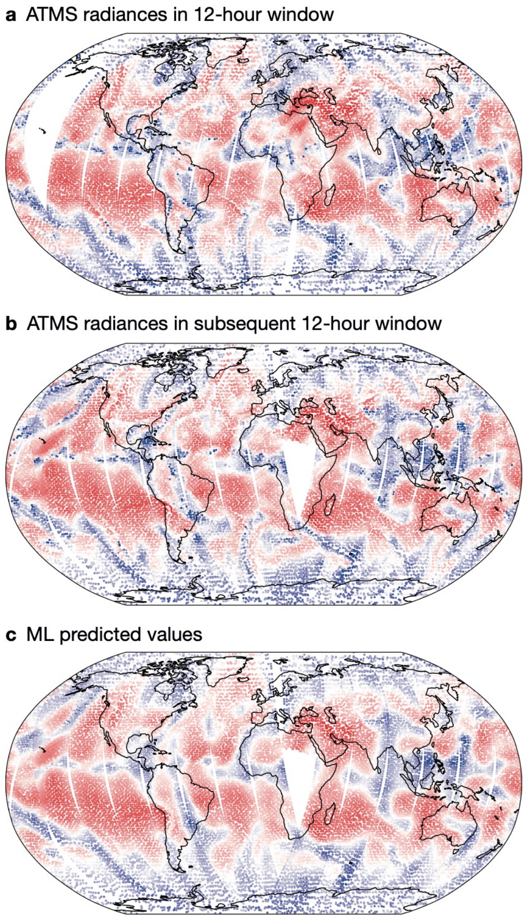


Direct observation prediction – a new project at ECMWF



12 UTC 22nd Mar 2024

00 UTC 23rd Mar 2024



Tony McNally et al. (2024, ECMWF newsletter) - https://www.ecmwf.int/en/newsletter/178/earth-system-science/red-sky-night-producing-weather-forecasts-directly

Hybrid empirical-physical modelling

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

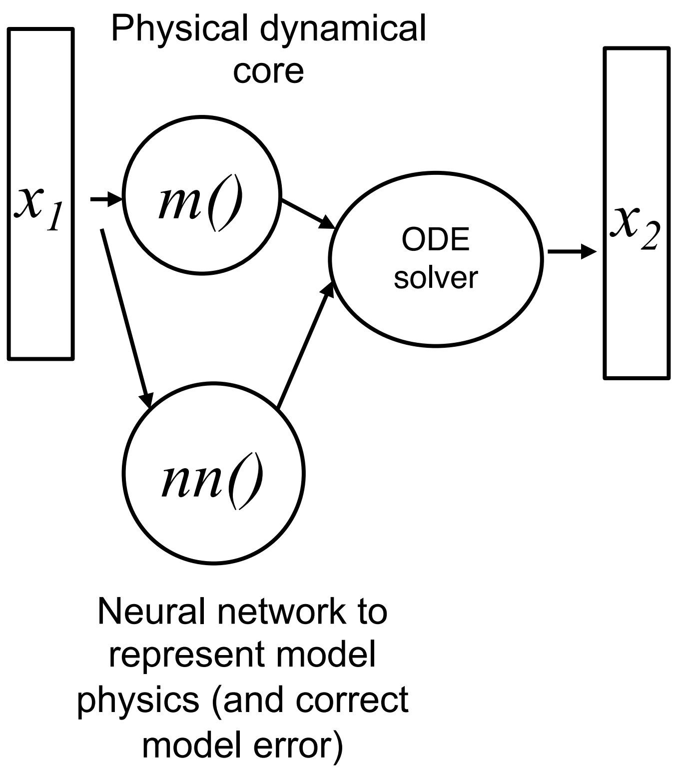
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \upsilon\frac{\partial^2 u}{\partial x^2} = 0$$

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)

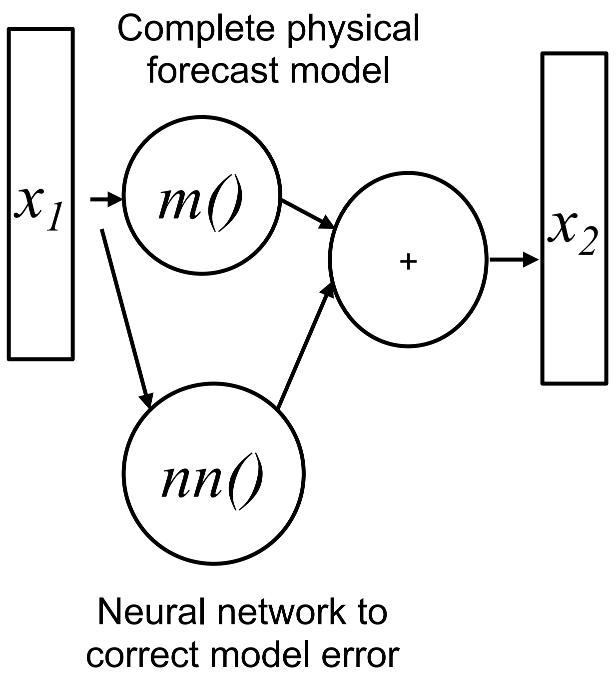
Hybrid physics – machine learning: "Neural GCM"



