Machine learning and weather forecasting: past, present and future?

Leif Denby, Danish Meteorological Institute, Copenhagen

18/7/2024, SMR3952, ICTP, Trieste

The past

(before the age of computing, what did we do to model the atmosphere and physical systems in general?)

Slides heavily inspired by talk by Weinan E (Princeton):
“AI for Science, and the implications for Mathematics” SIAM 2023 (Amsterdam)

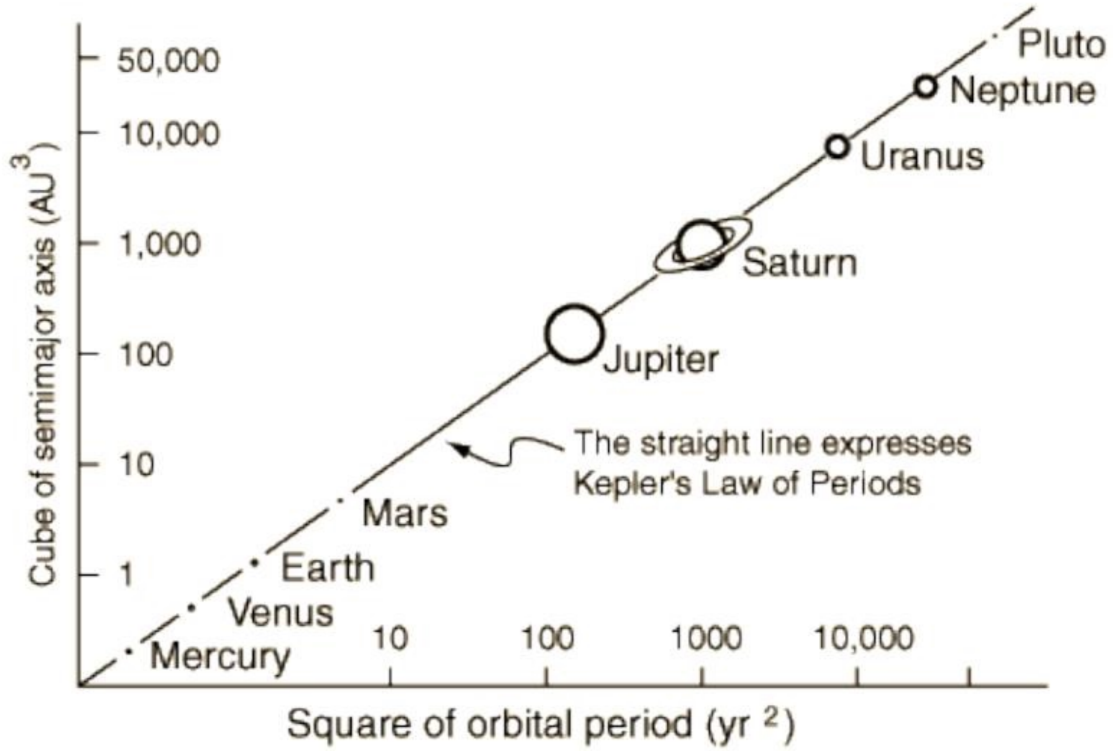
Taking a step back: Why do we do science?

- Find fundamental principles
    - laws of planet motion, thermodynamics, quantum mechanics
- Solve practical problems
    - engineering, industrial problems, e.g. weather and climate prediction

The Keplerian paradigm: data-driven approach

- Law's of planet motion
- Developel through purely data-driven means

Kepler’s third law



The Newtonian paradigm: search for first principles

- E.g. planet motion, start with Newton's laws:
  - Newton's 2nd Law: acceleration proportional to force
  - Law of gravitation: force inversely proportional to distance squared
- Reduce to ODE problem
  - solve ODE, get laws of planet motion

We mostly know the fundamental equations

- Paul Dirac (1929):

  "The underlying physical laws necessary for the mathematical theory of **a large part of physics and the whole of chemistry** are thus completely known, and the difficulty is only that the exact application of these laws leads to **equations much too complicated to be soluble**."

- We just need to solve the equations :)

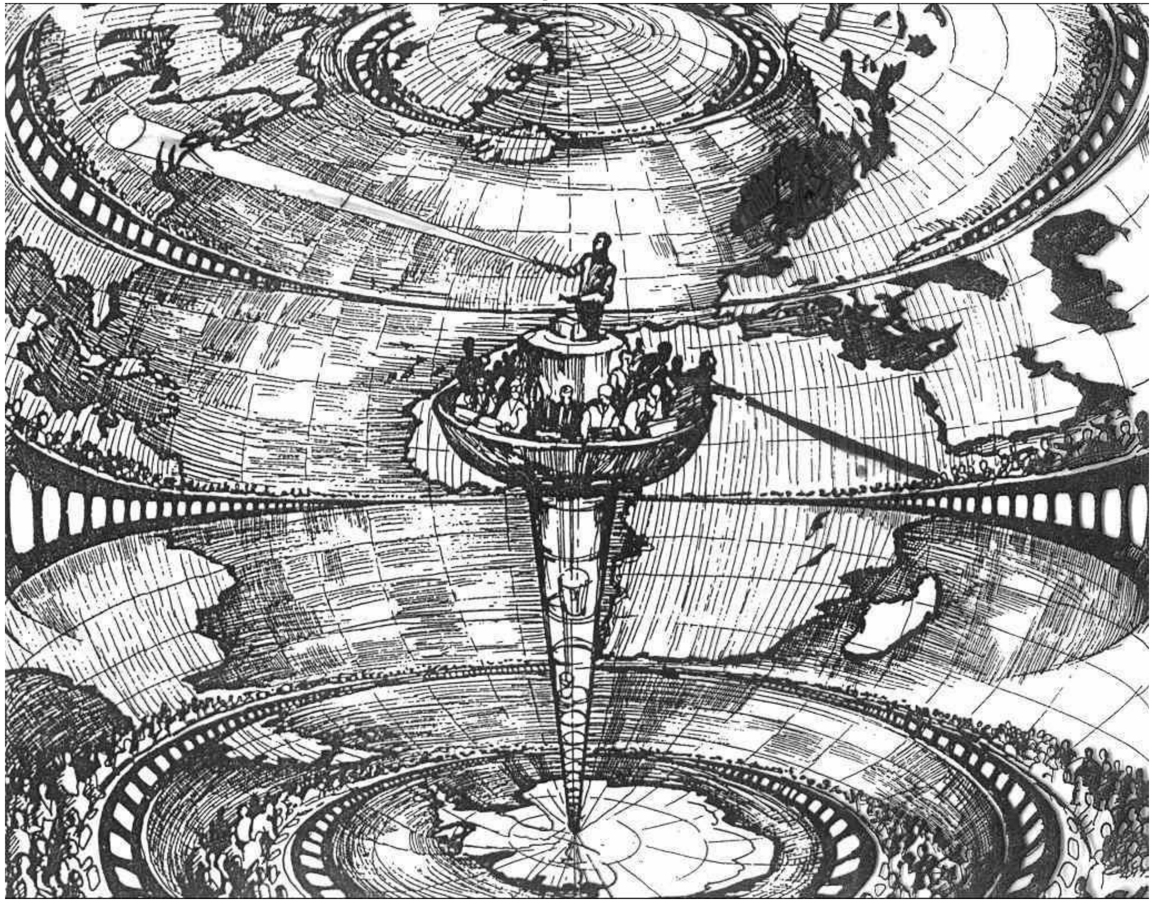
Hierarchy of physical models:
- Schrodinger equations (quantum mechanics)
- Navier-Stokes equations (fluid mechanics)
- Maxwell equations (electromagnetism)
- Boltzmann and Euler equations (gas dynamics)

Using the fundamental equations

- Good news:
  - All natural science and related engineering problems reduce to math problems (ODE/PDE problems)
- Bad news:
  - before effective math tools scientists had to simplify or ignore models to solve pratical problems

The first "weather prediction model" - Lewis Fry Richardson

With equations developed and approach developed by Abbe and Bjerkness, LF Richardson imagined a *Forecast Factory:*



"64,000 computers would be needed to race the weather for the whole globe. That is a staggering figure"

Richardson 1922: "Weather Prediction by Numerical Process"
Lynch 2008: "The origins of computer weather prediction and climate modeling"

The first “weather prediction model” - Lewis Fry Richardson

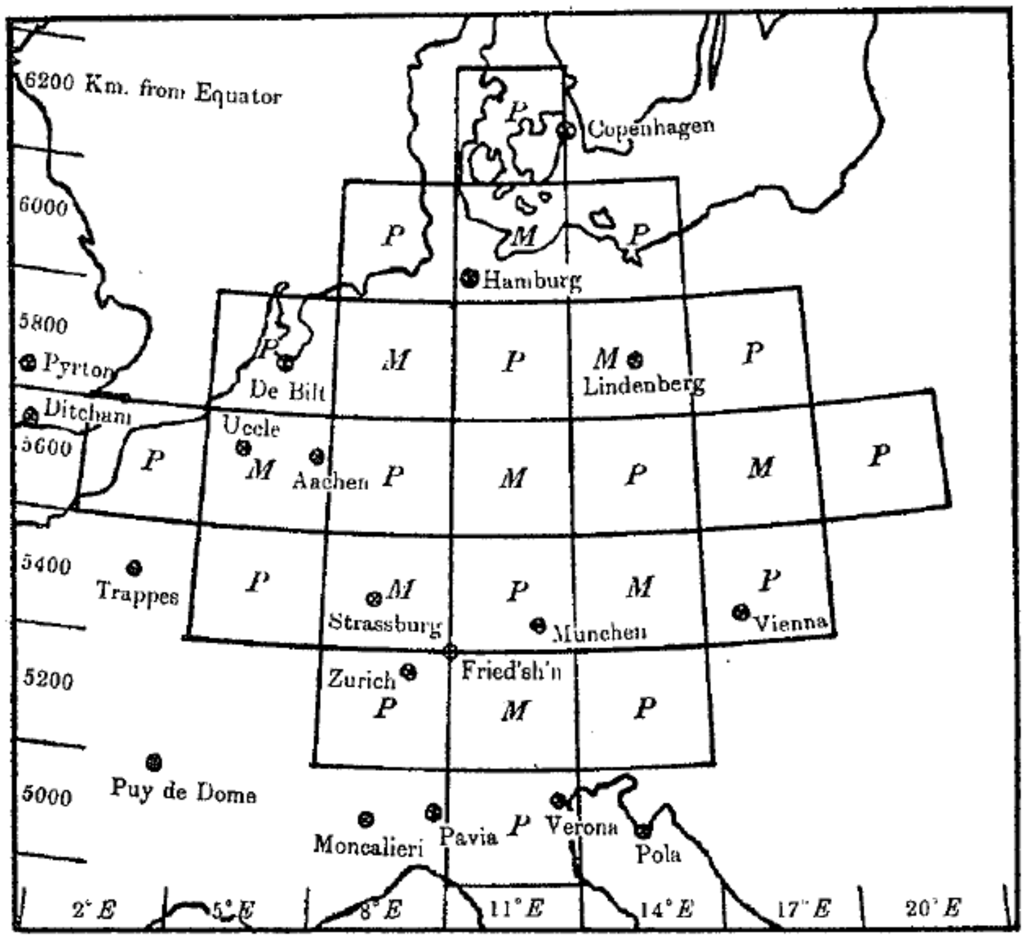


Figure 1.8 Forecast grid and observation stations for Richardson's experiment. (After Richardson 1922)

”Atmospheric Data Analysis”, R. Daley, Cambridge Univ. Press

Richardson completed the calculations **manually** using a numerical method that he devised.

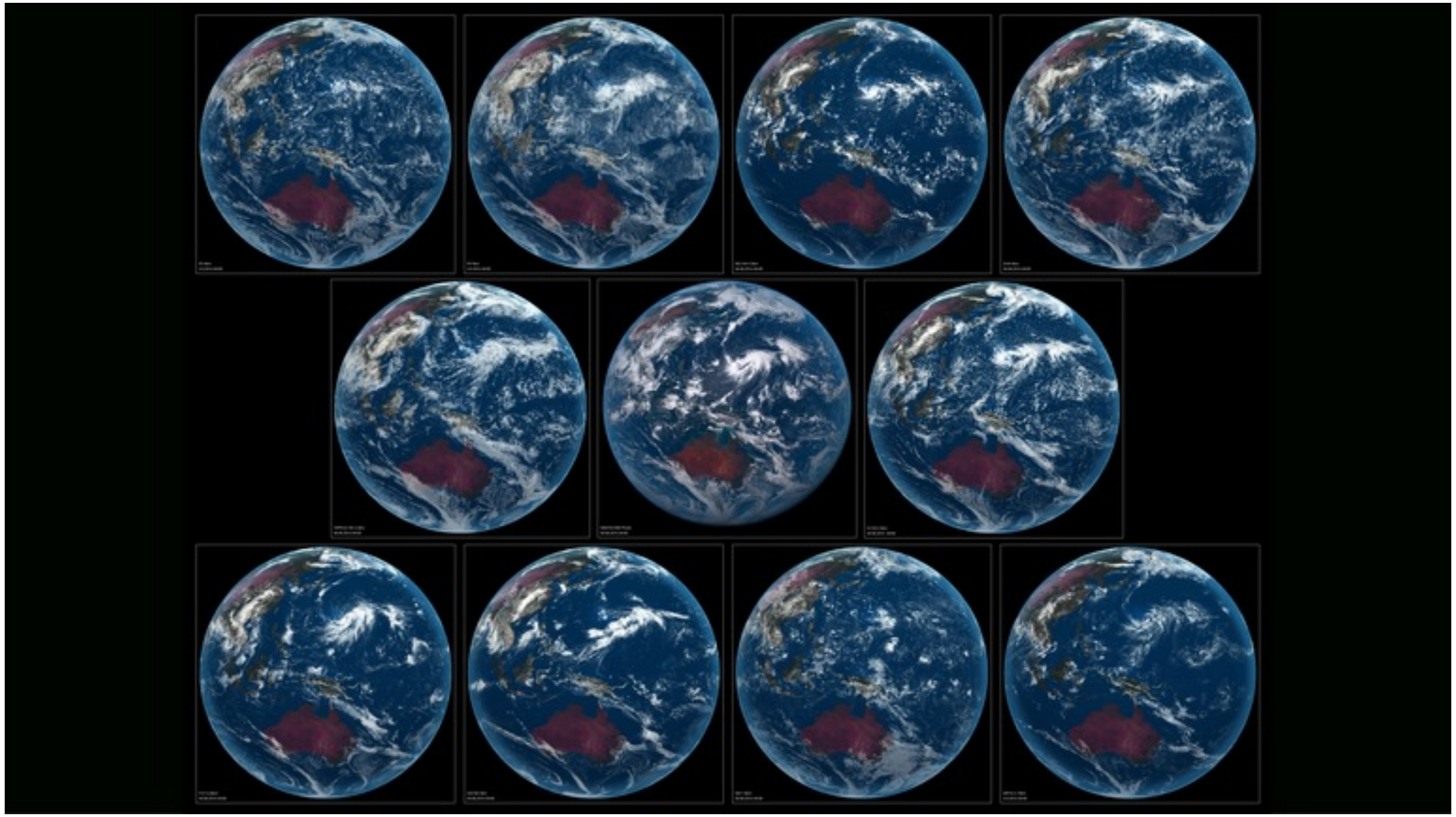
For various reasons his test, for part of Europe, failed, with huge deviations between forecast and observations.

The age of computing

- First major advance (von Neumann)
- Use of computers and numerical algorithms
  - Finite difference, finite element, spectral methods
  - Basic starting point: functions can be approximated by (piecewise) polynomials
- For the first time able to use fundamental principles to solve practical problems systematically
- Substantial impact
  - Modern engineering design, weather forecasting, etc

The present

The present



DYAMOND initiative: global storm-resolving ($\Delta x < 4$km) run for 40 days

$O(10^{12})$ scalar values for a single timestep

The challenge

- Many problems still remain not handled by fundamental principles
  - Material properties and design
  - Drug design
  - Turbulence, polymers
- Control problems
  - Theoretical work very challenging and separated from real world
  - Same happening in extension of computational applied maths to these fields

The challenge - The curse of dimensionality

- As dimensionality grows, complexity grows exponentially
  - In high dimension applications, (piecewise) polynomials are not efficient tools
- Mesh is too coarse
  - (10 billion points uniformly spaced in unit cube with 1000 dimensions, mesh size ~ 0.97723)
- Too many monomials
  - How many p^{th} order monomials in d dimensions?

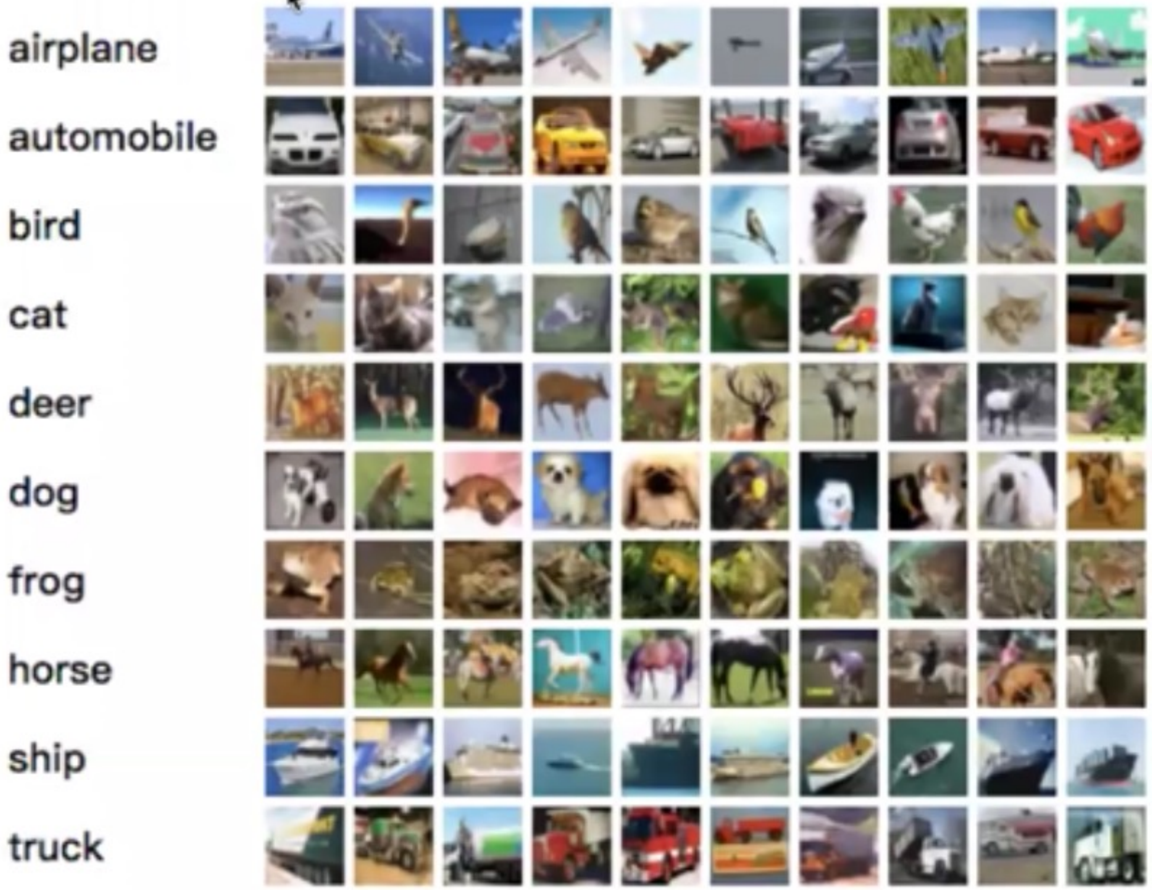
A high-dimensional problem: image classiciation