Kochkov et al. (2023) Neural General Circulation Models https://doi.org/10.48550/arXiv.2311.07222

Trained on data assimilation outputs (ERA5)

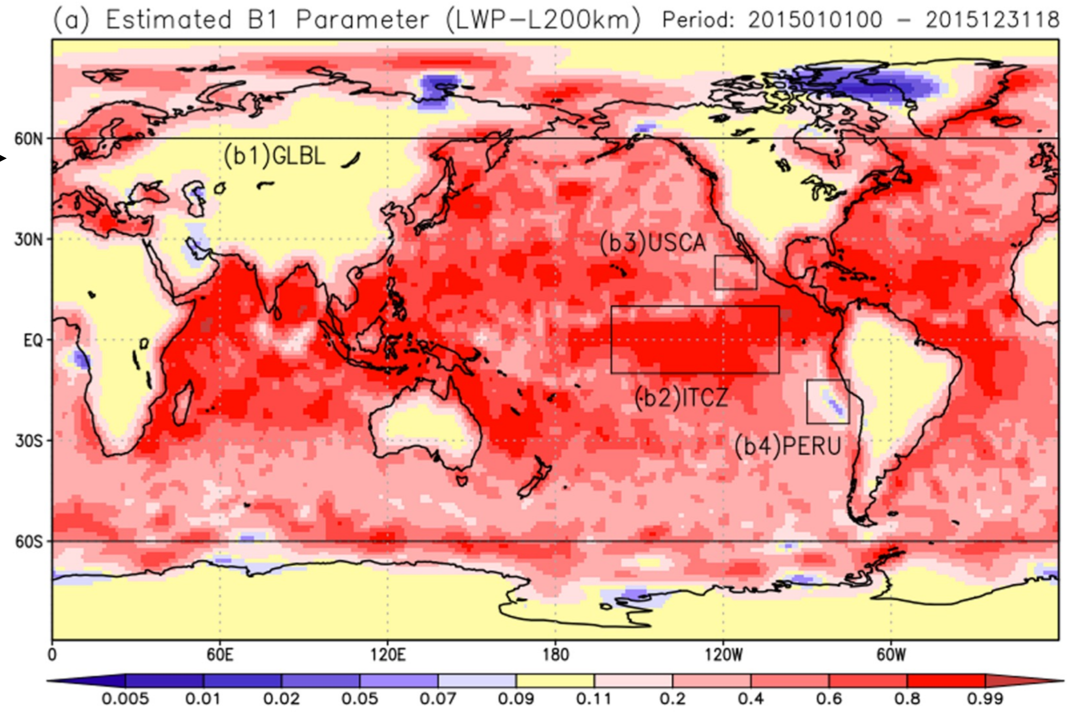
Hybrid physics – machine learning: ”Model error correction"



See Massimo Bonavita’s lecture

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

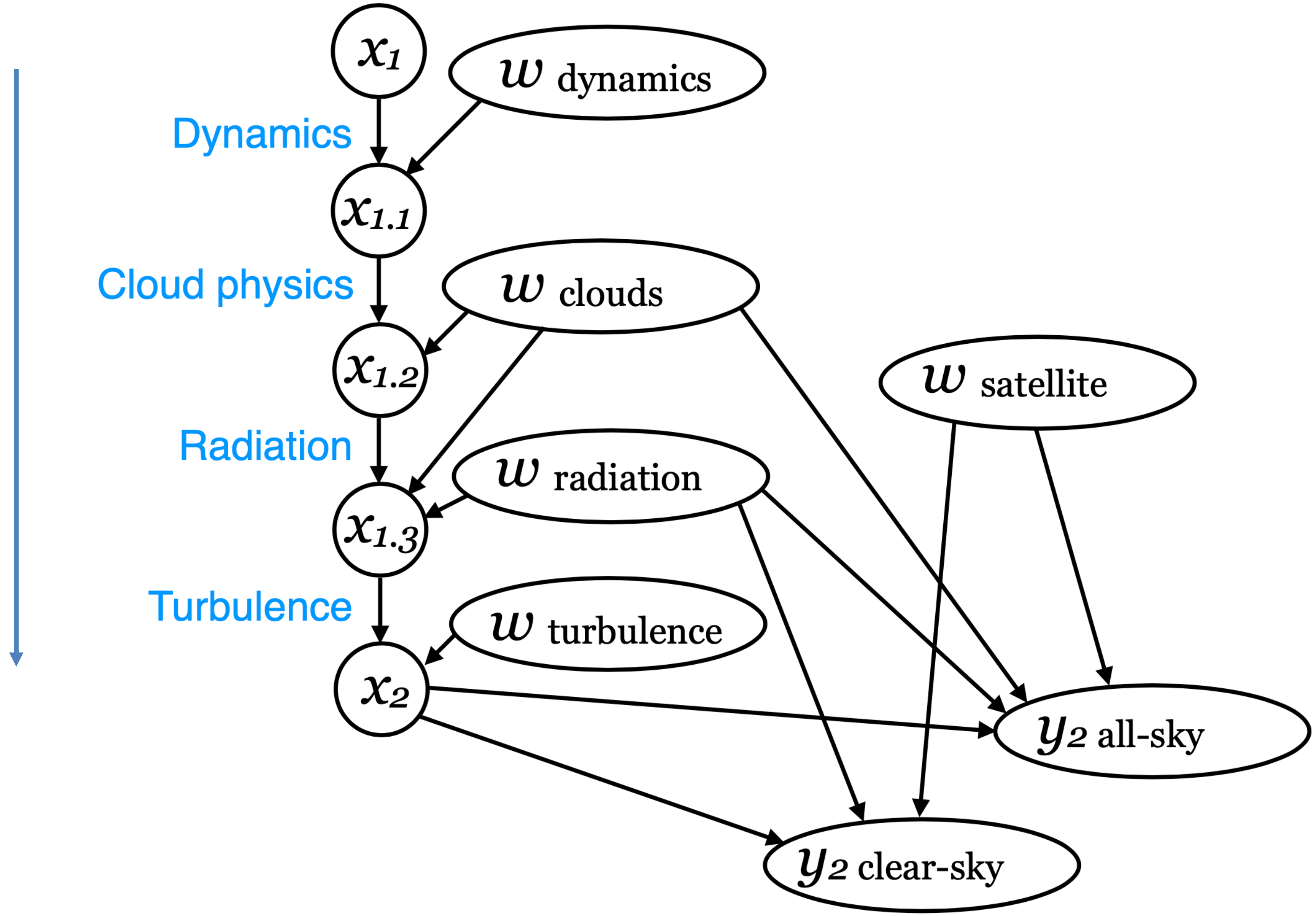


Hybrid empirical-physical modelling

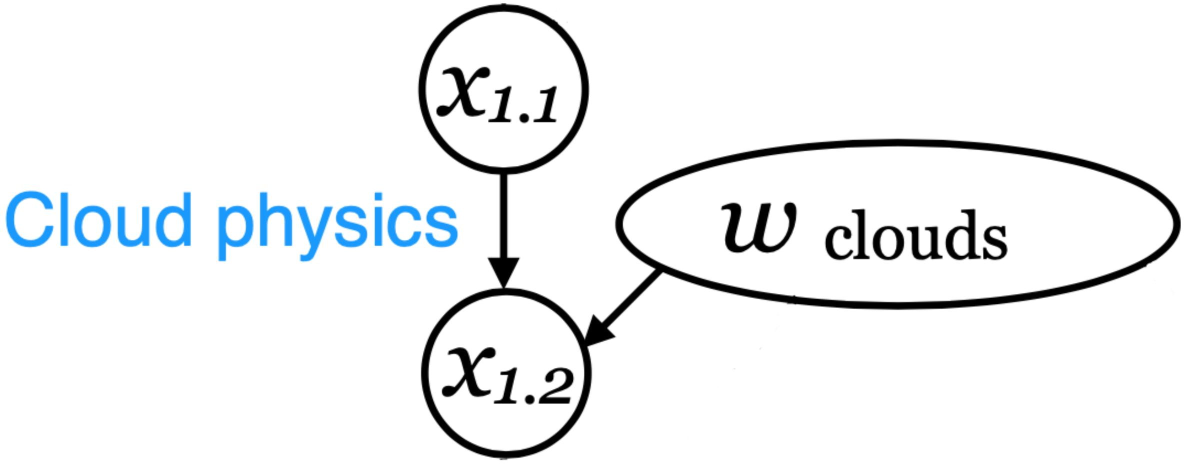
A more granular (network) approach

Inside an atmospheric model & data assimilation timestep

One model time-step

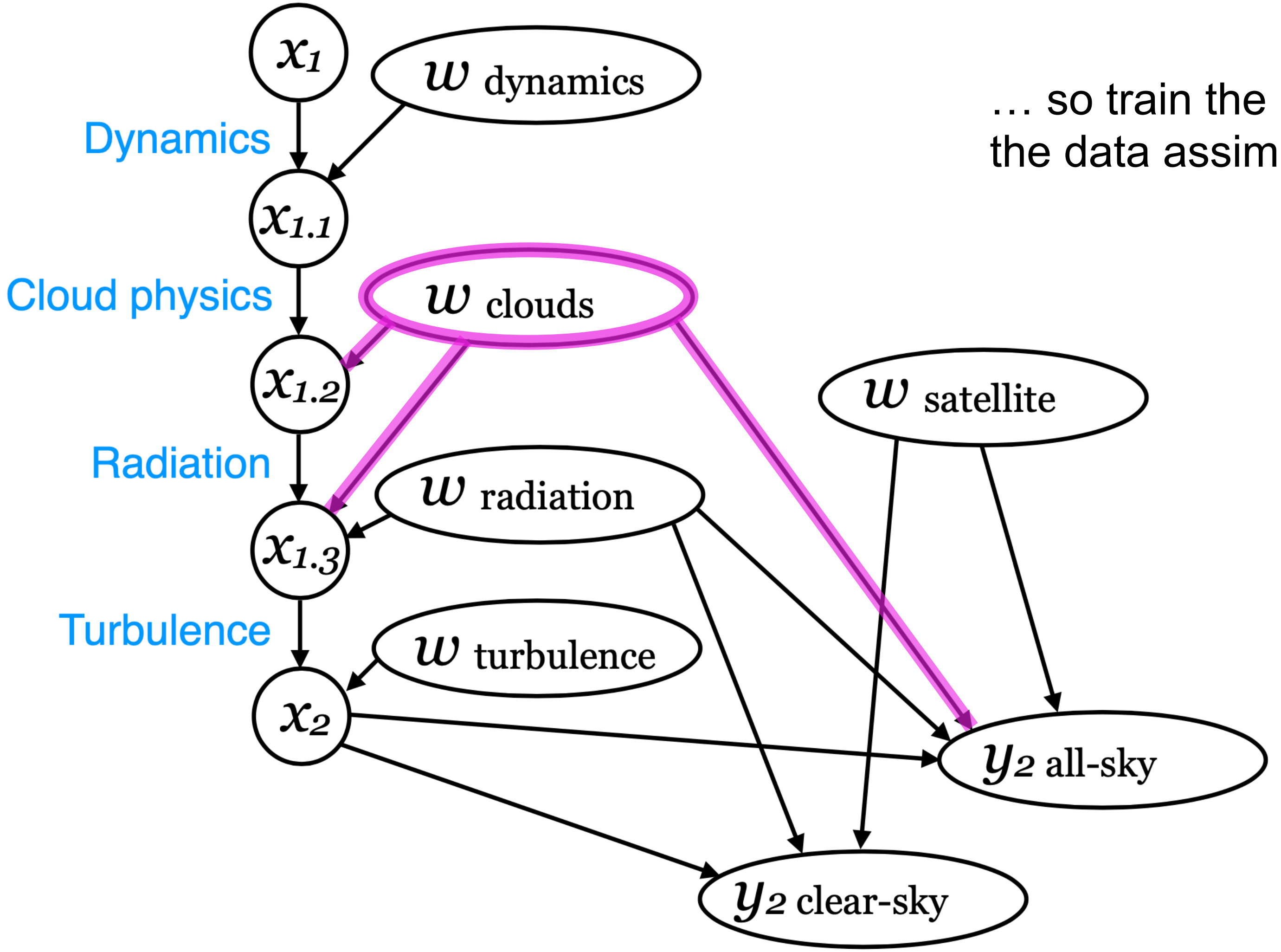


Learning an improved model of cloud physics (ML or DA)



We want to train a model against observations, but we cannot directly observe gridded intermediate states $x_{1.1}$ and $x_{1.2}$ … or more precisely model tendencies …

Inside an atmospheric model



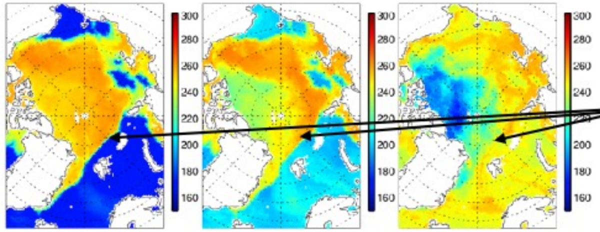
… so train the model inside the data assimilation system