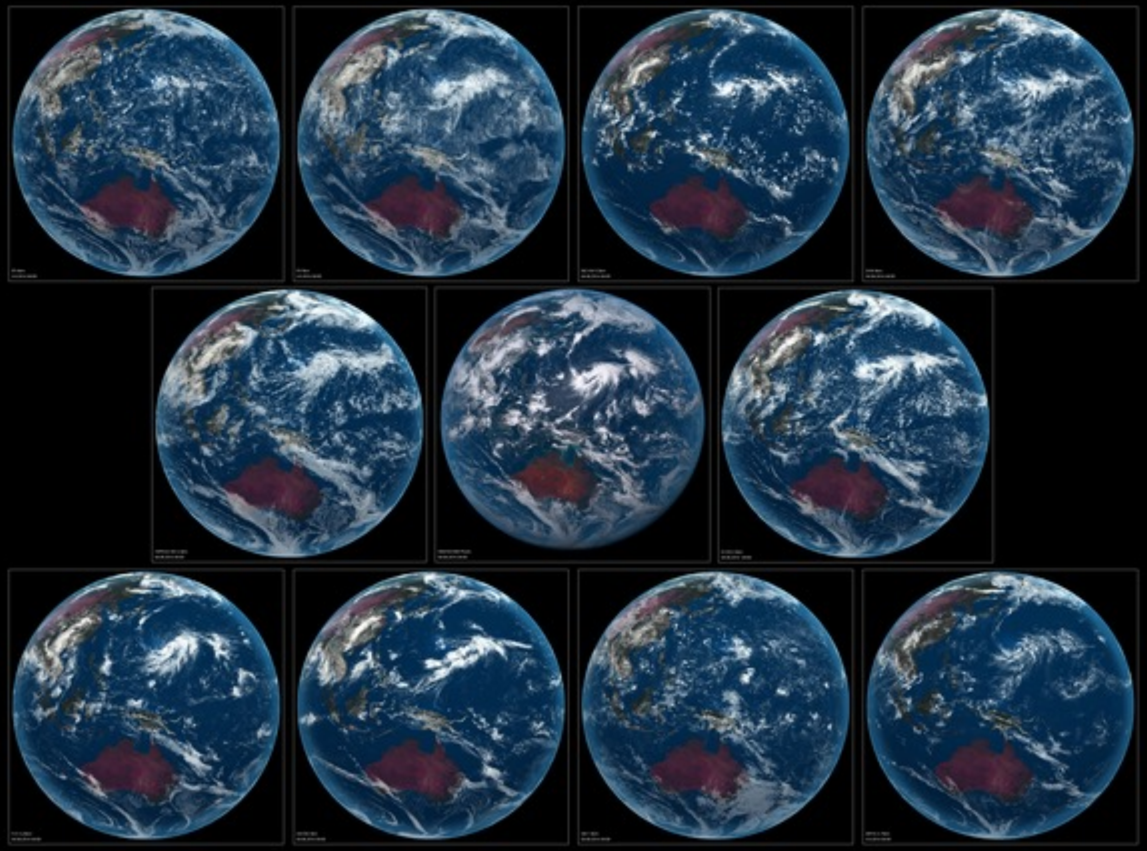
A high-dimensional problem: image classiciation

Counting the dimensionality of Cifar 10

Dim=32 x 32 x 3

= 3072

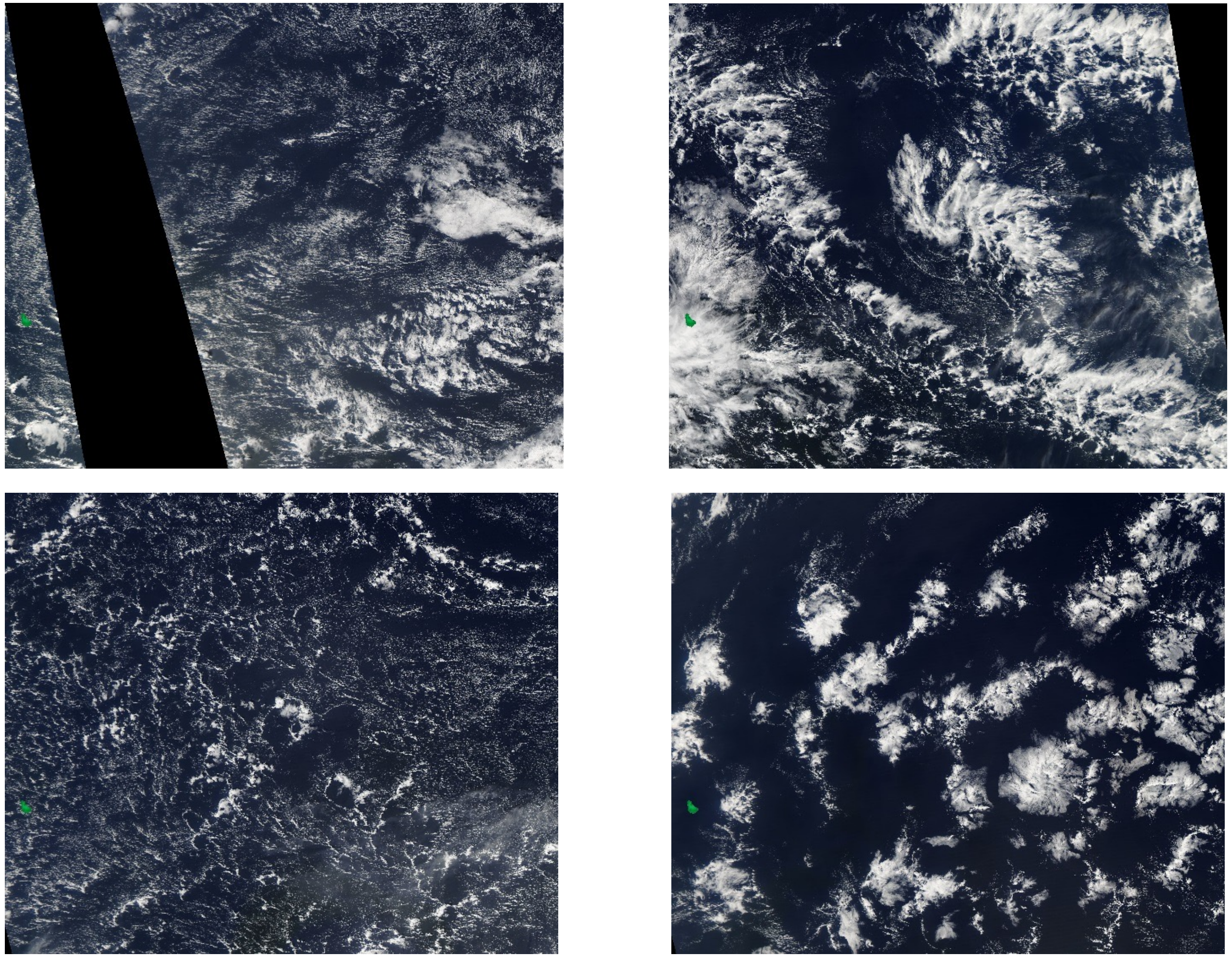




Interlude

Using self-supervised learning to study clouds

“Archetypes” of convective organisation



“sugar”

“fish”

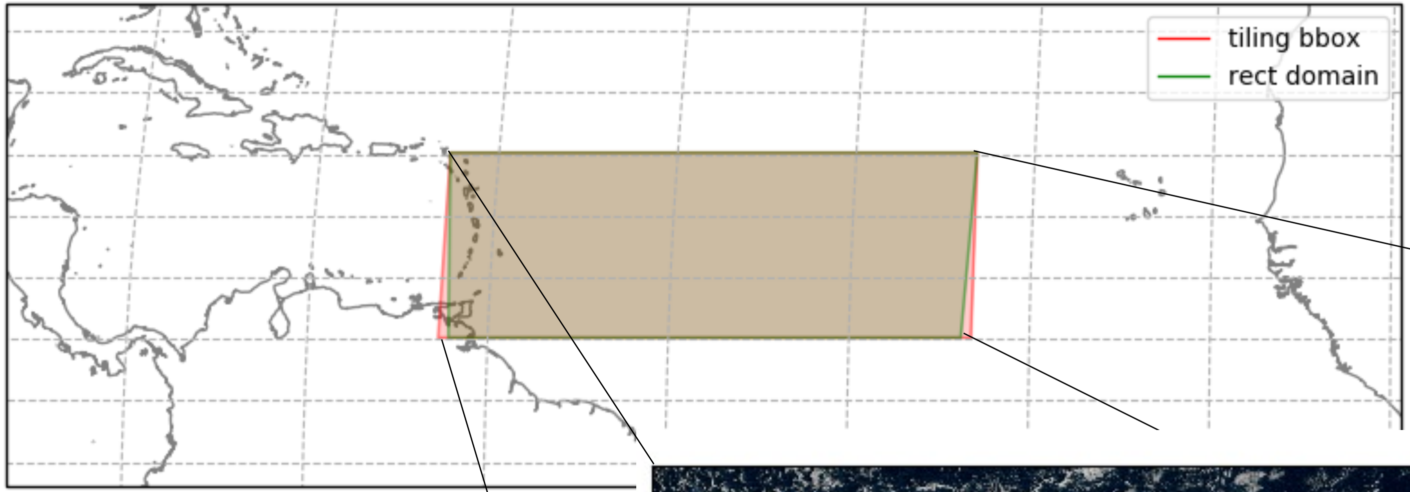
“gravel”

“flower”

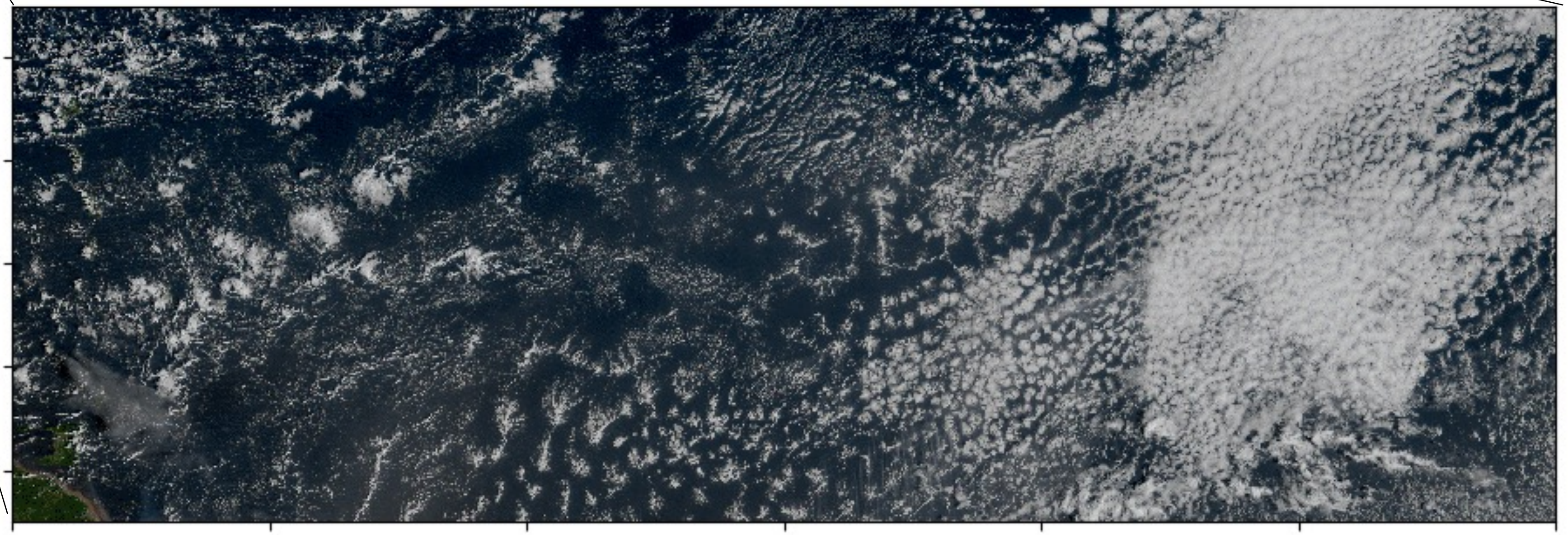
Stevens et al 2020, QJRMS

What happens between the “archetypes”?

Are they all that exist?

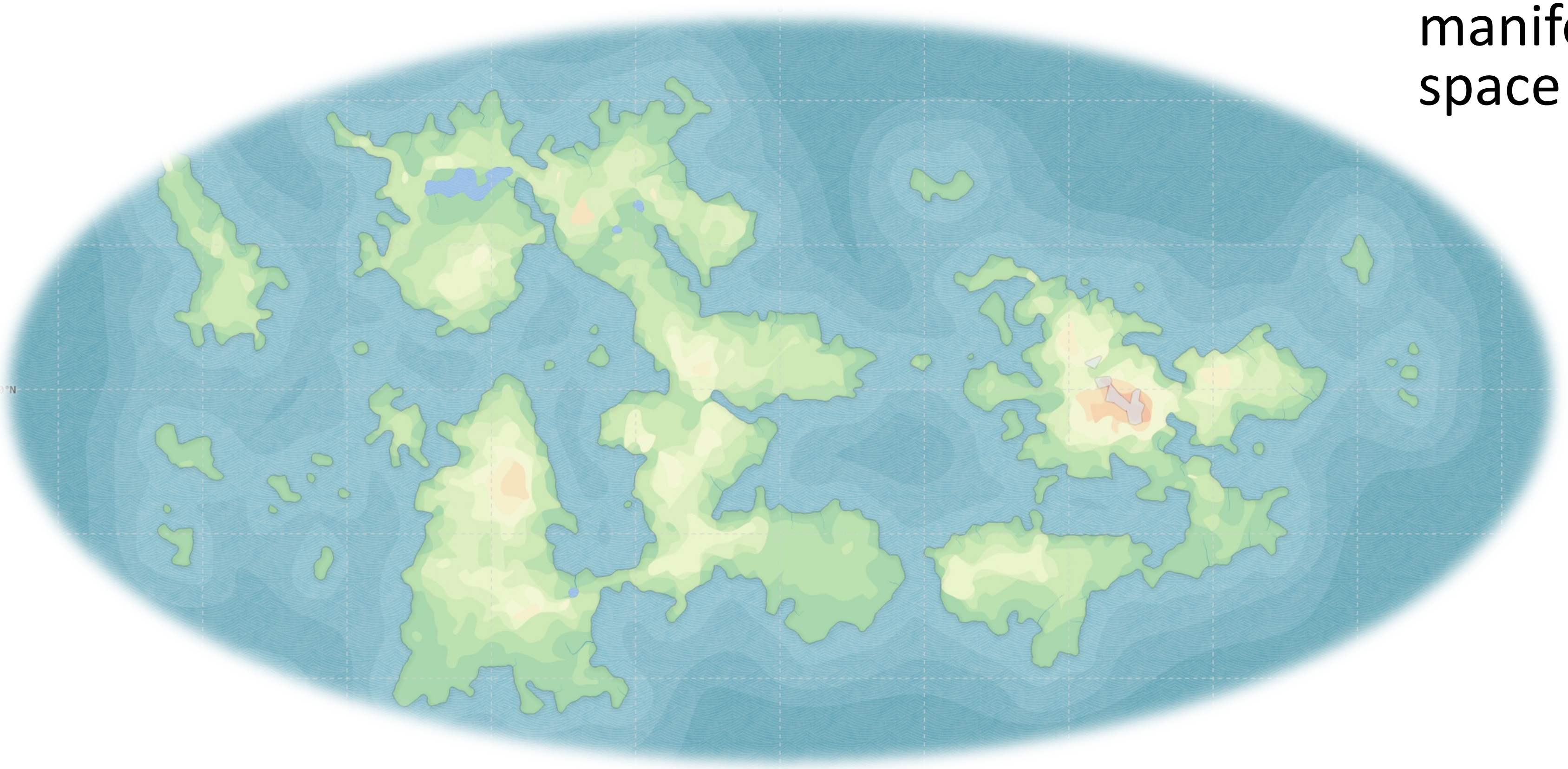


1000km meridional and 3000km zonal width local Cartesian reprojection centered on (lat, lon) = (14, -48) in tropical Atlantic



truecolor RGB composite from GOES-16 from daytime on 2nd Feb 2020

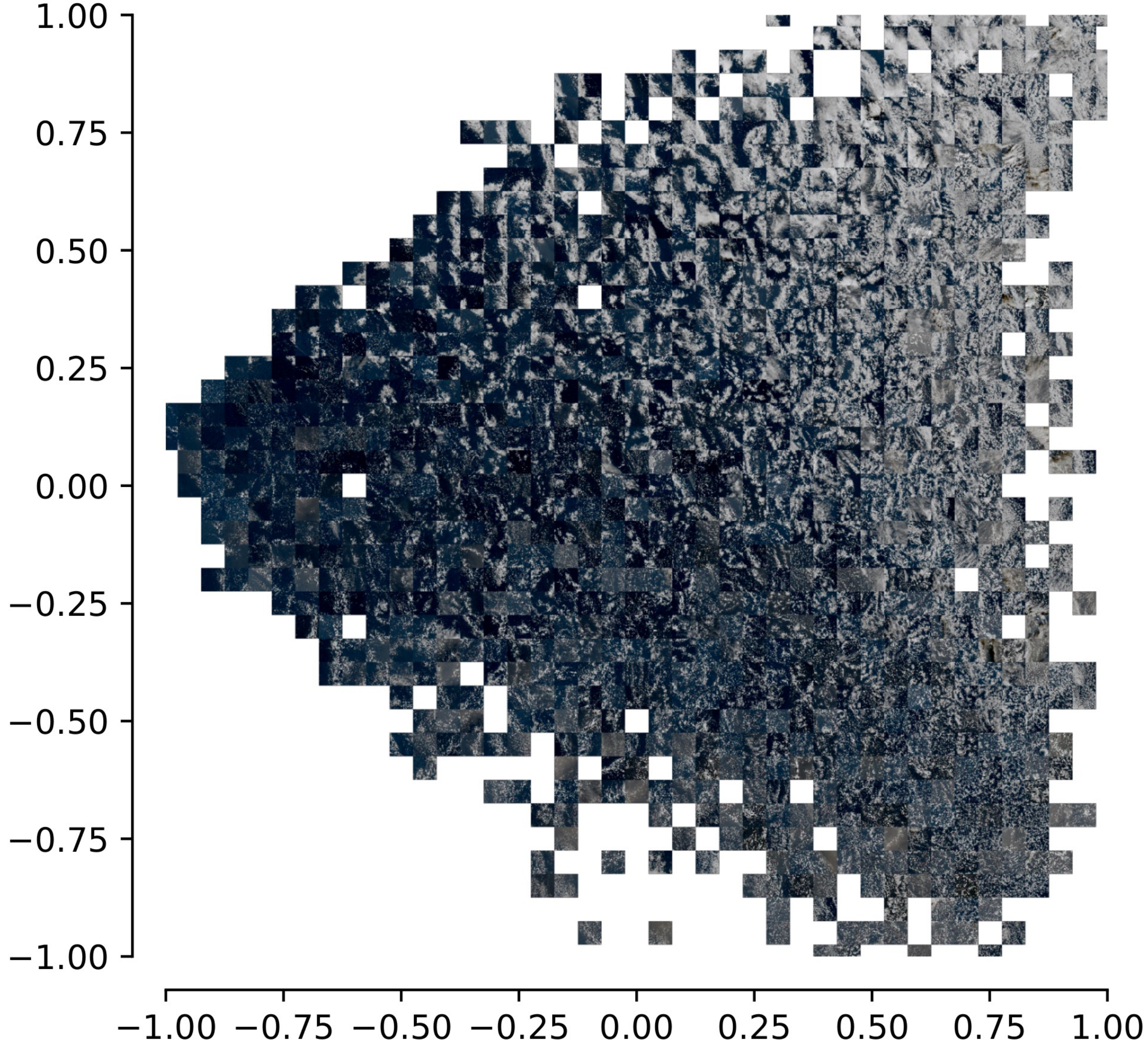
Extracting the embedding manifold



- Idea: maybe all the tile embeddings lie on some manifold in the embedding space

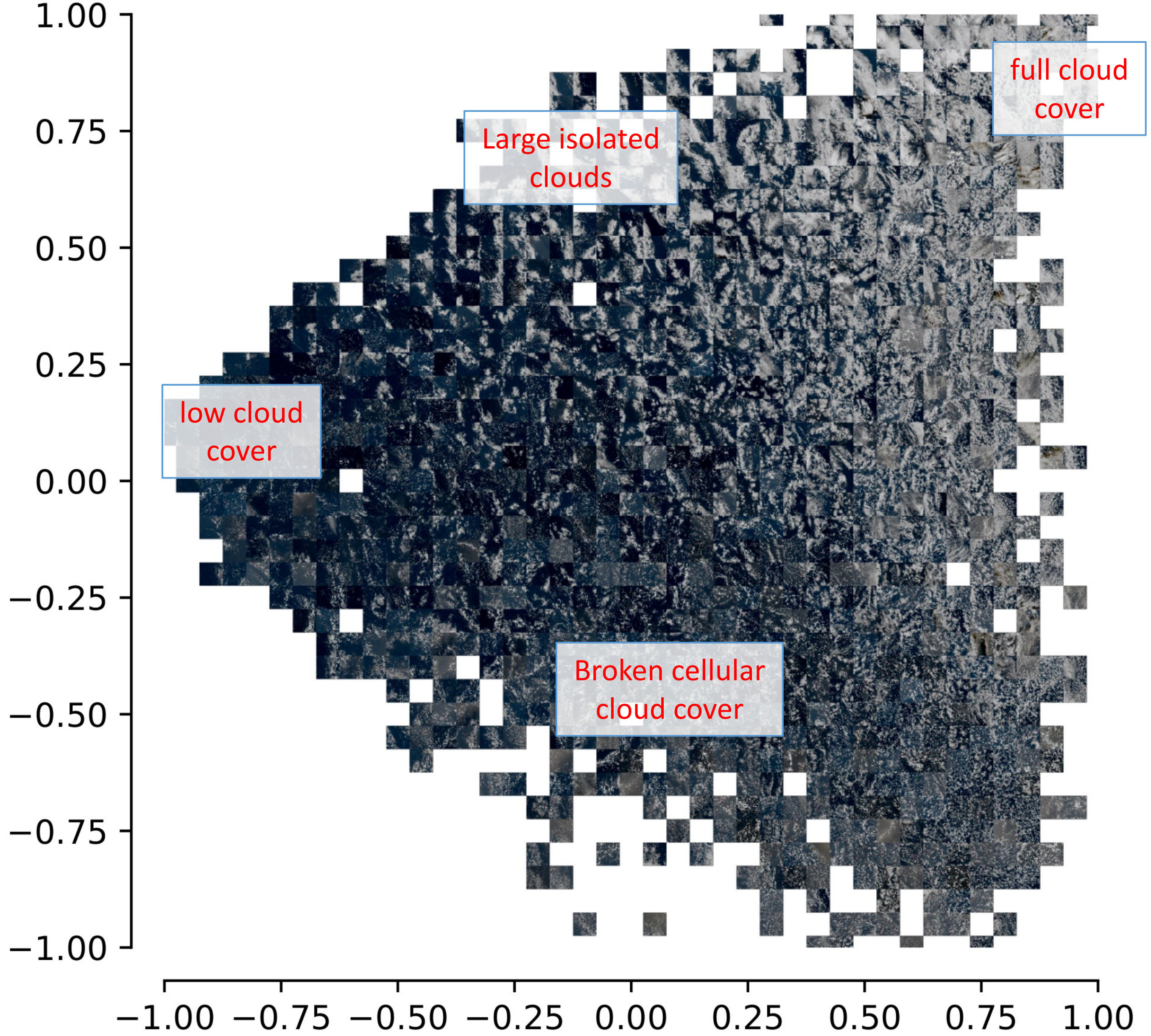
What does the world of cloud organisation look like?