Hybrid data assimilation and machine learning

Sea ice example

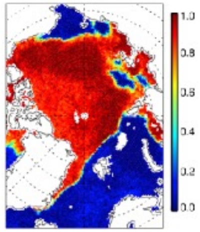
A trainable empirical-physical network for sea ice assimilation

AMSR2 observations

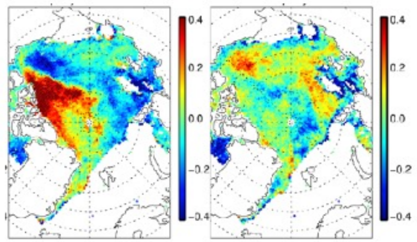


$$J_{\text{obs}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(y_{\text{obs},ij} - y_{\text{sim},ij})^2}{r_j^2}$$

Map of sea ice fraction to be estimated



Maps of empirical parameters representing **unknown** sea ice state including microstructure

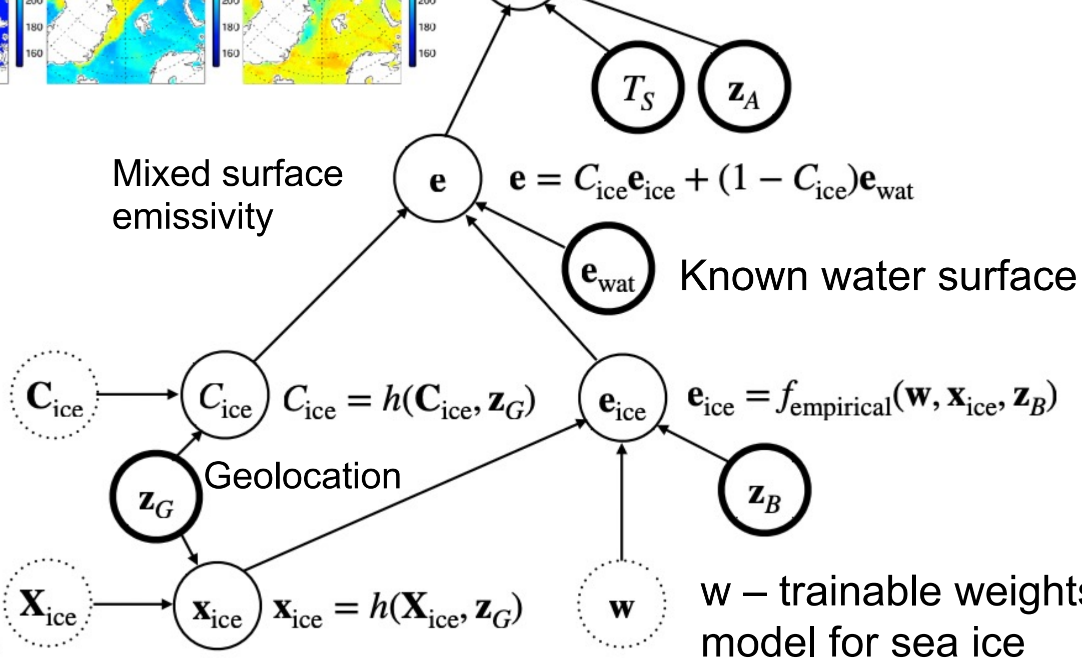


Known atmosphere radiative transfer

Fixed parameter

Dependent parameter

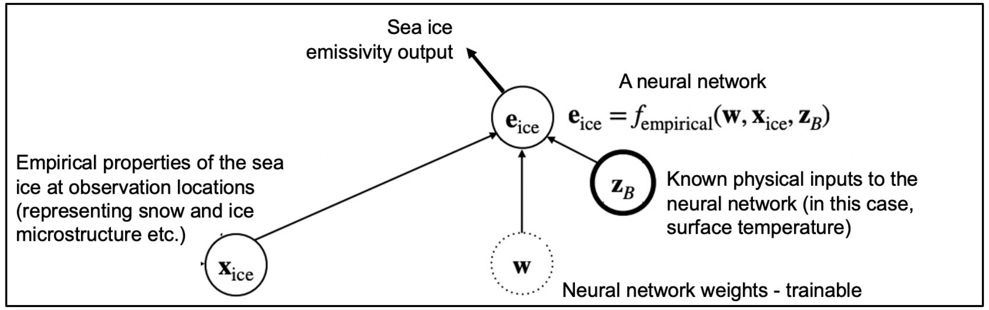
Trainable parameter



w – trainable weights of NN model for sea ice

h() Interpolation operator: map to observation location in time and space

Built in Python and Tensorflow




```python
class SeaiceEmis(tf.keras.layers.Layer):
    """
    Linear dense layer representing the sea ice emissivity empirical model.

    The sea ice loss applies to just the first mean emissivity (e.g. channel 10v); it's a single number as required.
    """
    def __init__(self, channels=10, bg_error=0.1, nobs=1, background=0.93):
        super(SeaiceEmis, self).__init__()
        self.dense_1 = tf.keras.layers.Dense(channels,activation='linear',bias_initializer=tf.keras.initializers.Constant(background))
        self.bg_error = bg_error
        self.background = background
        self.nobs = nobs
    def call(self, tsfc, ice_properties):
        inputs = tf.concat([tf.reshape(tsfc,(-1,1)),ice_properties],1)
        ice_emis = self.dense_1(inputs)
        emis_loss = tf.math.squared_difference((self.weights[1])[0],self.background)/tf.square(self.bg_error)/self.nobs
        self.add_loss(emis_loss)
        self.add_metric(emis_loss,name='emis_loss',aggregation='mean')
        return ice_emis
```

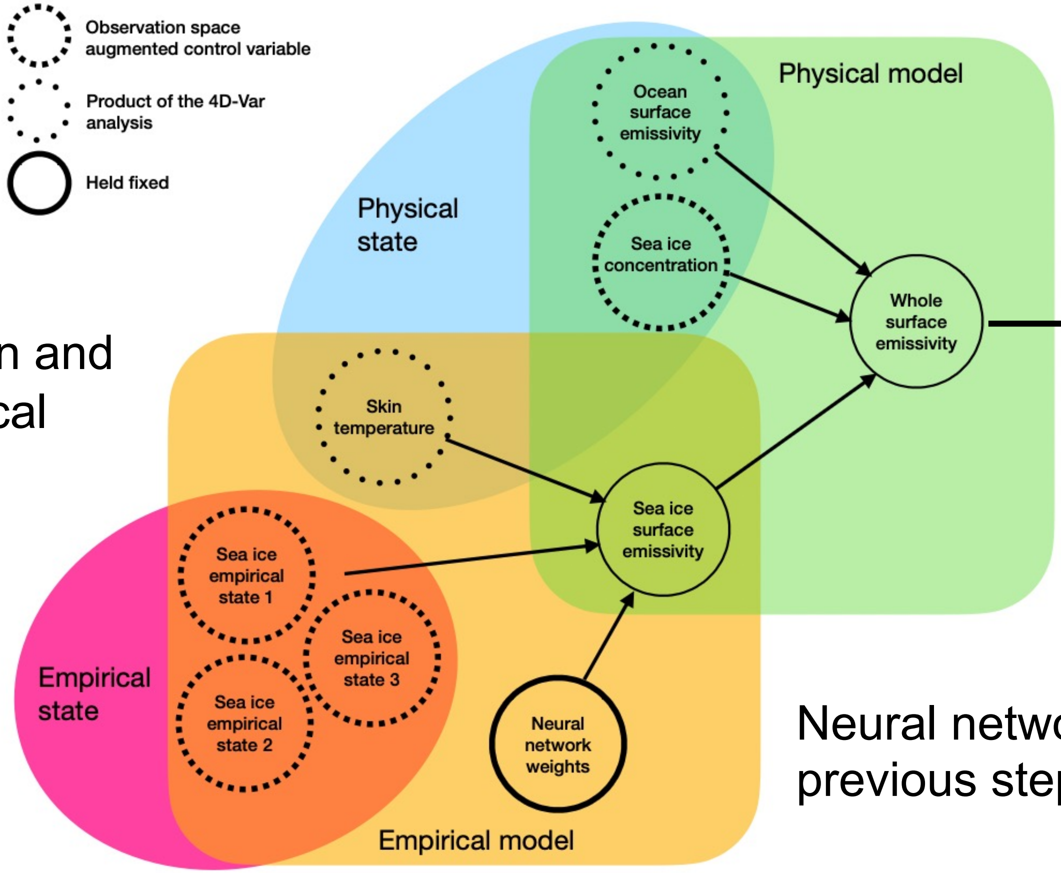
A standard dense neural network layer with linear activations

Custom loss functions to regularise / constrain the solution

https://github.com/ecmwf-projects/empirical-state-learning-seaice-emissivity-model/blob/master/seaice_layers.py

Empirical sea ice emissivity model used to retrieve sea ice concentration in atmospheric 4D-Var and to allow radiance assimilation over sea ice

Sea ice concentration and three empirical parameters retrieved at observation locations