Extracting the embedding manifold

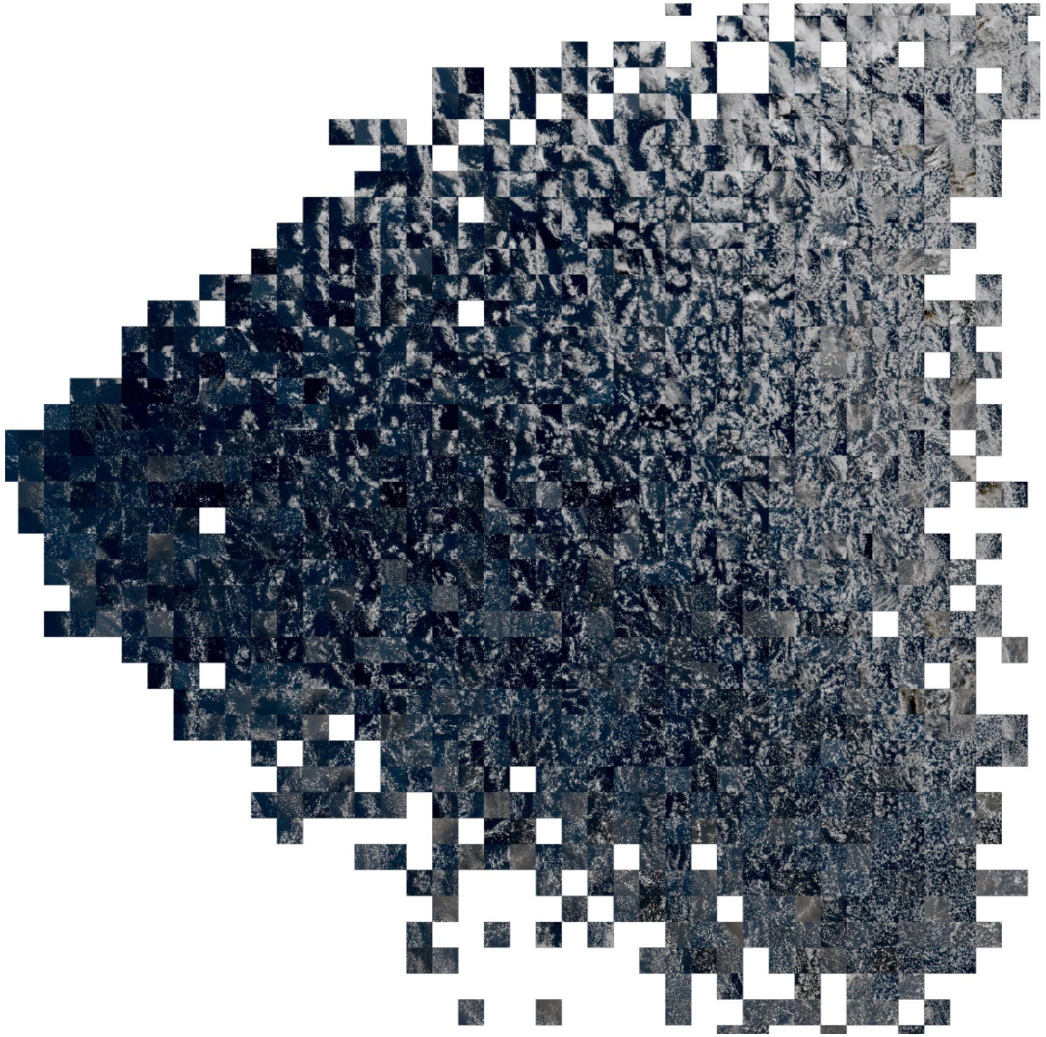


- Idea: maybe all the tile embeddings lie on some manifold in the embedding space
- Use Isomap method (Tenenbaum et al 2000) to extract manifold in high-dimensional embedding space and map to 2D
  - *“Isomap seeks a lower-dimensional embedding which maintains geodesic distances between all points”*
- With this I now have a “map” of all possible types of organisation

Extracting the embedding manifold

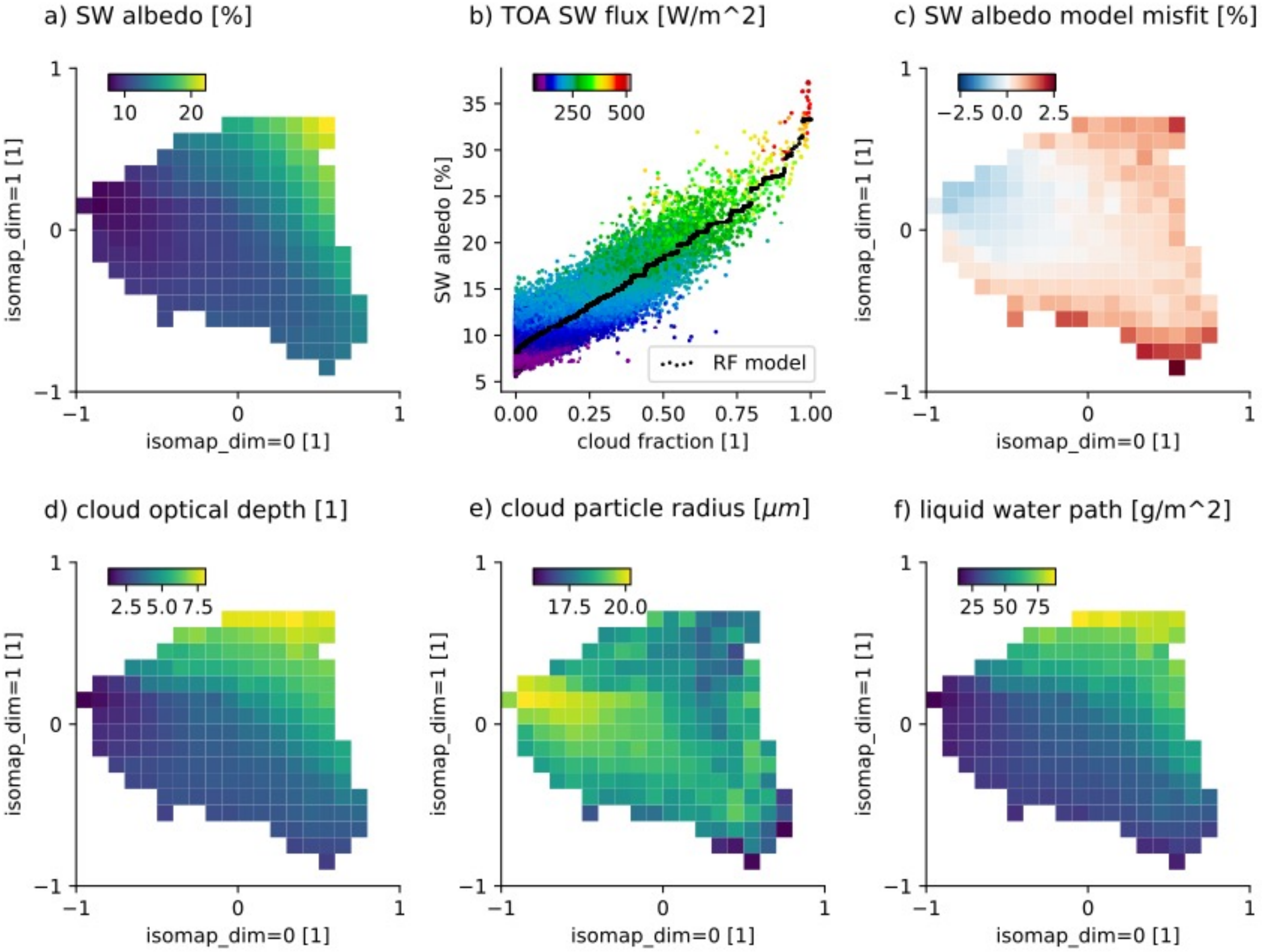


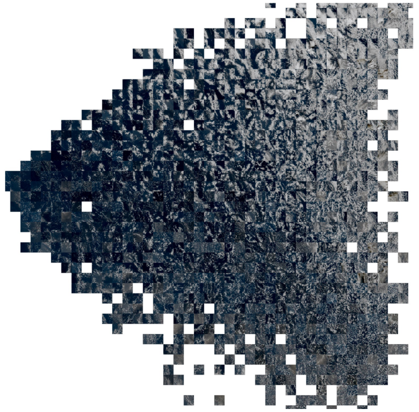
- Idea: maybe all the tile embeddings lie on some manifold in the embedding space
- Use Isomap method (Tenenbaum et al 2000) to extract manifold in high-dimensional embedding space and map to 2D
  - *"Isomap seeks a lower-dimensional embedding which maintains geodesic distances between all points"*
- With this I now have a "map" of all possible types of organisation



What can I do with this map of the world of cloud organisation?

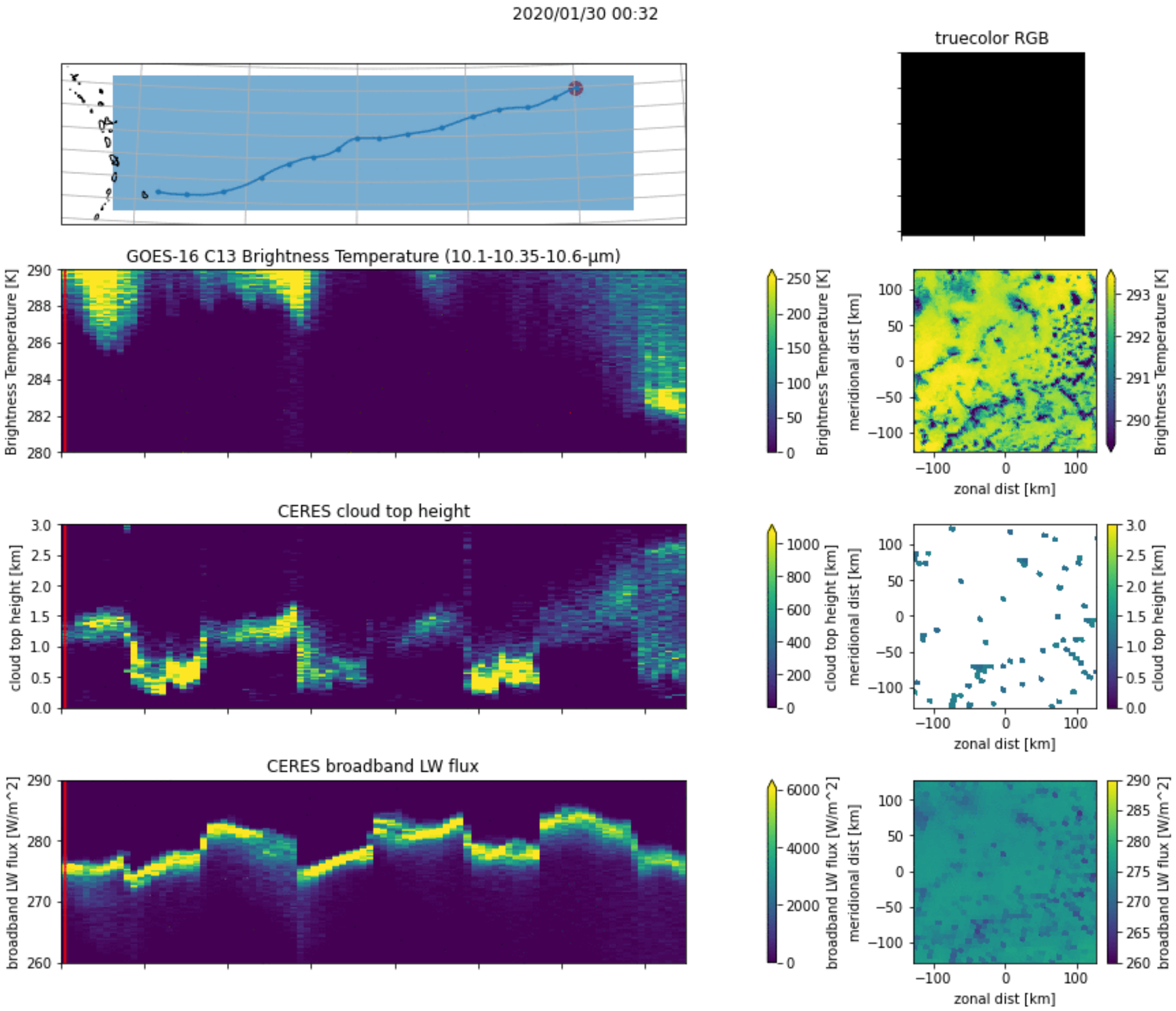
Radiative effects of cloud organisation





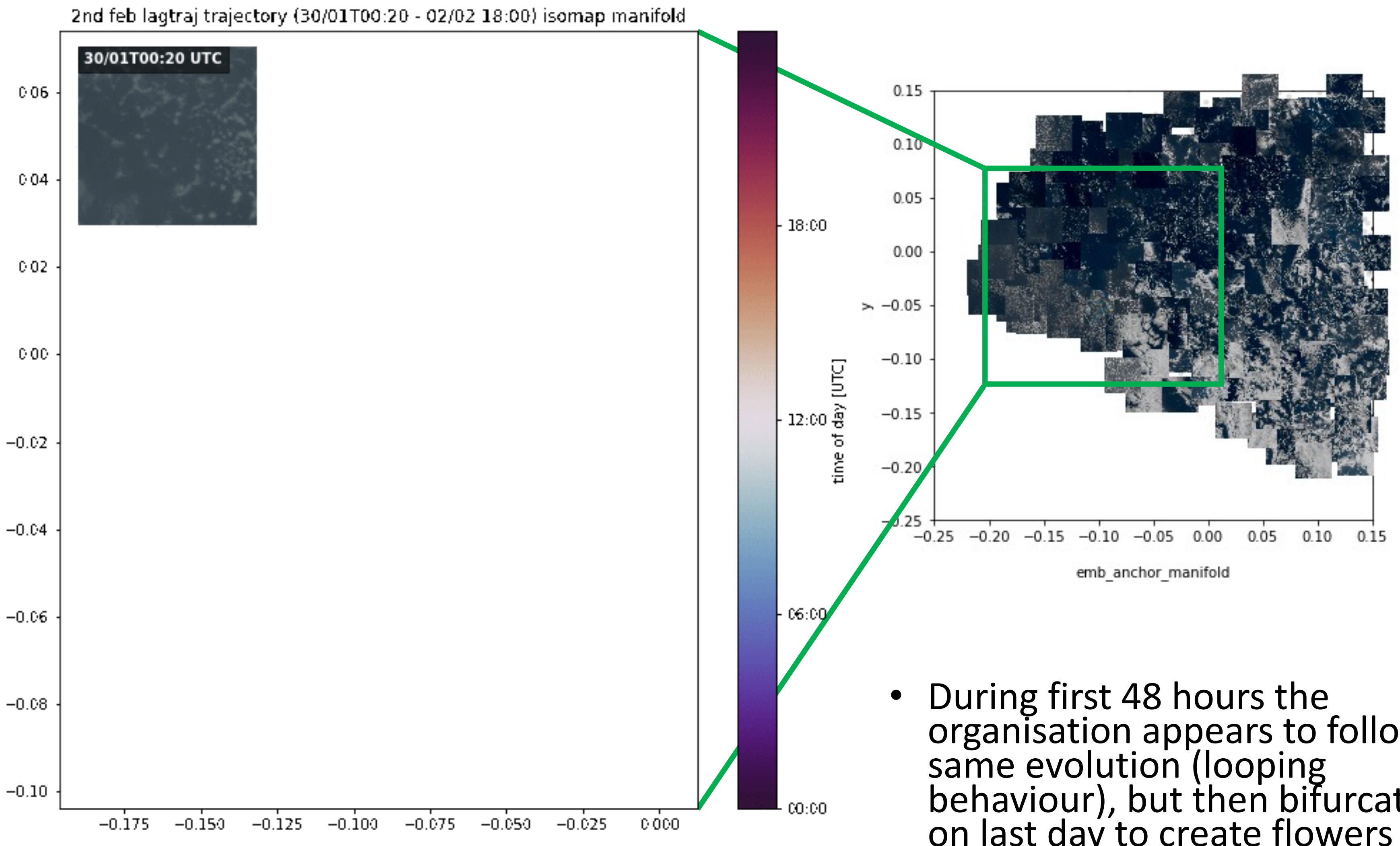
- isolated small cumuli lowest optical depth (larger cloud droplets and lower liquid water path)
    - lower SW albedo
- larger isolated cumuli have higher optical depth (smaller droplets and larger liquid water path)
    - higher SW albedo

But what about the evolution of organisation?