RTTOV all-sky radiative transfer model

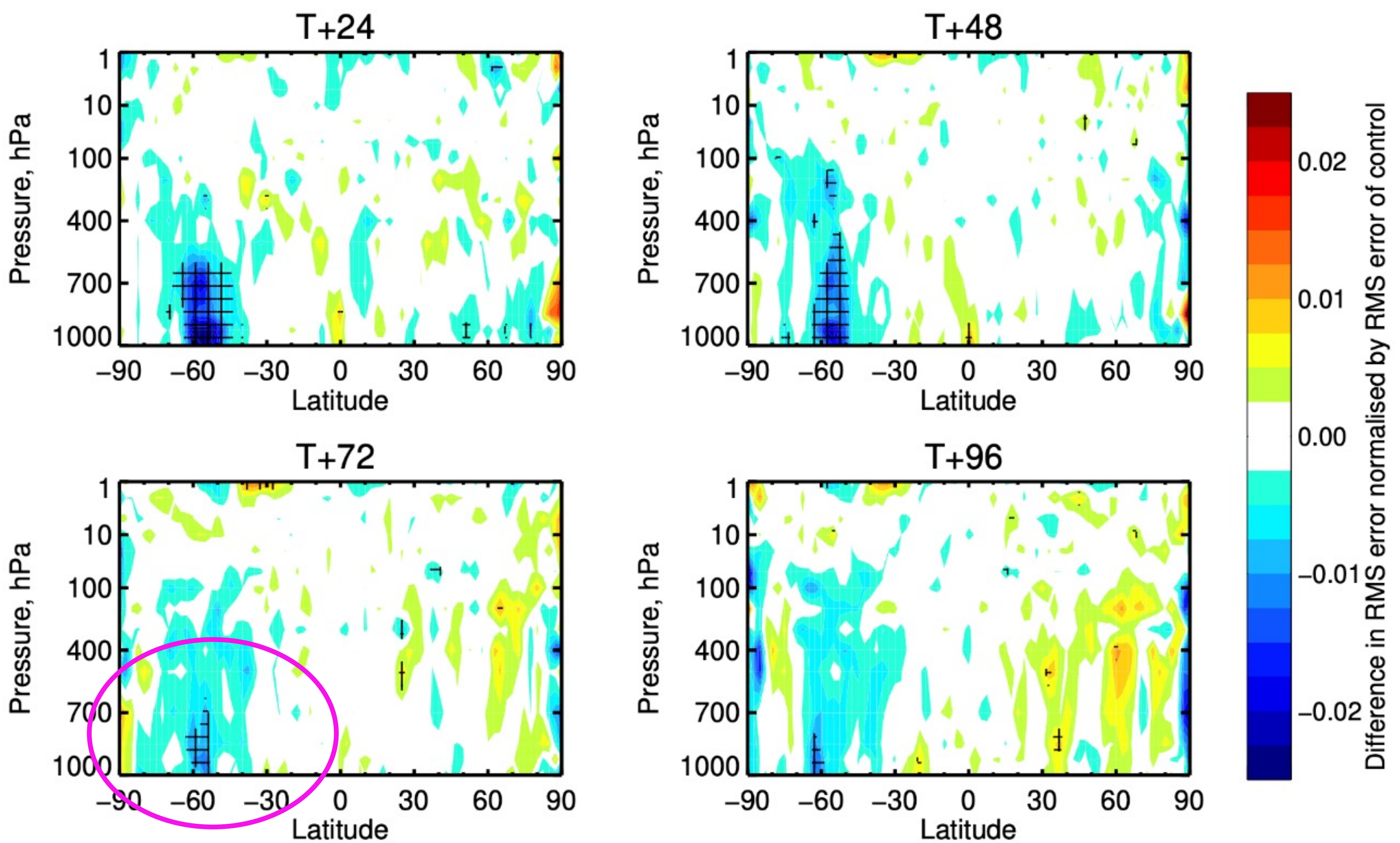
AMSR2 and GMI microwave imager observations

Atmosphere is also updated in 4D-Var, also with the benefit of all other observations, and can be improved by observations over sea ice

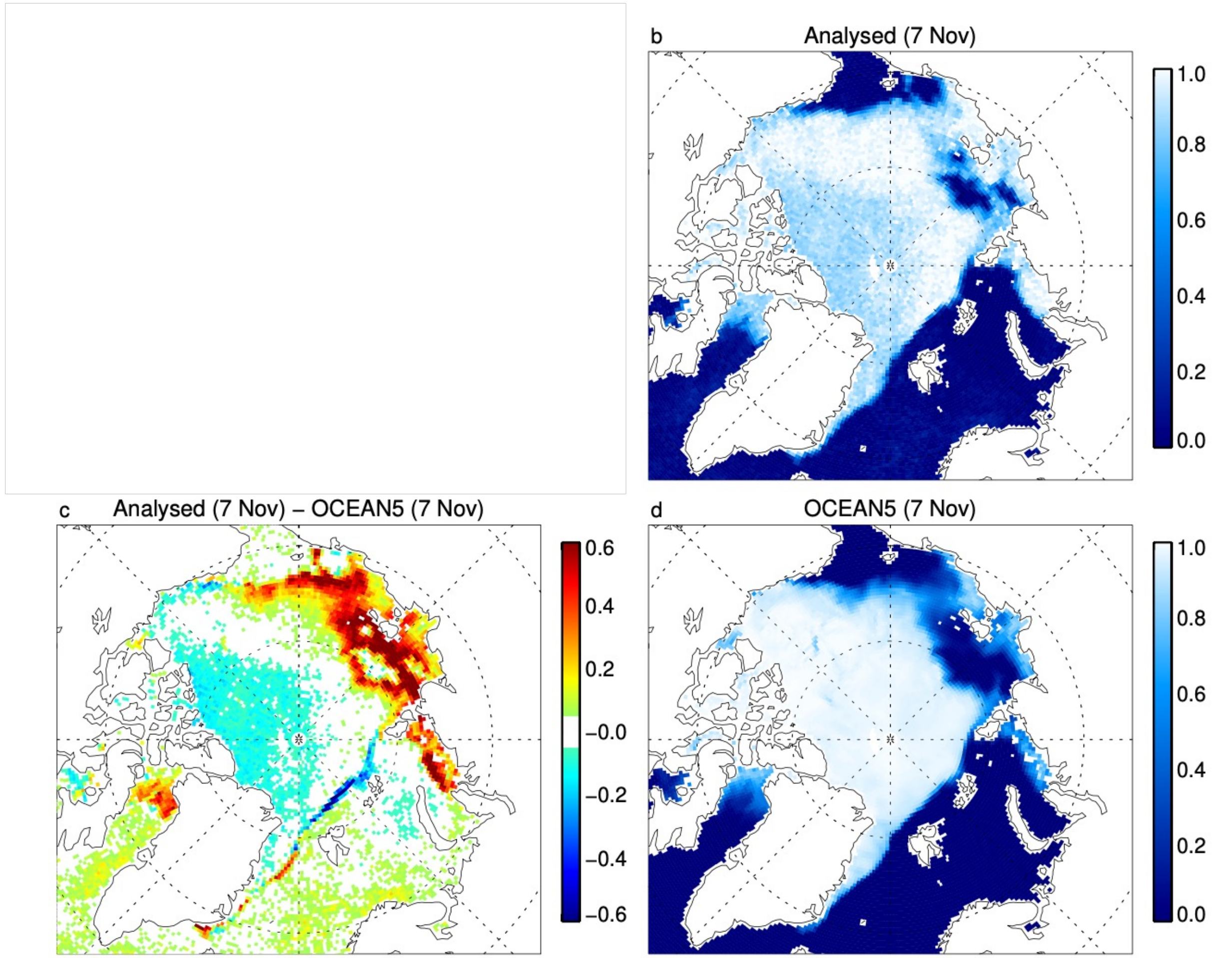
Neural network weights trained in previous step are now held fixed

Forecast impact - temperature
(blue = reduced error; +++ = statistical significance)

Improved temperature forecasts out to 72 hours in the Southern Ocean



Sea ice fraction retrieval: rapid freezing 7th Nov 2020



New retrieval from AMSR2 using a hybrid physical – empirical observation operator

Existing ECMWF sea ice analysis

Difference in sea ice concentration

Sea ice concentration

Summary: generating new empirical models using ML and DA

Machine learning with physical constraints

Machine learning and data assimilation: hybrid physical-empirical networks

Data assimilation with parameter estimation

- Typical machine learning and variational data assimilation are similar implementations of Bayes' theorem
- Including known physics into a trainable network is a way of adding prior information in a Bayesian sense
- Existing data assimilation approaches can be very helpful in machine learning:
  - Physically-based loss functions
  - Physically-based observation (label) and background (feature) errors
  - Observation operators to map from grid to irregular and transformed observation space (e.g. satellite radiances)
- Data assimilation frameworks (e.g. weather forecasting) are evolving to be able to train and update empirical models (e.g. neural networks) as part of routine data assimilation activities
  - – E.g. model error correction: don't throw away the physical model – improve it!
  - – E.g. assimilation of microwave observations sensitive to sea ice – a hybrid ML-DA component in cycle 49r1