- Follow airmass along Lagrangian trajectory (from *lagtraj*) to capture evolution of organisation
  - Same trajectories that Steef Boeing and I are running Large-Eddy Simulations
- Flower organisation appears on very last day (!) first two days look very similar in terms of cloud-top height and organisation (by eye)

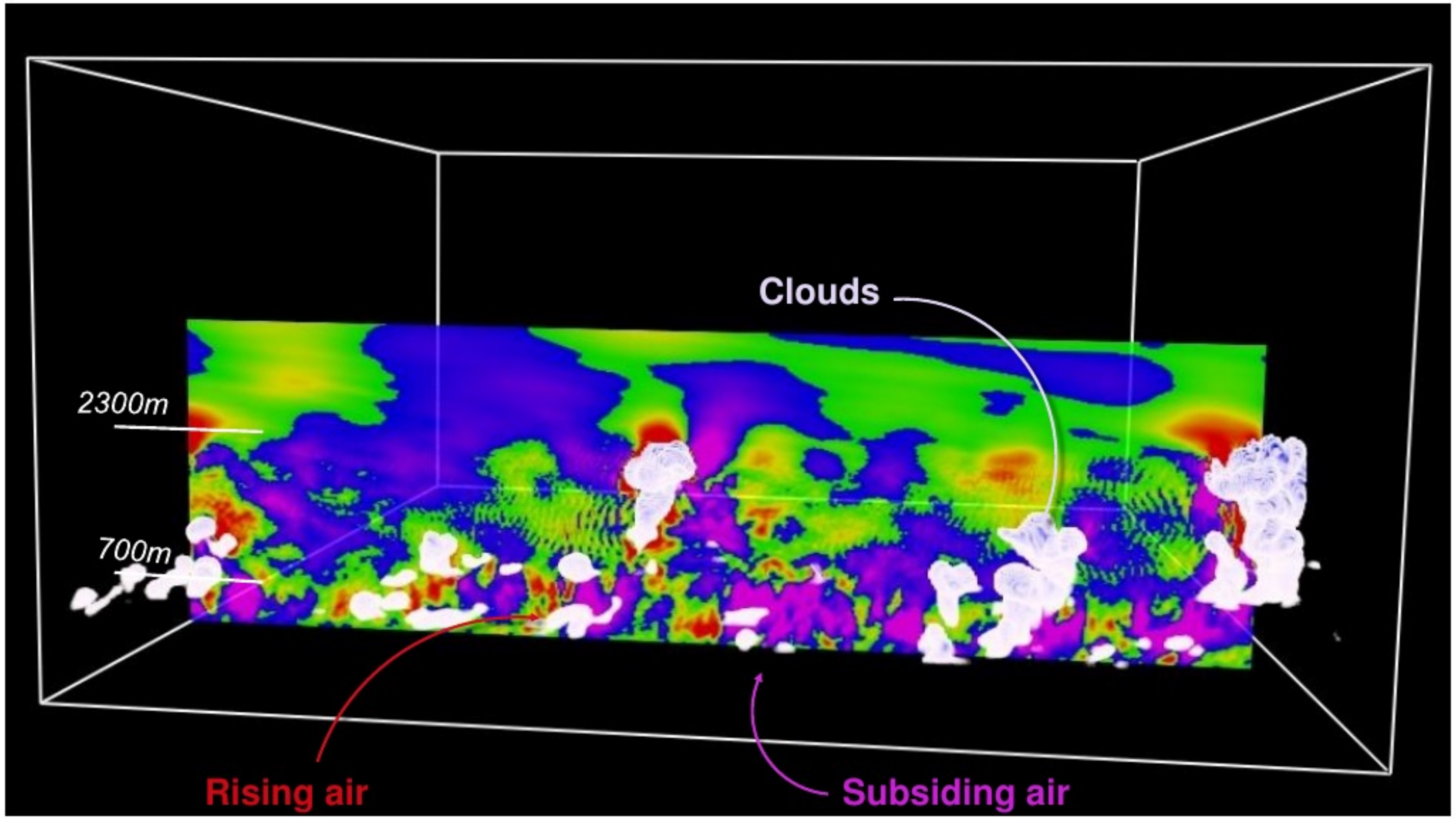
Mapping evolution of organisation



- Sample tiles along trajectory that is following clouds
  - Tiles created brightness temperature of IR channels in "water vapour window" (11, 14, 15)
- Use embeddings produced by neural network from tiles, to map evolution onto embedding manifold
  - Network trained on IR-triplets, covering tropical Atlantic domain over boreal winter

- During first 48 hours the organisation appears to follow same evolution (looping behaviour), but then bifurcates on last day to create flowers (!) What's happening here?

What does the boundary layer look like?
What are the structures that trigger these clouds?

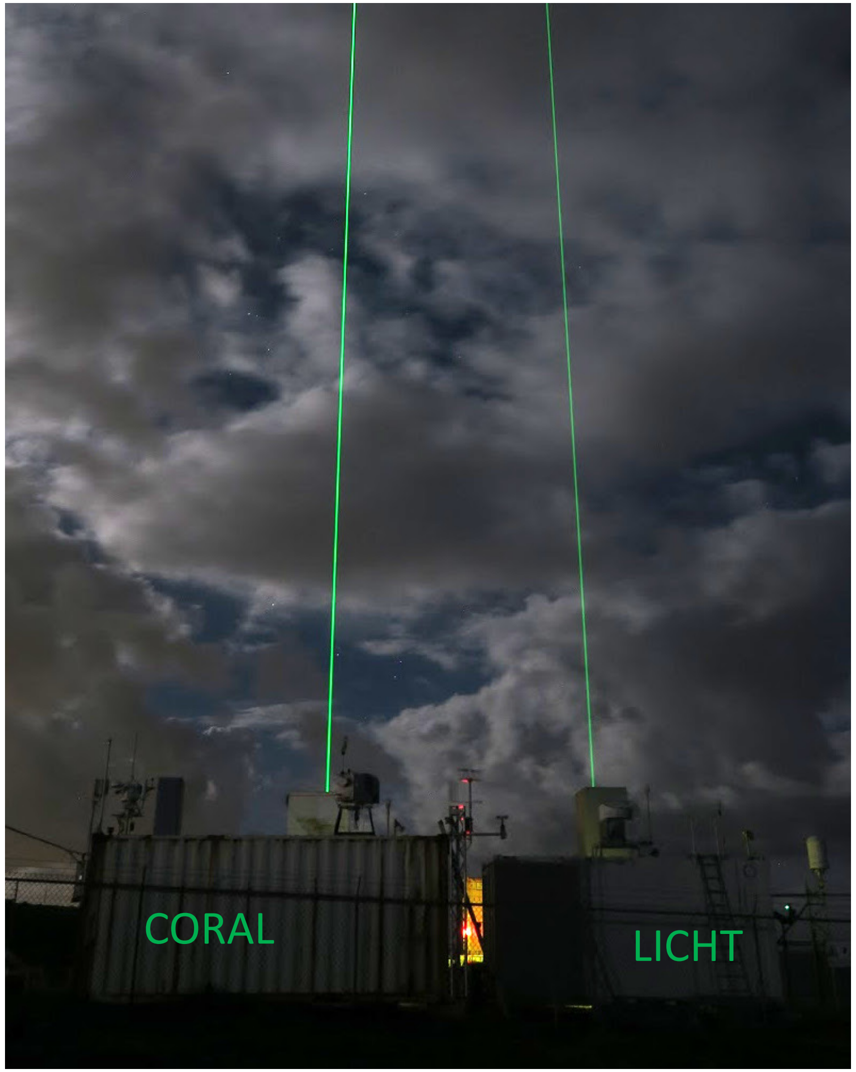


Δx=25m Large-Eddy Simulation, RICO test-case

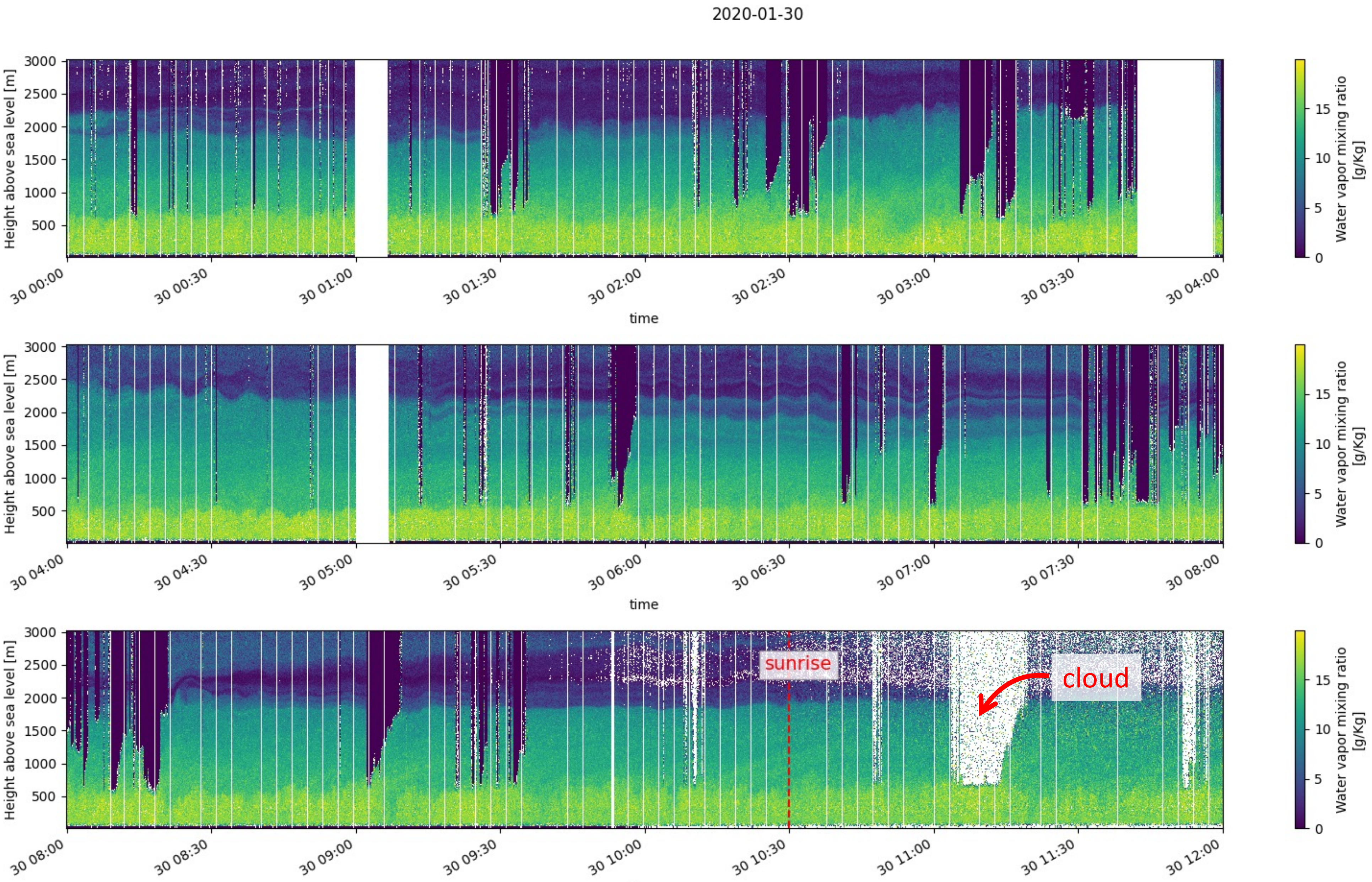
Rendered with VAPOR

How do I "see" these structures?
The Barbados Cloud Observatory CORAL Raman LIDAR

- Measure <u>water-vapour profiles</u> (below cloud), air temperature, aerosols and cloud properties.
- resolution:
  - horizontal wind: v ~ 5m/s
  - temporal resolution: Δt = 4s
  - => horizontal res: Δx ~ 20m
  - vertical res: Δz ~ 15m
- Developed and run by Ilya Serikov (MPI-Meteorlogy, Hamburg)

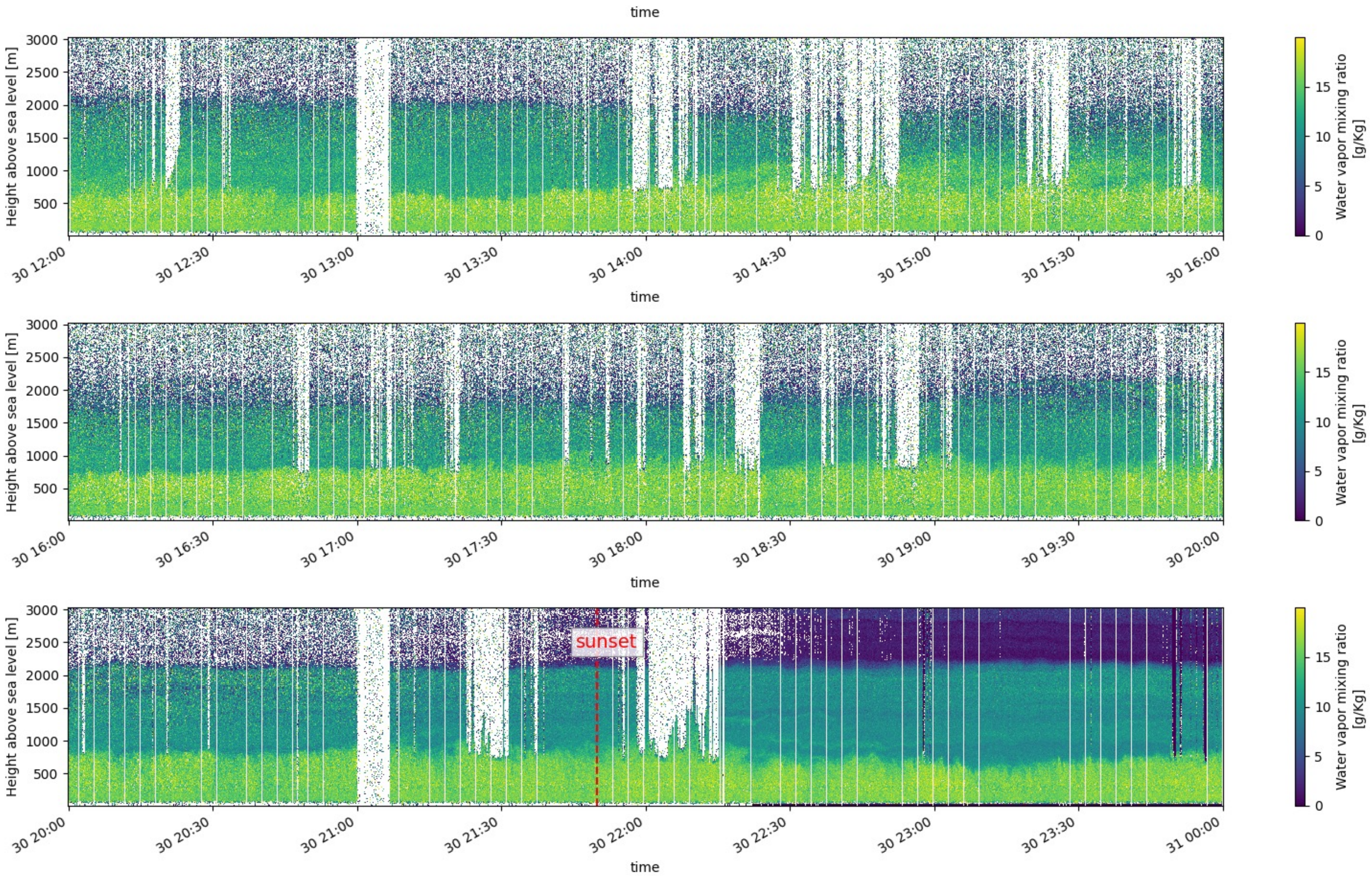


One day of LIDAR observations



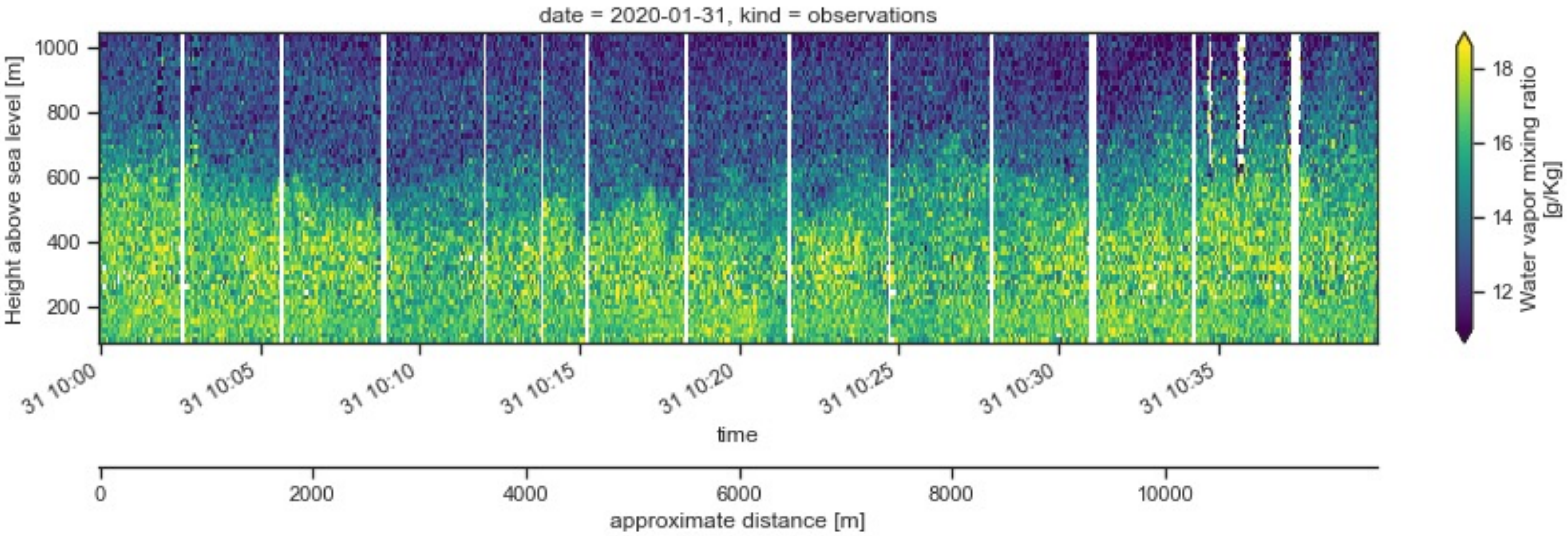
- Depth of mixed boundary layer clearly seen (~600m)
- Clouds block LIDAR, cloud-base at ~600m altitude
- More noise during daylight hours

One day of LIDAR observations - cont.



- Depth of mixed boundary layer clearly seen (~600m)
- Clouds block LIDAR, cloud-base at ~600m altitude
- More noise during daylight hours

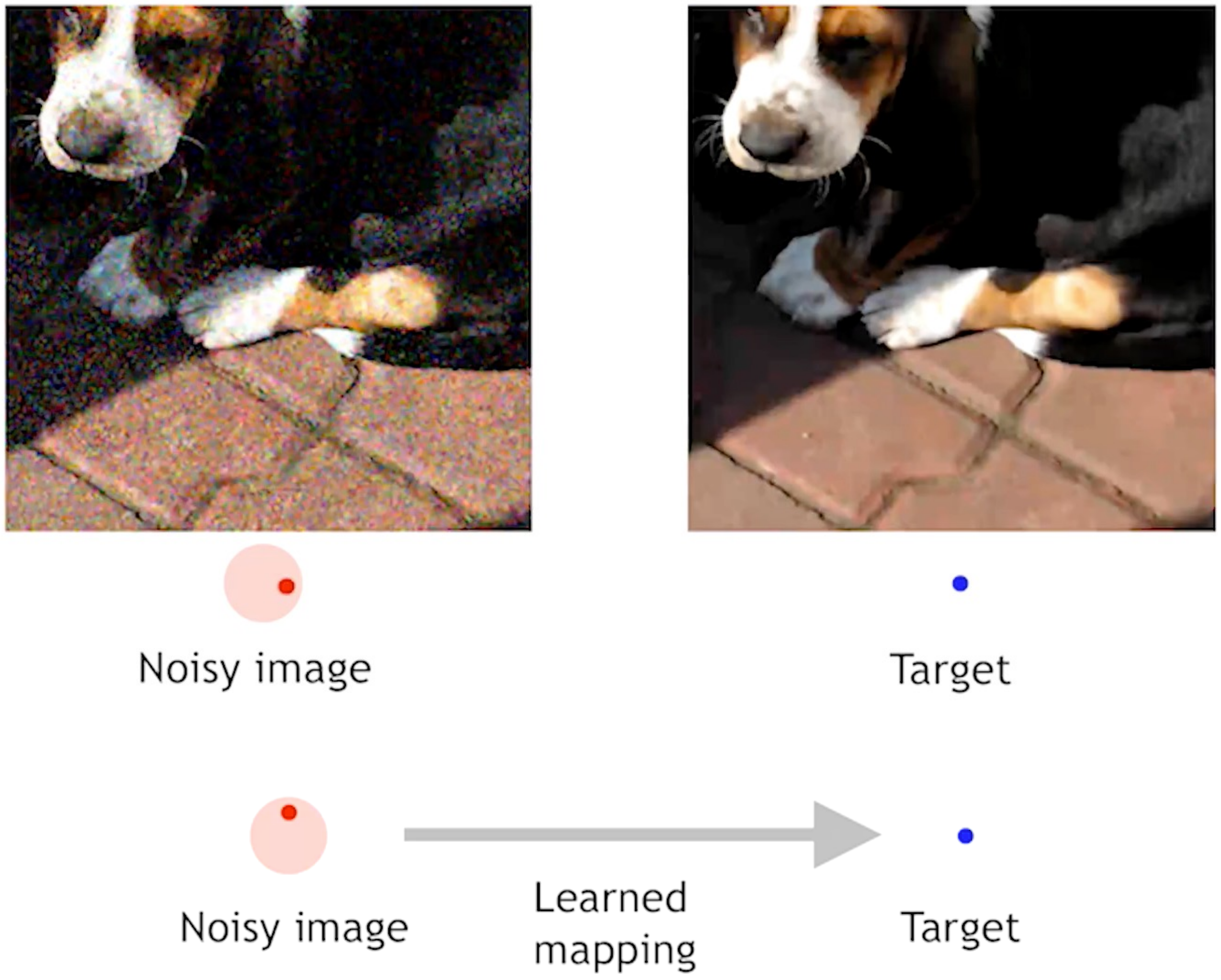
Denoising CORAL LIDAR water vapour profiles



- Although data is noisy (if you squint) individual coherent structures are visible
- Assuming ~ 5m/s wind speed these structures are on order of hundreds of meters

Traditional denoising with neural networks: supervised learning

- For supervised learning we need pairs of noisy input and clean target data, but for real-life observations we may not have clean data
  - Could *synthesize* training data using an assumed noise distribution applied to synthetic data - need simulated data and noise model
- Can I do something with just the noisy observations?



noise2void: Learning Denoising From Single Noisy Images
(Krull et al 2019)

- Assume noise at any two points in input is uncorrelated
- Exploit that image contains a high degree of structure
- Learn correction to point value from looking only at neighbouring pixels. Network forced to ignore central pixel by overwriting with random pixel in neighbourhood during training
  - If central pixel is included network simply learns identity
- Idea: if noise is uncorrelated then the only thing the network can learn from the context (surrounding) pixels is the true denoised value of a pixel

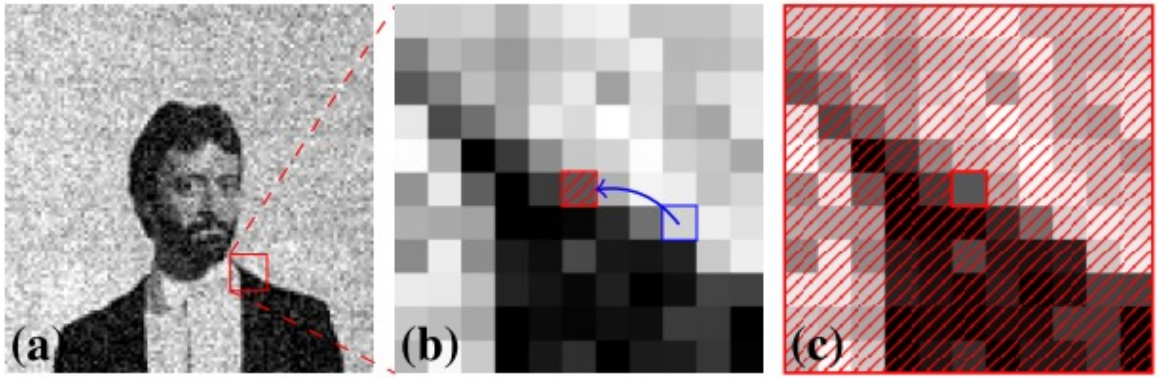
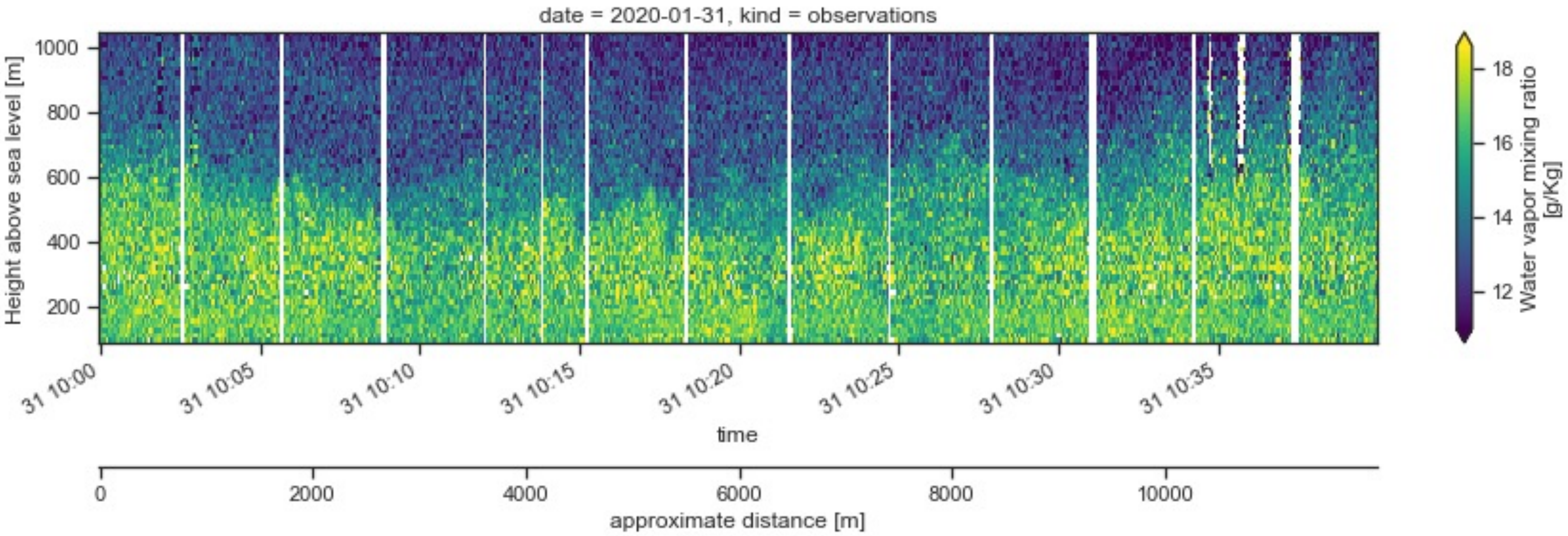


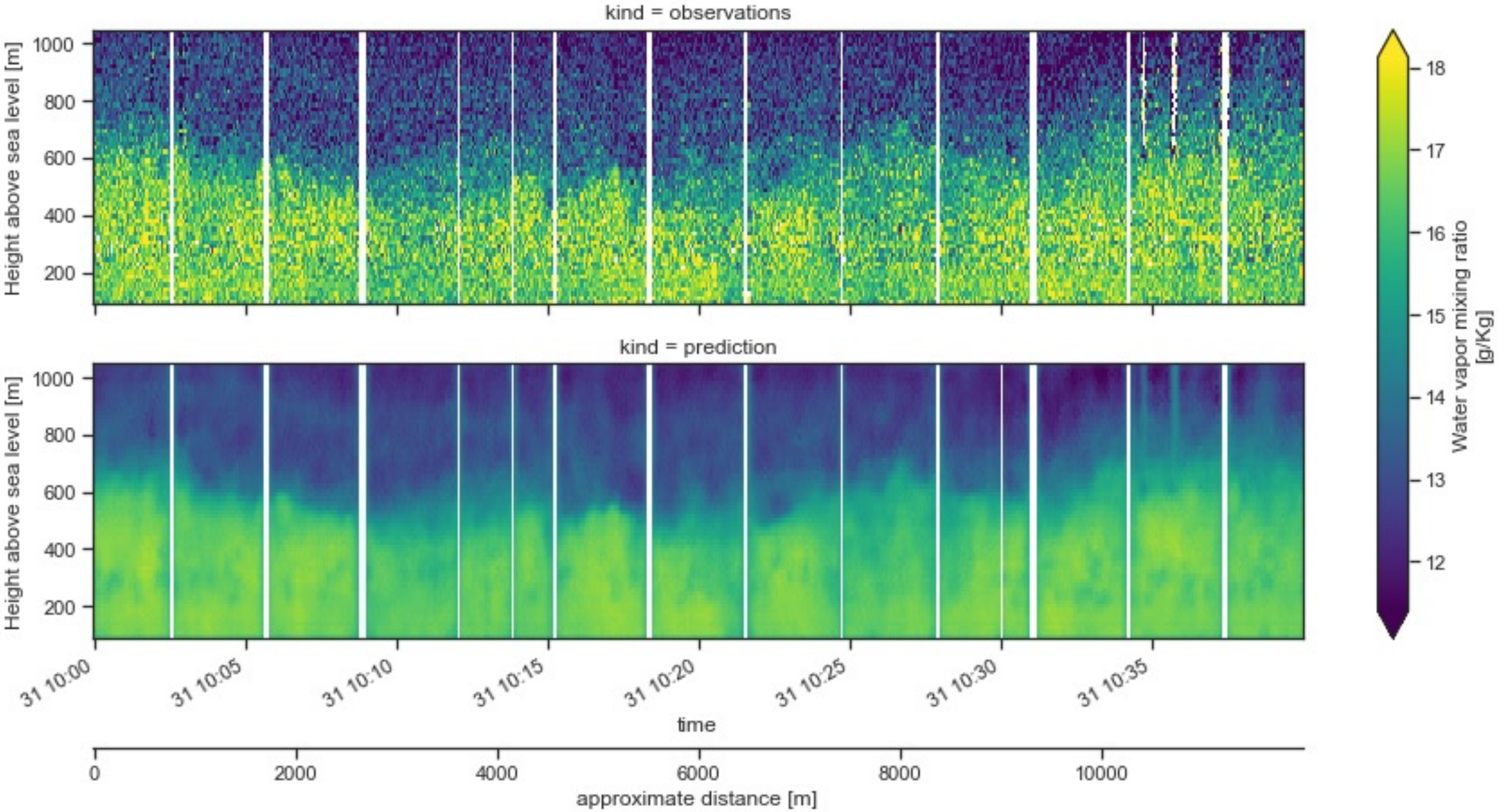
Figure 3: Blind-spot masking scheme used during NOISE2VOID training. **(a)** A noisy training image. **(b)** A magnified image patch from (a). During N2V training, a randomly selected pixel is chosen (blue rectangle) and its intensity copied over to create a blind-spot (red and striped square). This modified image is then used as input image during training. **(c)** The target patch corresponding to (b). We use the original input with unmodified values also as target. The loss is only calculated for the blind-spot pixels we masked in (b).

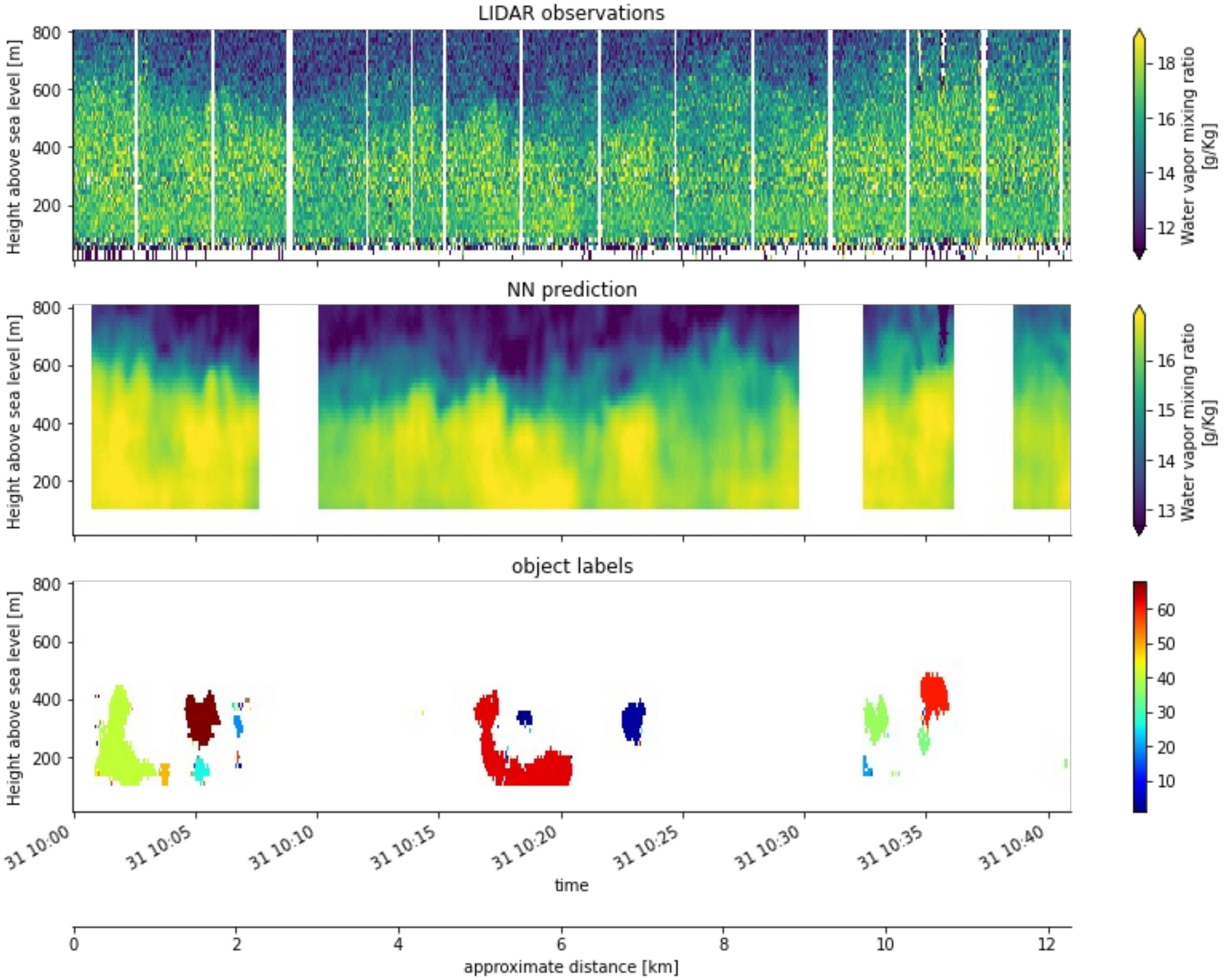
Teaser: Denoising CORAL LIDAR water vapour profiles



- Although data is noisy (if you squint) individual coherent structures are visible
- Assuming ~ 5m/s wind speed these structures are on order of hundreds of meters

Teaser: Denoising CORAL LIDAR water vapour profiles





Very much work-in-progress

The future

(is now!)

Article

Accurate medium-range global weather forecasting with 3D neural networks

https://doi.org/10.1038/s41586-023-06185-3
Received: 5 January 2023
Accepted: 9 May 2023
Published online: 05 July 2023
Open access

Kaifeng Bi[1], Lingxi Xie[1], Hengheng Zhang[1], Xin Chen[1], Xiaotao Gu[1] & Qi Tian[1]✉

Weather forecasting is important for science and so
forecast system is the numerical weather predictior
atmospheric states as discretized grids and numeri
equations that describe the transition between thos

The rise of data-driven weather forecasting

A first statistical assessment of machine learning-based weather forecasts in an operational-like context

A preprint

Zied Ben Bouallègue, Mariana C A Clare, Linus Magnusson, Estibaliz Gascón, Michael Maier-Gerber, Martin Janoušek, Mark Rodwell, Florian Pinault, Jesper S Dramsch, Simon T K Lang, Baudouin Raoult, Florence Rabier, Matthieu Chevallier, Irina Sandu, Peter Dueben, Matthew Chantry, Florian Pappenberger

ECMWF

2023

Comment

3 August 2023

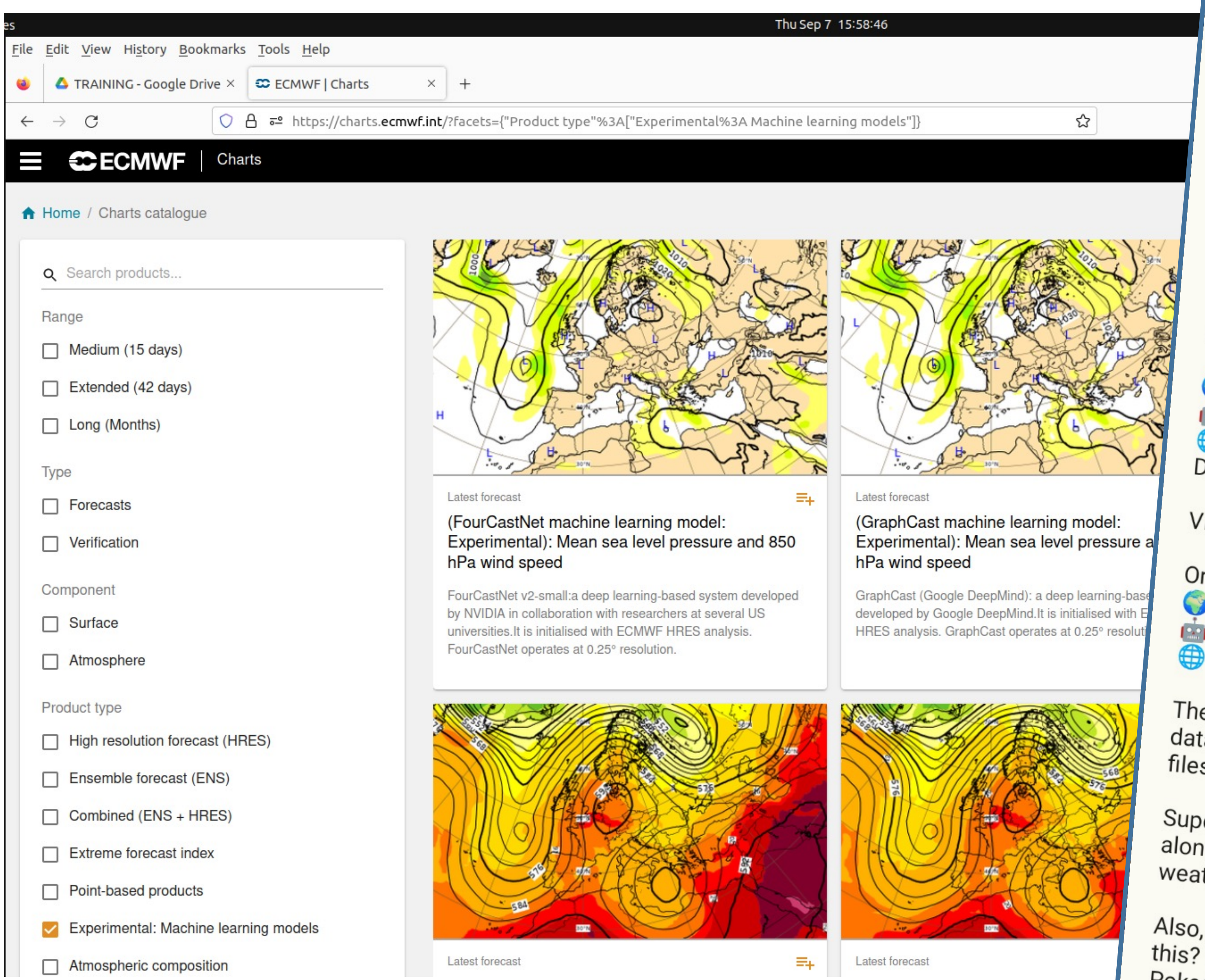
https://doi.org/10.1038/s43017-023-00468-z

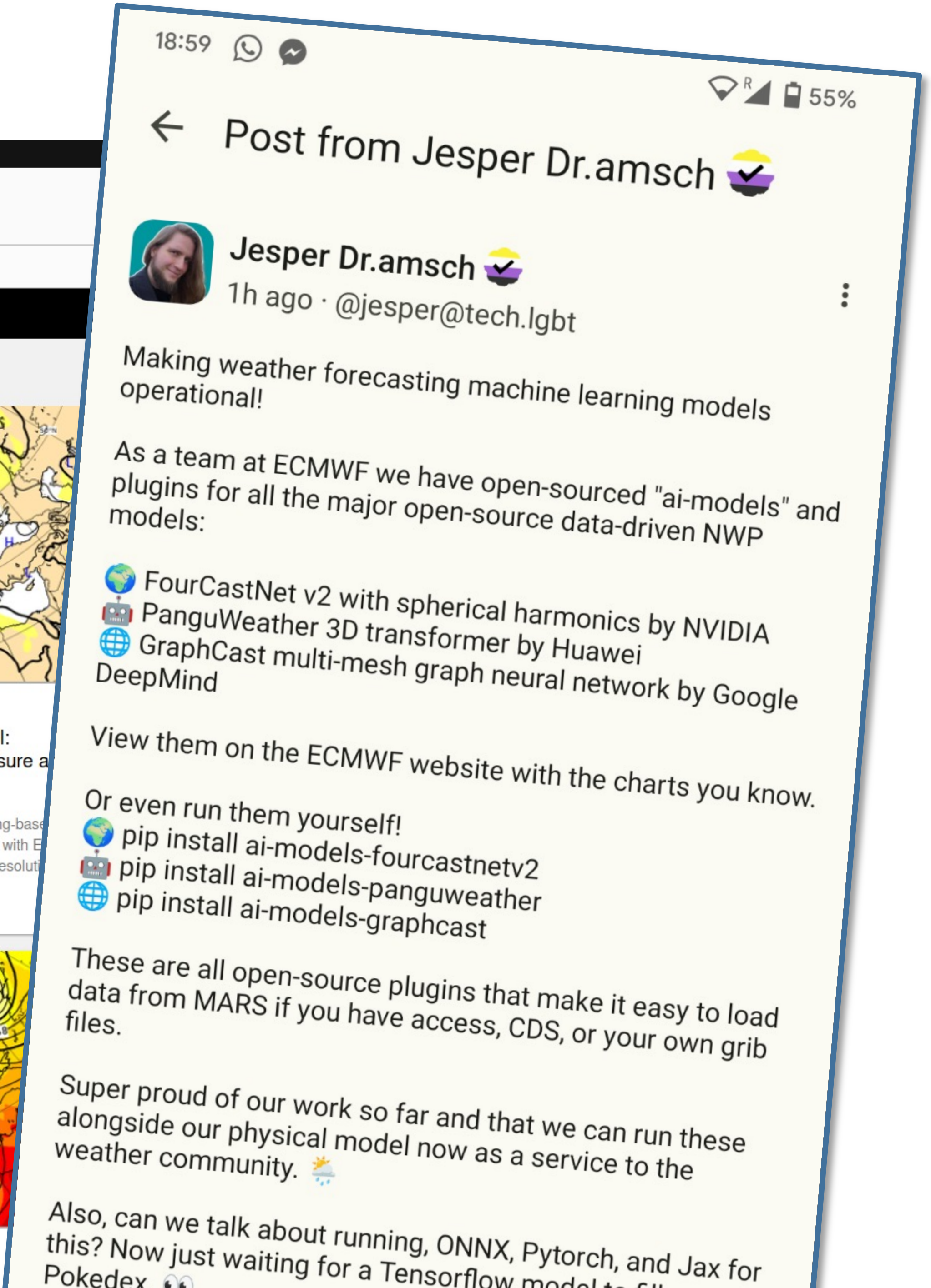
Deep learning and a changing economy in weather and climate prediction

Peter Bauer, Peter Dueben, Matthew Chantry, Francisco Doblas-Reyes, Torsten Hoefler, Amy McGovern & Bjorn Stevens

Check for updates

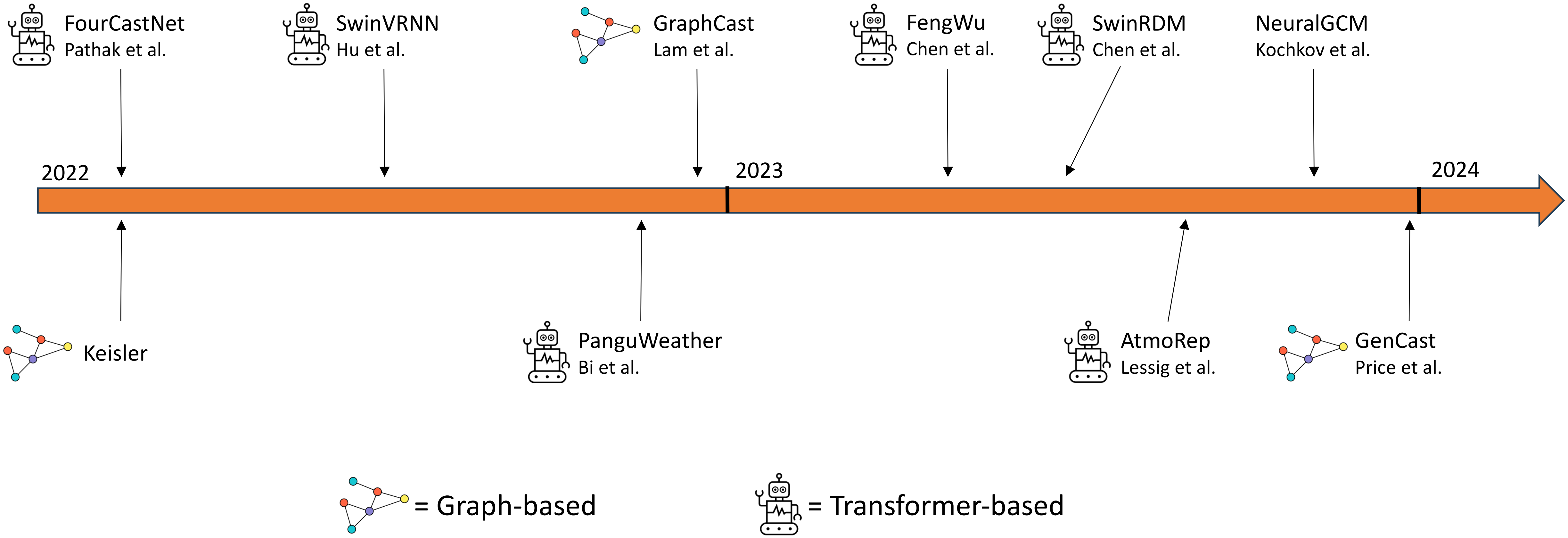
And things are moving fast...





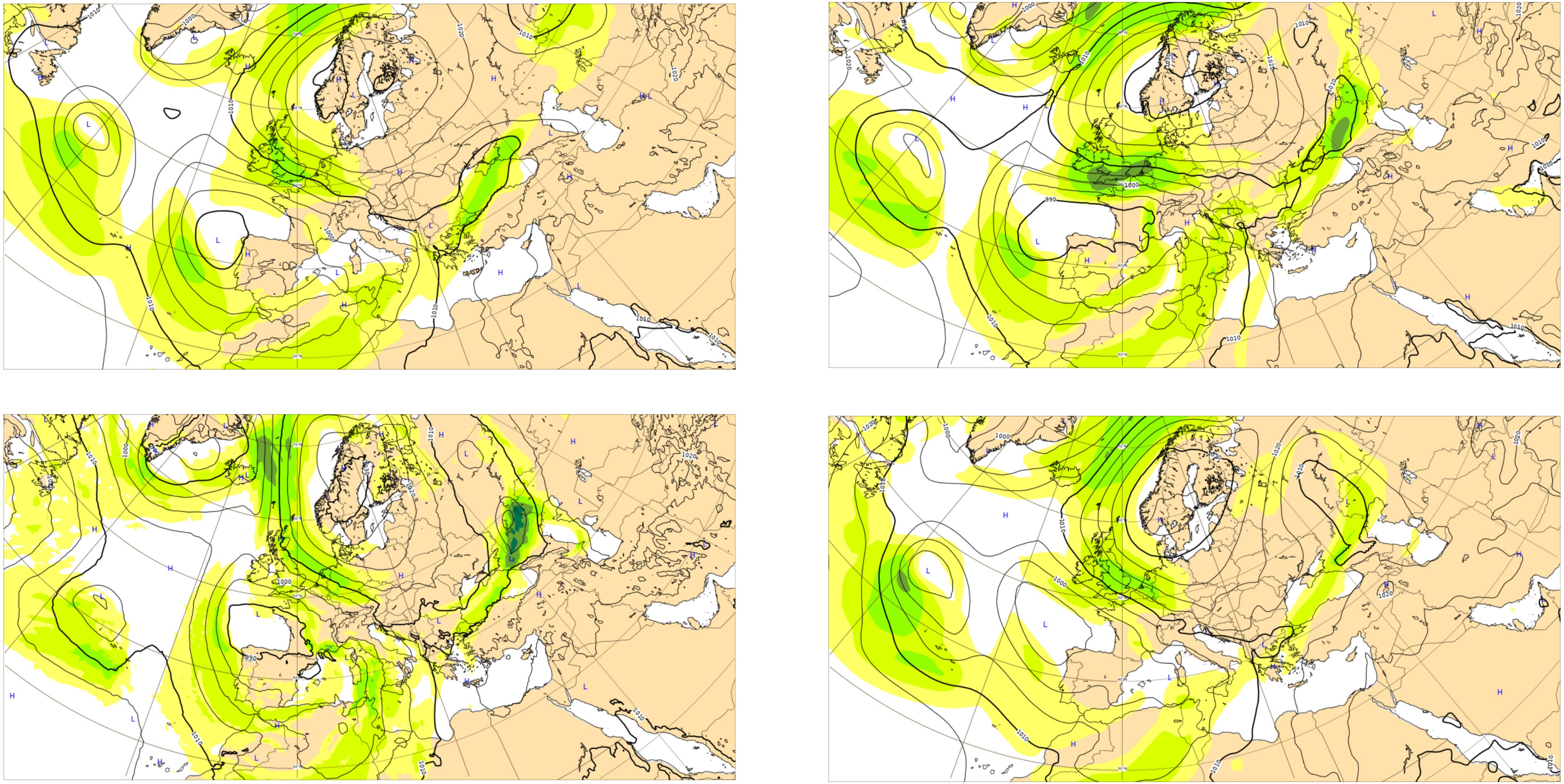
Things have been moving very fast...

A timeline of global forecasting models



Graphic from Joel Oskarsson

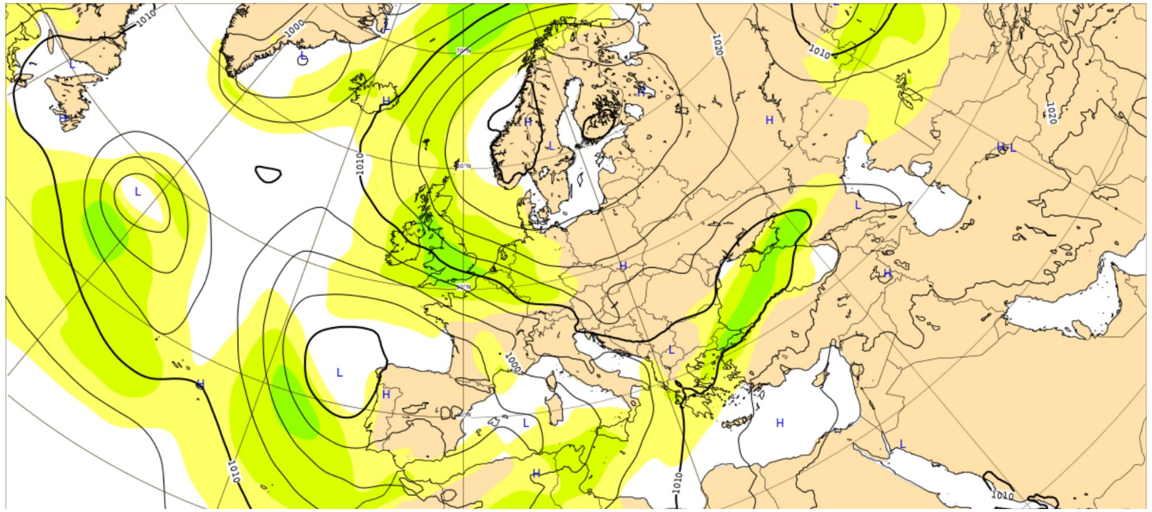
Which one is IFS (ECMWFs global model)?



https://charts.ecmwf.int/

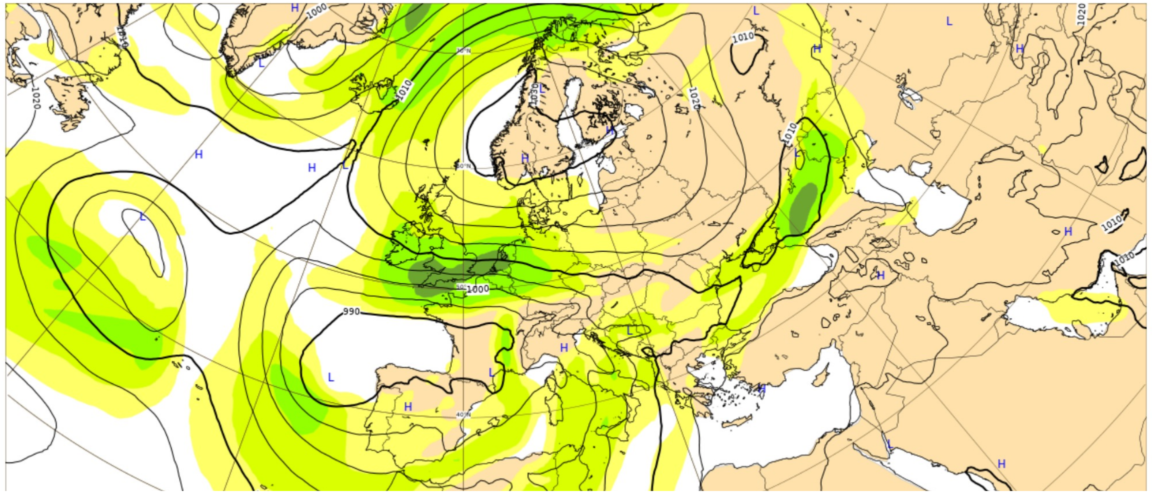
Experimental: FourCastNet ML model: Mean sea level pressure and 850 hPa wind speed

Base time: Fri 13 Oct 2023 00 UTC Valid time: Fri 20 Oct 2023 12 UTC (+180h) Area : Europe



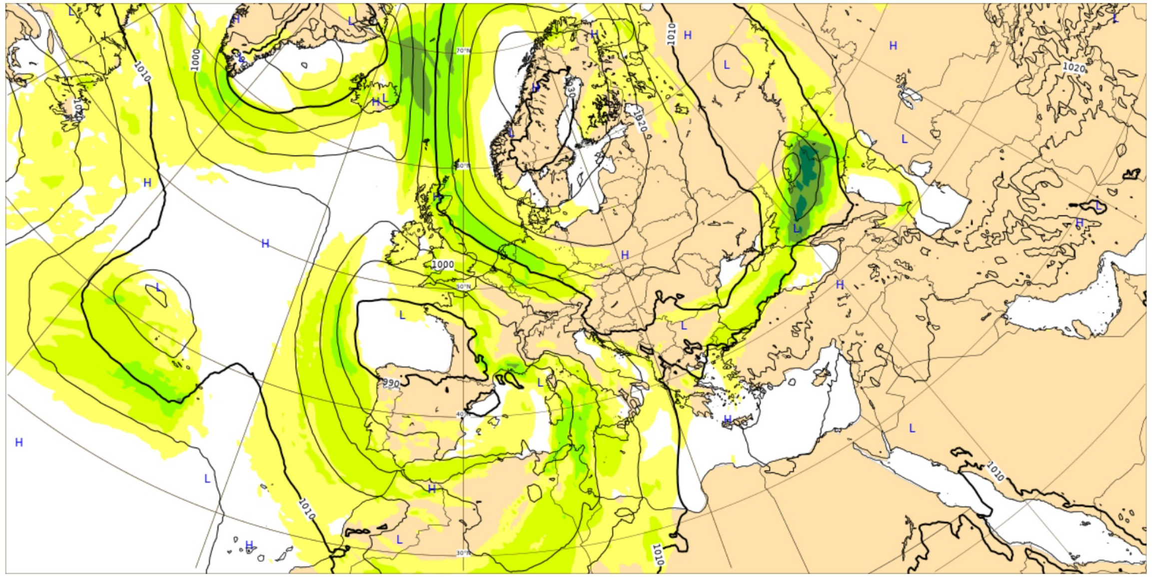
Experimental: Pangu-Weather ML model: Mean sea level pressure and 850 hPa wind speed

Base time: Fri 13 Oct 2023 00 UTC Valid time: Fri 20 Oct 2023 12 UTC (+180h) Area : Europe



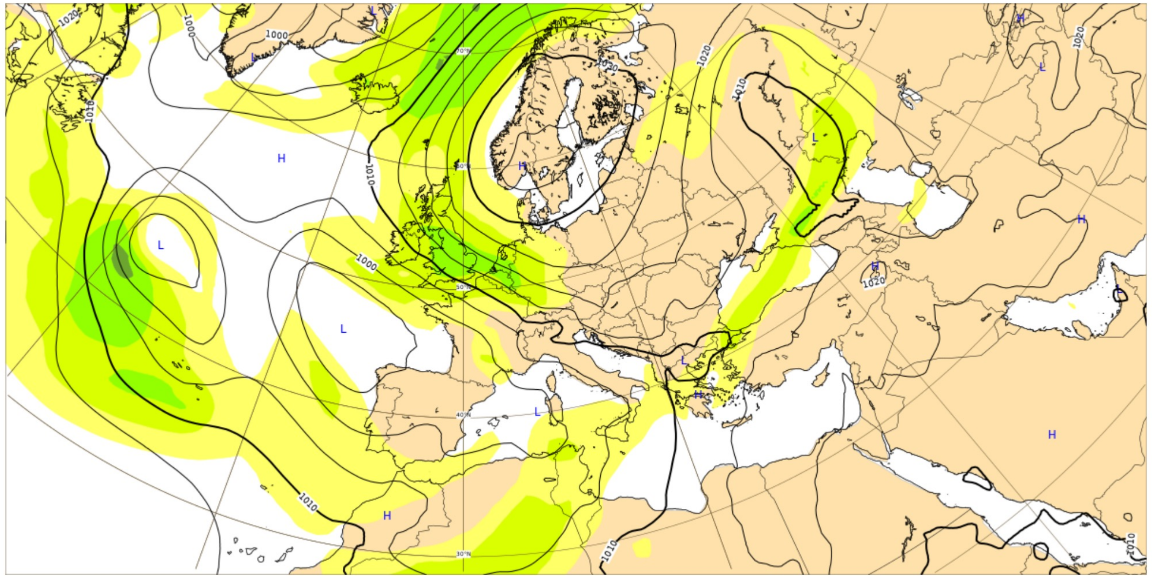
Mean sea level pressure and 850 hPa wind speed

Base time: Fri 13 Oct 2023 00 UTC Valid time: Fri 20 Oct 2023 12 UTC (+180h) Area : Europe



Experimental: AIFS (ECMWF) ML model: Mean sea level pressure and 850 hPa wind speed

Base time: Fri 13 Oct 2023 00 UTC Valid time: Fri 20 Oct 2023 12 UTC (+180h) Area : Europe



https://charts.ecmwf.int/

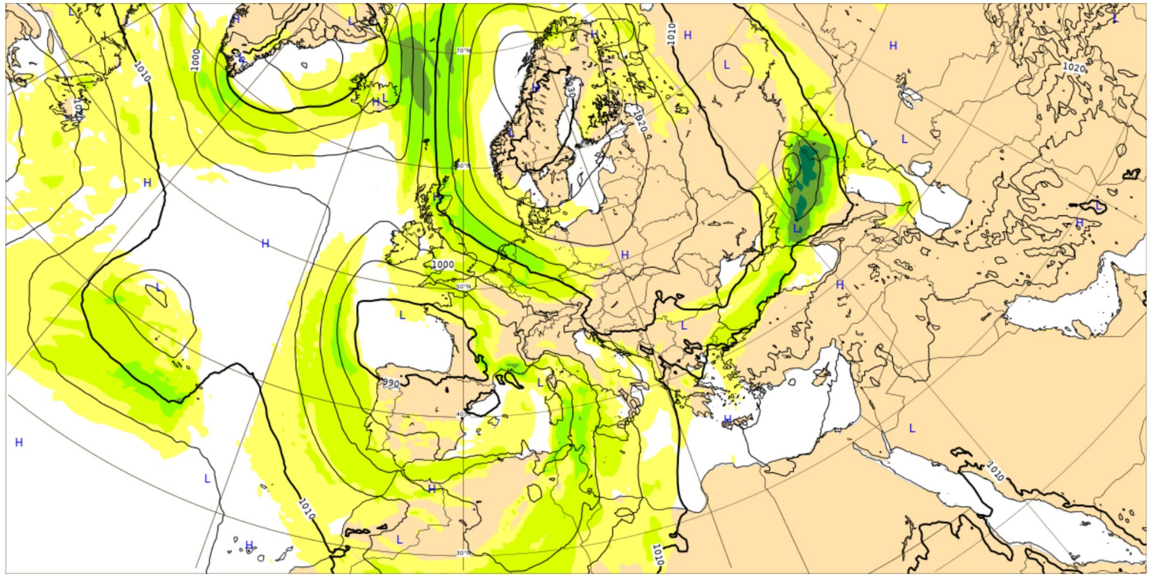
How long does it take to produce a forecast (IFS vs AIFS)?

~ 6hr on HPC

~ 25s on a GPU (A100), including write to disc

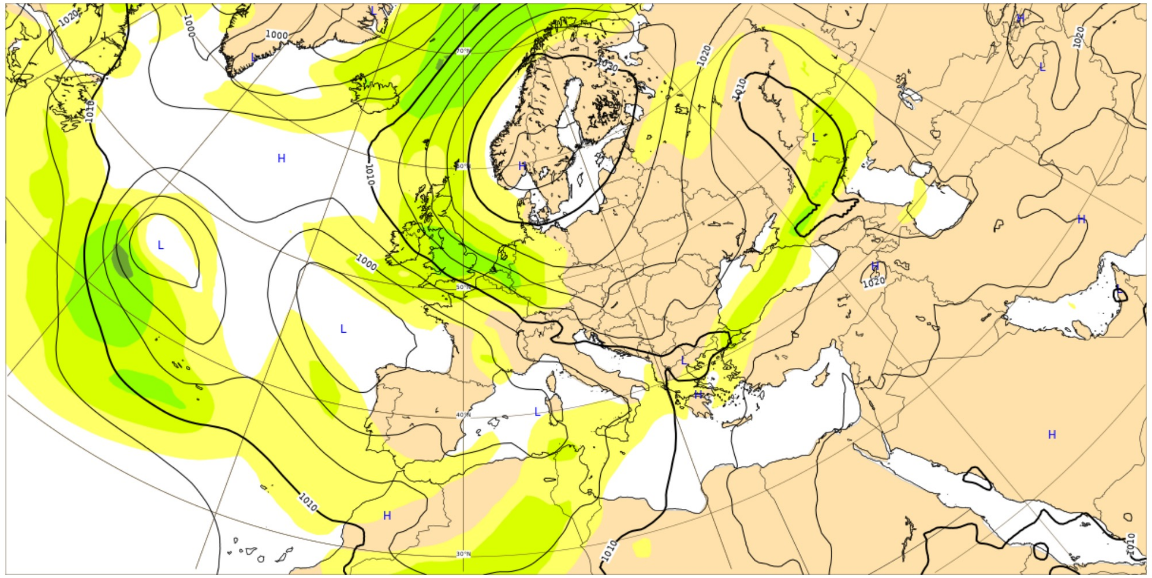
Mean sea level pressure and 850 hPa wind speed

Base time: Fri 13 Oct 2023 00 UTC Valid time: Fri 20 Oct 2023 12 UTC (+180h) Area : Europe



Experimental: AIFS (ECMWF) ML model: Mean sea level pressure and 850 hPa wind speed

Base time: Fri 13 Oct 2023 00 UTC Valid time: Fri 20 Oct 2023 12 UTC (+180h) Area : Europe



https://charts.ecmwf.int/

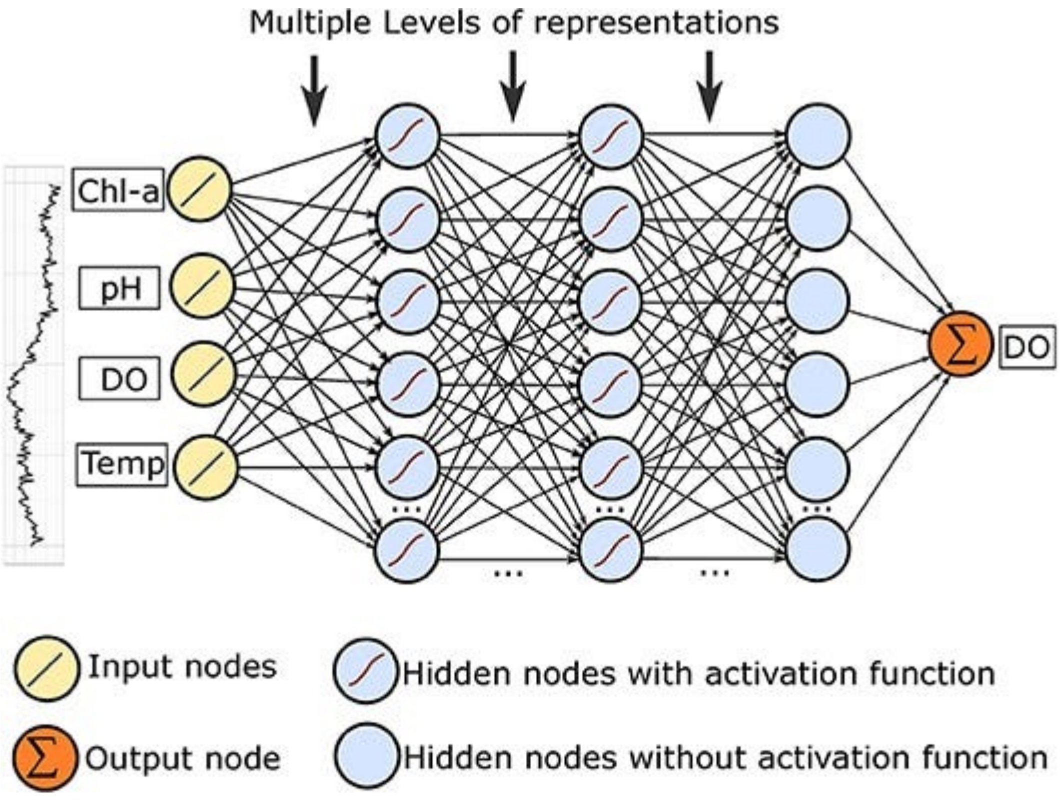
But how is this possible?

- Like polynomials, neural networks are just another class of special functions

In fact they are "Universal Function Approximators"

- They can approximate any function as number and width of layers goes to ∞

(Hornik et al 1989)



Neural networks are just:

- Linear transformations (matrix-vector products)
- Followed by non-linear scaling

$$f(\boldsymbol{x}, \theta) = \boldsymbol{W}_L \sigma \circ (\boldsymbol{W}_{L-1} \sigma \circ (\cdots \sigma \circ (\boldsymbol{W}_0 \boldsymbol{x}))), \quad \theta = (\boldsymbol{W}_0, \boldsymbol{W}_1, \cdots, \boldsymbol{W}_L)$$

But how is this possible?

- Unlike polynomials, neural networks don't suffer from curse of dimensionality:

Approximation theory:

Error $= \varepsilon$, d=dimensionality,
m=number of parameters in the model

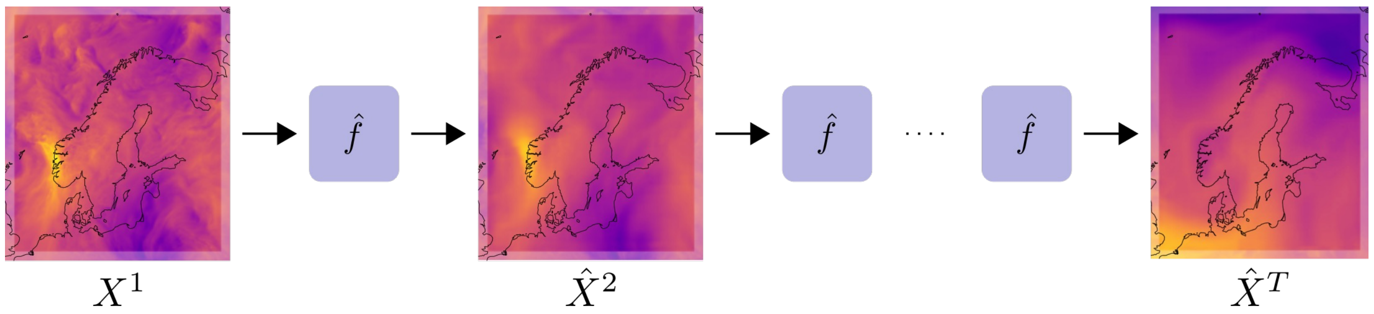
Polynomial approximation: $\varepsilon \sim m^{-1/d}$
For $\varepsilon \sim 0.1$, we need $m \sim 10^d$.

Neural network approximation: $\varepsilon \sim m^{-1/2}$
For $\varepsilon \sim 0.1$, we need $m \sim 10^2$.

See Weinan E's talk for the mathematical detail

How do these models work?

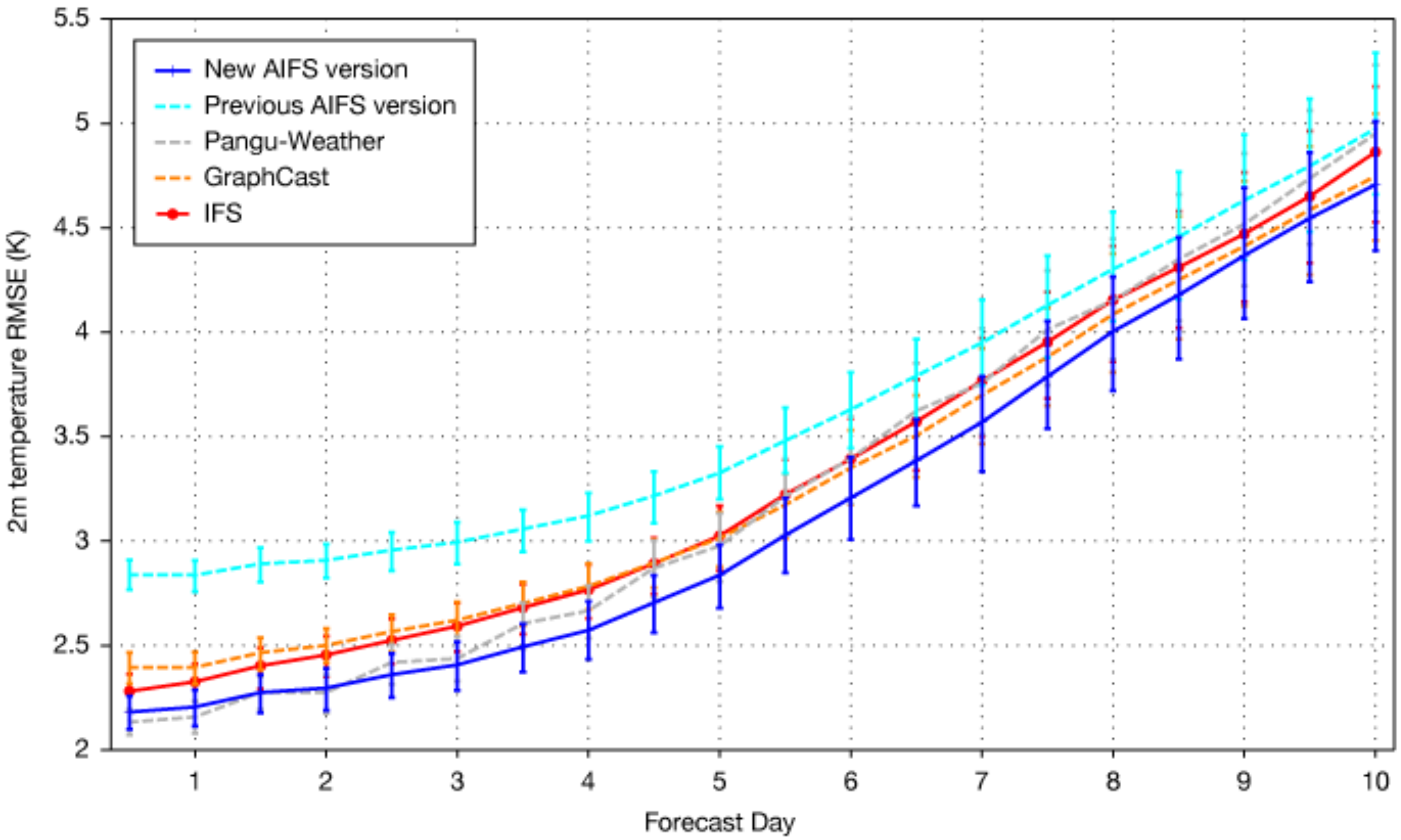
- Weather state X^t
- Dynamics model $X^t = f\left(X^{t-1}, \ldots, X^{t-p}\right)$
- Approximate with machine learning model $\hat{f} \approx f$



- Train on dataset of trajectories $X^1, X^2, \ldots, X^T.$
  - Forecast data: Fast surrogate model
  - Reanalysis data: Surpass existing NWP

Slide from Joel Oskarsson

2m temperature mean-squared error against synoptic observations



northern hemisphere, September–October–November period of 2023 [2]

"In our view, we are currently placed at an exciting moment in weather forecasting history." - ECMWF[1]

lower is better

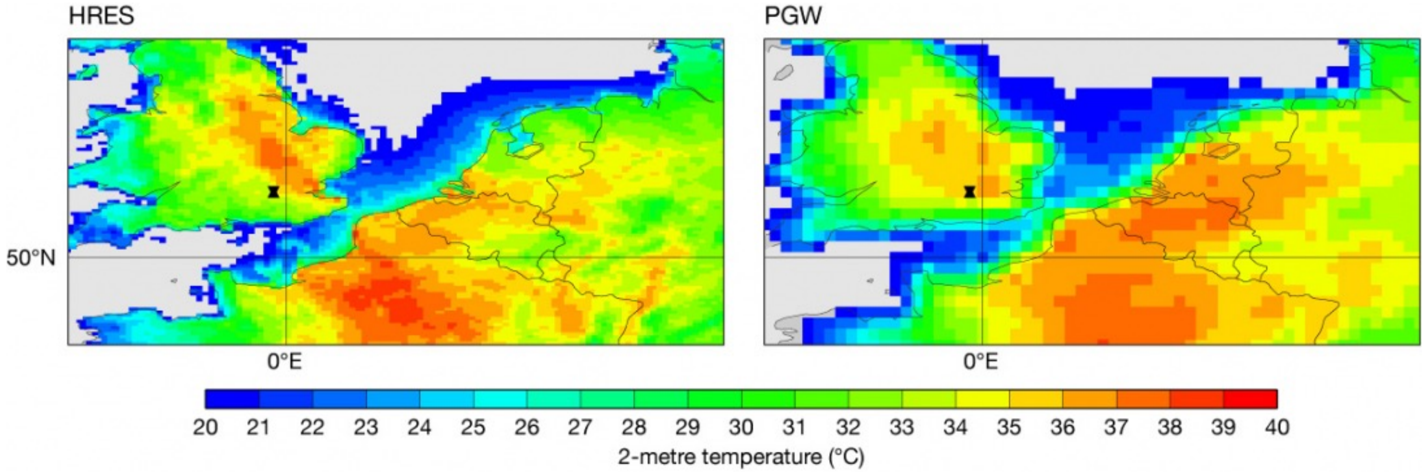
“Previous AIFS”:
- GNN with message-passing on graph, 1deg

“New AIFS”:
- Attention-based GNN, 0.25 deg

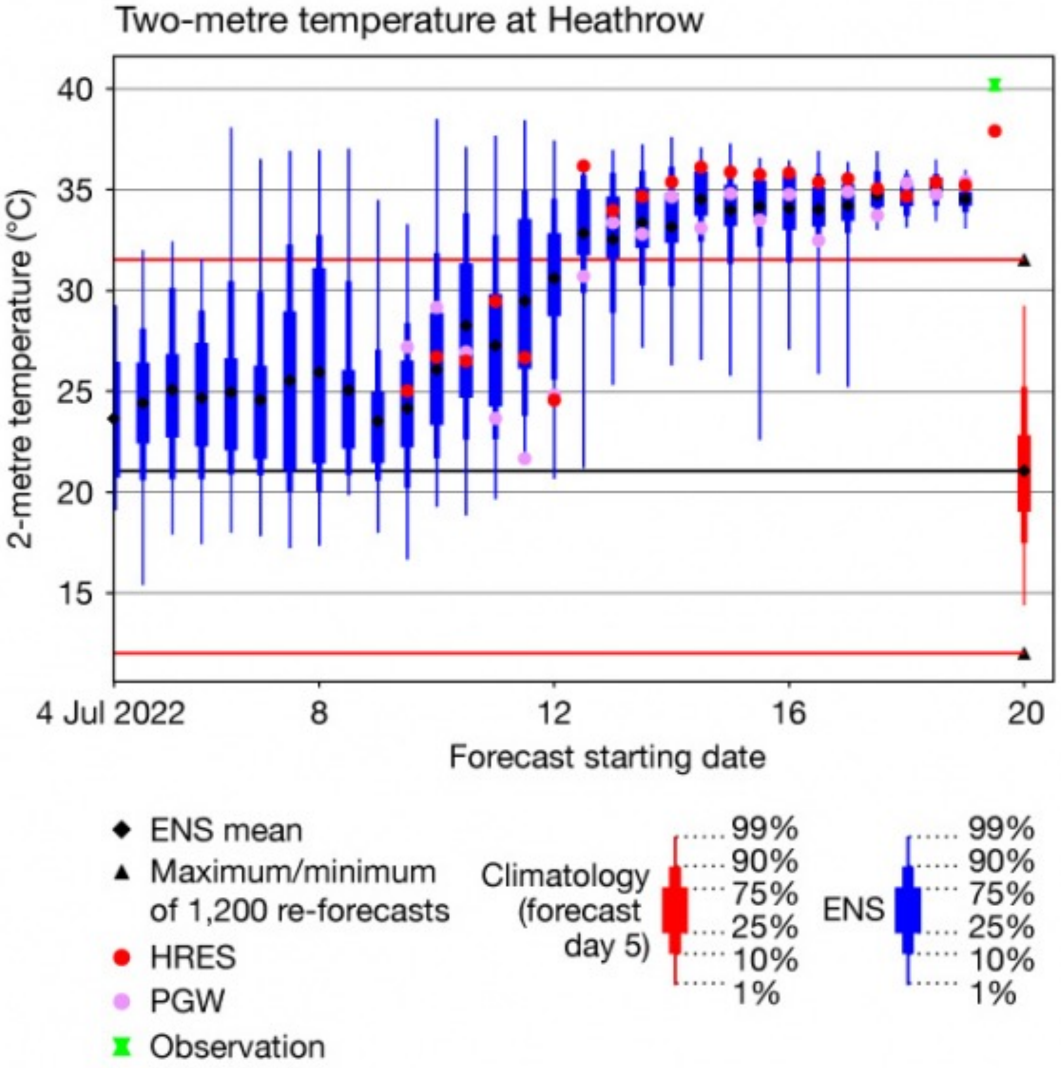
[1] *20/6/2023: https://www.ecmwf.int/en/about/media-centre/science-blog/2023/rise-machine-learning-weather-forecasting*

[2] *16/1/2024: https://www.ecmwf.int/en/about/media-centre/aifs-blog/2024/first-update-aifs*

Heatwave forecast July 2022



- Pangu-Weather (PGW) predicts heatwave tempreature with similar skill to high-res forecast (HRES) and within ensemble spread
- Pangu-Weather lacks some of fine-scale structure in HRES



https://www.ecmwf.int/en/newsletter/176/news/exploring-machine-learning-forecasts-extreme-weather

ML & IFS: tropical cyclones

The cyclone tracks are looking very good, but the central pressure is under-predicted

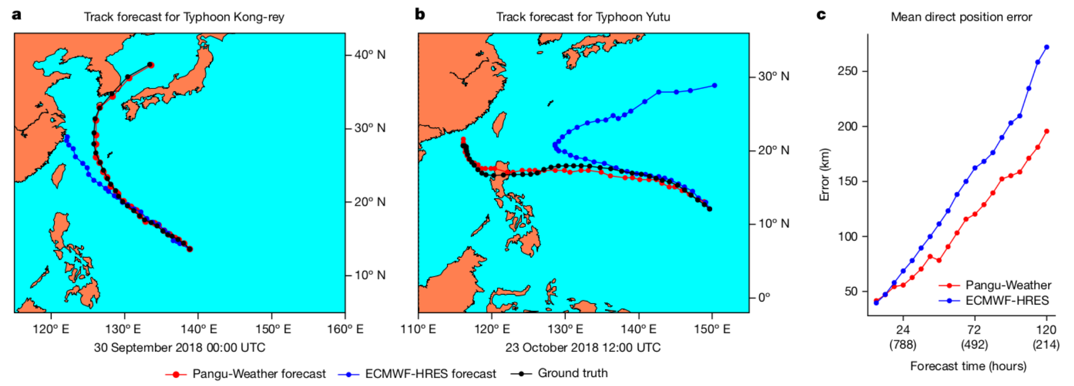


Fig. 4 | Pangu-Weather is more accurate at early-stage cyclone tracking than ECMWF-HRES. a,**b**, Tracking results for two strong tropical cyclones in 2018, that is, Typhoon Kong-rey (2018–25) and Yutu (2018–26). The initial time point is shown below each panel. The time gap between neighbouring dots is 6 h. Pangu-Weather forecasts the correct path of Yutu (that is, it goes to the Philippines) at 12:00 UTC on 23 October 2018, whereas ECMWF-HRES obtains the same conclusion 2 days later, before which it predicts that Yutu will make a big turn to the northeast. **c**, A comparison between Pangu-Weather and ECMWF-HRES in terms of mean direct position error over 88 cyclones in 2018. Each number in brackets in the *x*-axis indicates the number of samples used to calculate the average. For example, '(788)' means that there are in total 788 initial points from which the typhoon lasts for at least 24 hours, and the 788 direct position errors of Pangu-Weather and ECMWF-HRES were averaged into the final results. Panels **a** and **b** were plotted using the Matplotlib Basemap toolkit.

Bi et al. 2023

The rise of data-driven weather forecasting A PREPRINT V2

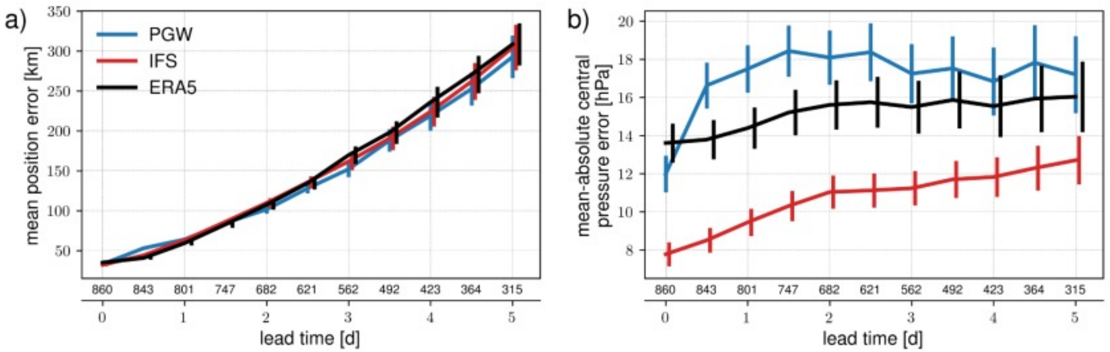


Figure 8: Tropical cyclone verification results: (a) mean position error and (b) mean absolute central pressure error as a function of the lead time for 2018. Forecasts are verified against the IBTrACS dataset and homogenized to have a consistent number of cases between models. For each lead time, the number of cases is displayed directly below the graphs. The vertical bars indicate the 2.5%-97.5% confidence intervals.

Ben-Bouallegue et al., (2023), https://doi.org/10.48550/arXiv.2307.10128

Pangu-Weather vs ECMWF HRES – forecast bust

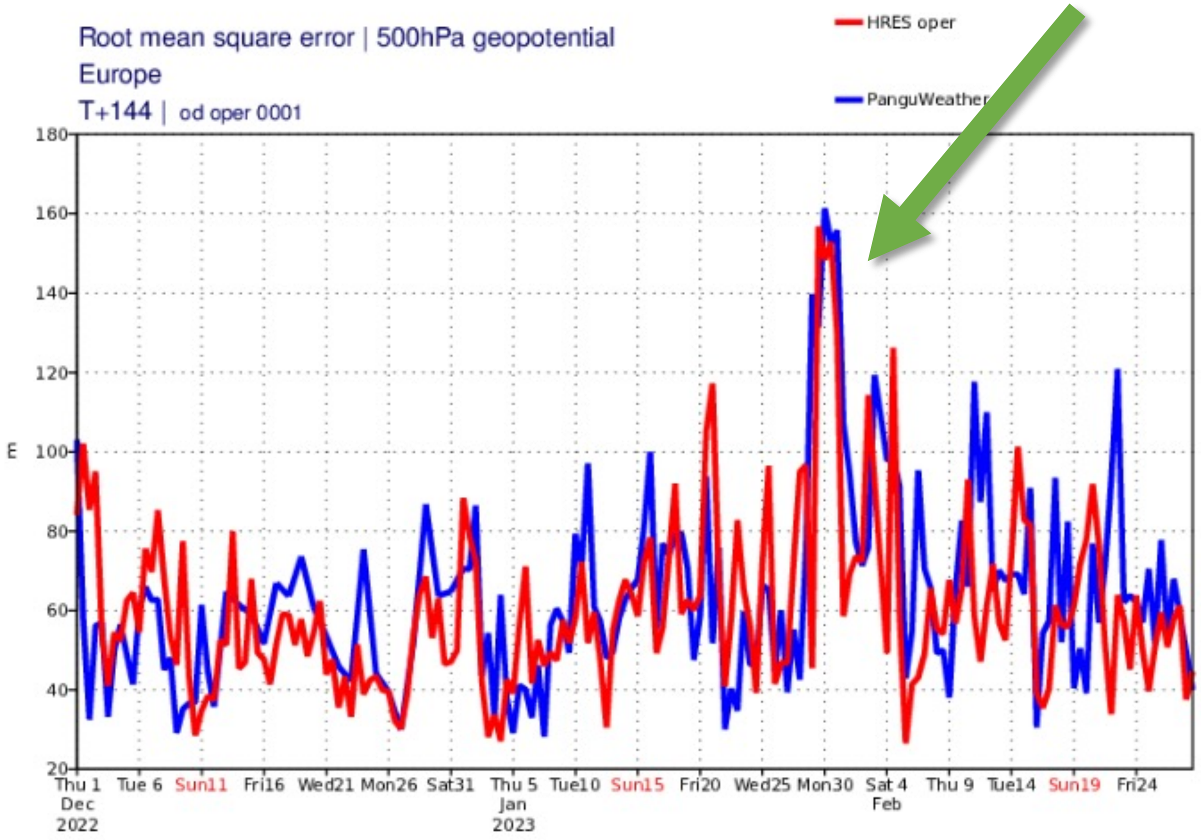


Figure 2: Root-mean-square error for HRES (red) and Pangu-Weather (blue) of 500hPa geopotential 6-day forecasts over Europe for the winter (December-January-February) 2022/2023. Reference is the HRES operational analysis.

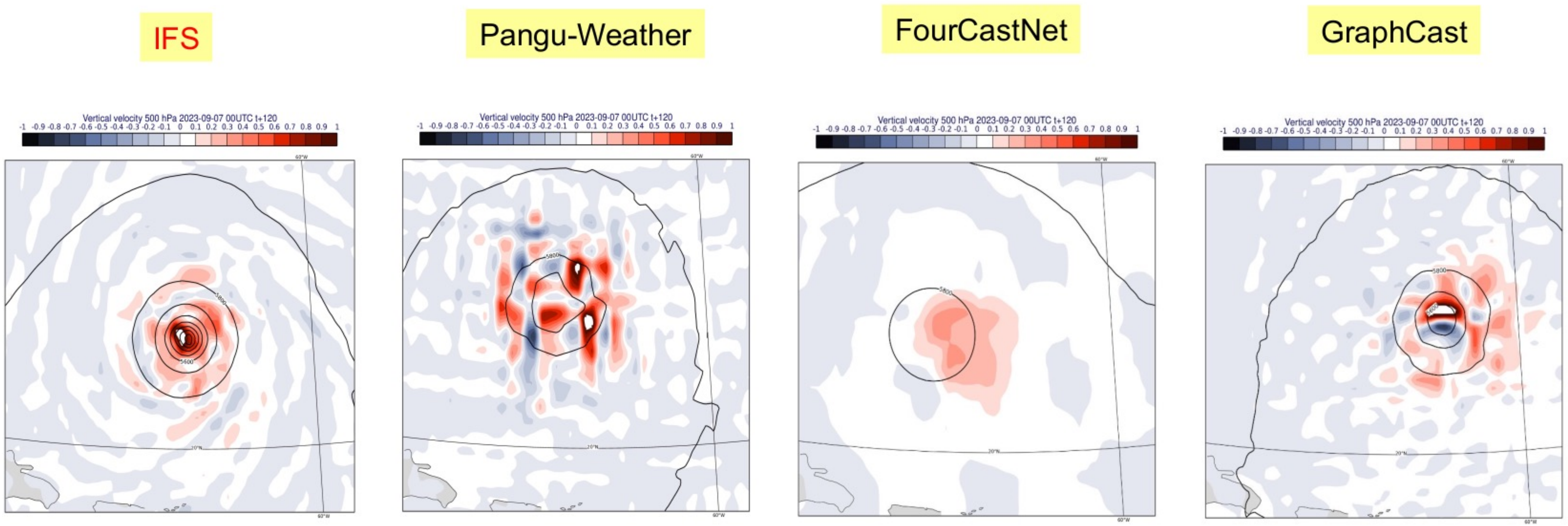
- timing of forecasts busts similar in ML and IFS model

ML & IFS: tropical cyclones

But the local-scale dynamics are not right...

ML models dynamical fields (3)

Z500 + vert. vel. (m/s, shaded)
2023-09-07 00UTC t+120h



Slide from Massimo Bonavita, see https://arxiv.org/abs/2309.08473 for details

Global ML NWP: Take-home messages and caveats

- ML models **competitive with IFS** in forecast of **upper-air vars against operational analysis** and **surface vars against obs**
- **Good ML performance** in prediction of some aspects of **extreme** events (TCs tracks for example), but lacking finer scale physical structure (cloud processes?)
- Once trained, **ML model runs 10^4 times faster** than IFS
- ~~ML trained on ERA5 (0.25deg) -> **lack of small-scales** in forecasts~~
- ~~No ensemble forecasts, no uncertainty estimates~~
- ~~**Rain not included** in predictions (reanalysis deemed poor reference)~~

As of Dec 2023:
GenCast produces forecast ensembles using Diffusion Models in GNN

As of 4/3/2024:
AIFS includes precipitation forecast

Dec 2023: Ensemble data-driven model (GenCast, Google)

“Producing a single 15-day trajectory with GenCast takes around a minute on a Cloud TPU v4, and so N ensemble members can also be generated in around a minute with N TPUs, enabling the use of orders of magnitude larger ensembles in the future”

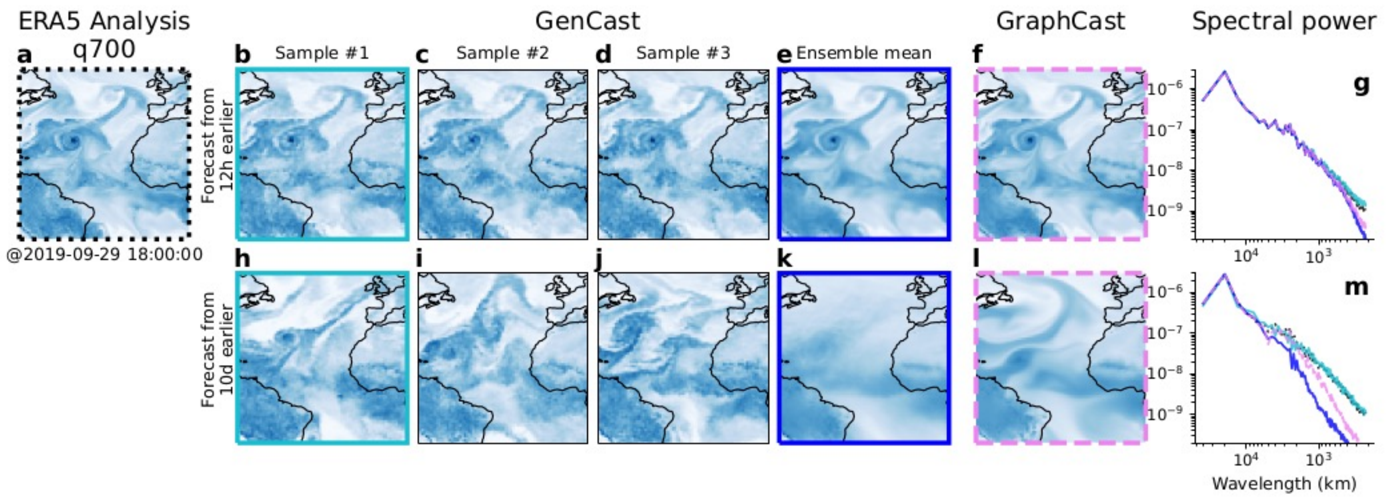
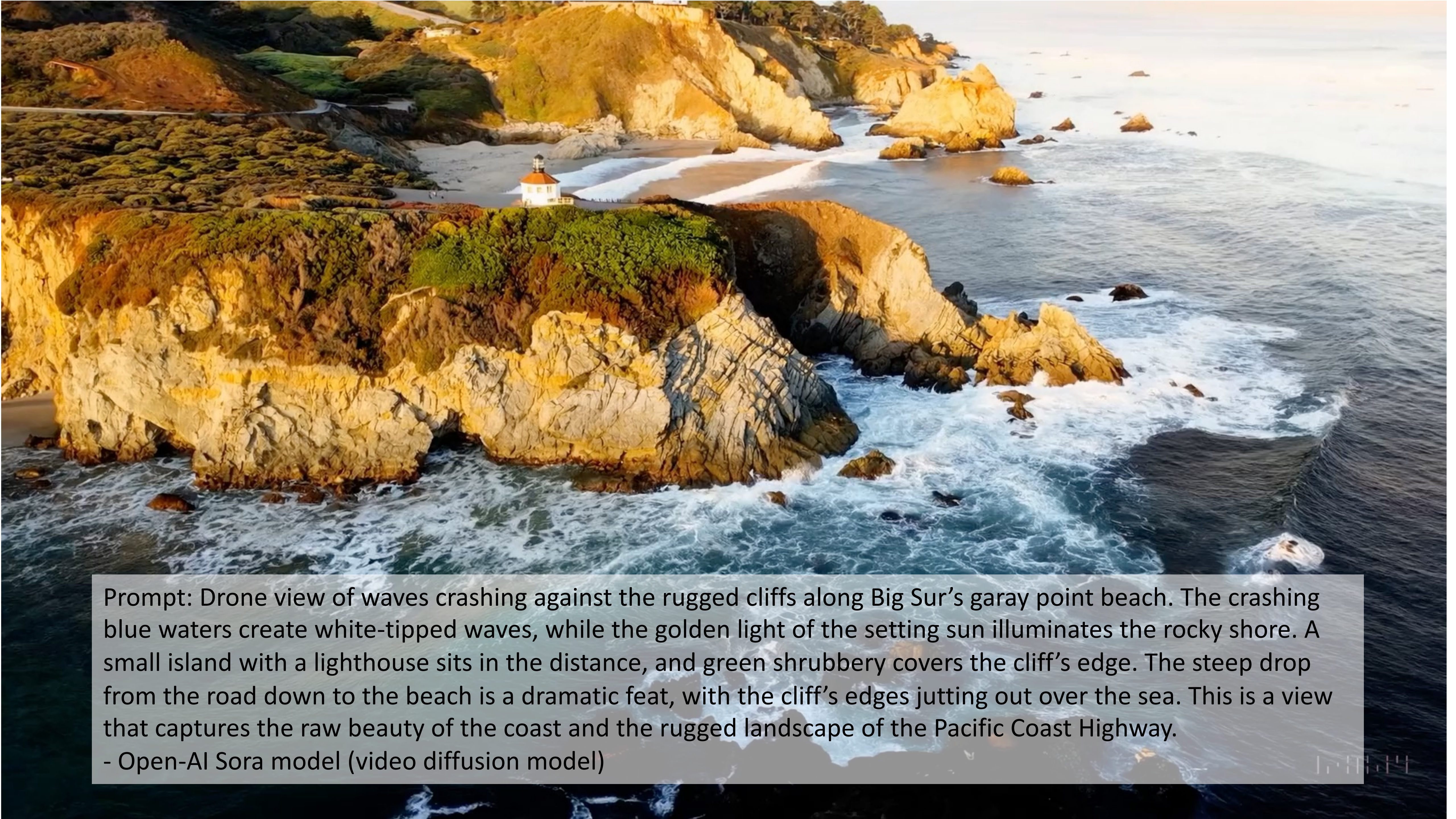


Figure 3 | Visualization of representative states produced by GenCast compared to GraphCast. (a) ERA5 analysis state for specific humidity at 700hPa at 6pm on the 29th of September of 2019. (b-d) 3 sample forecasts of this state by GenCast from 12 hours earlier. (e) Ensemble average obtained by taking the mean of 50 sample forecasts by GenCast from 12 hours earlier. (f) Forecast by the GraphCast (model which is deterministic), made 12 hours earlier. (g) Spectrum of the fields shown in panels (a-f), with ~~colors matching the frames of the panels~~. (h-m) Same as (b-g), but for forecasts made 10 days earlier. Unlike deterministic GraphCast, which expresses uncertainty as blurring which increases with lead time (f, l), we observe how the sample forecasts produced by GenCast are sharp (g, m), regardless of whether the forcasts are for 12 hours ahead (g, b-d) (where the three samples are very similar) or 10 days ahead (m, h-j) (where the three samples differ more). The samples can still be averaged to produced a blurry mean state (e, k). Additional visualizations and an explanation of how this date/time was selected for visualisation are available in Appendix A.8.



Prompt: Drone view of waves crashing against the rugged cliffs along Big Sur's garay point beach. The crashing blue waters create white-tipped waves, while the golden light of the setting sun illuminates the rocky shore. A small island with a lighthouse sits in the distance, and green shrubbery covers the cliff's edge. The steep drop from the road down to the beach is a dramatic feat, with the cliff's edges jutting out over the sea. This is a view that captures the raw beauty of the coast and the rugged landscape of the Pacific Coast Highway.
- Open-AI Sora model (video diffusion model)

How do this GNN-based forecasting models work?

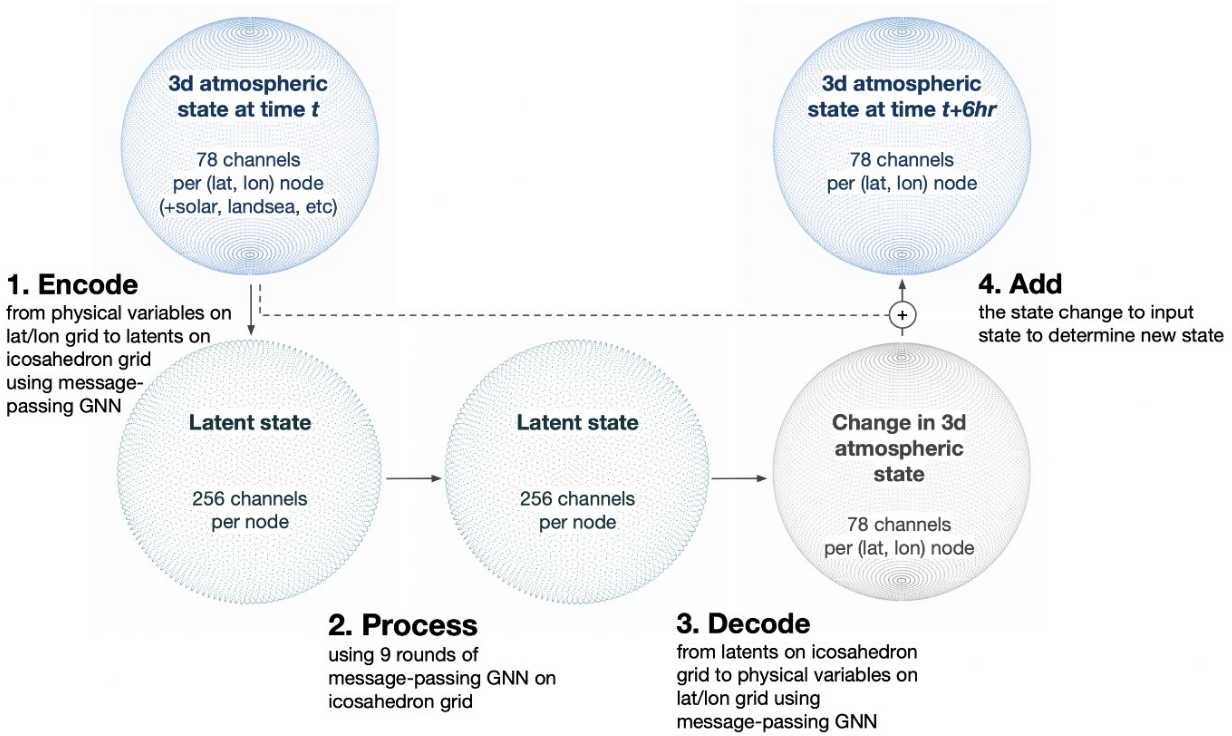


Figure 1: Using the current atmospheric state, the model evolves the state forward by 6 hours. The 3D atmospheric state is defined on a uniform latitude/longitude grid, with 78 channels per pixel (6 physical variables $\times$ 13 pressure levels = 78 channels). An Encoder GNN encodes onto latent features defined on a icosahedron grid, a Processor GNN performs additional processing of the latents, and a Decoder GNN maps back to the atmospheric state on a latitude/longitude grid.

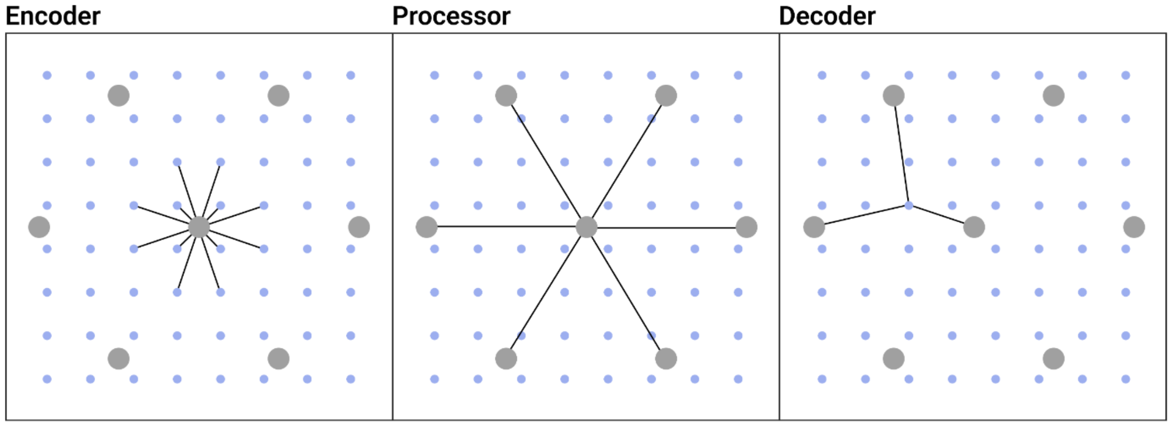
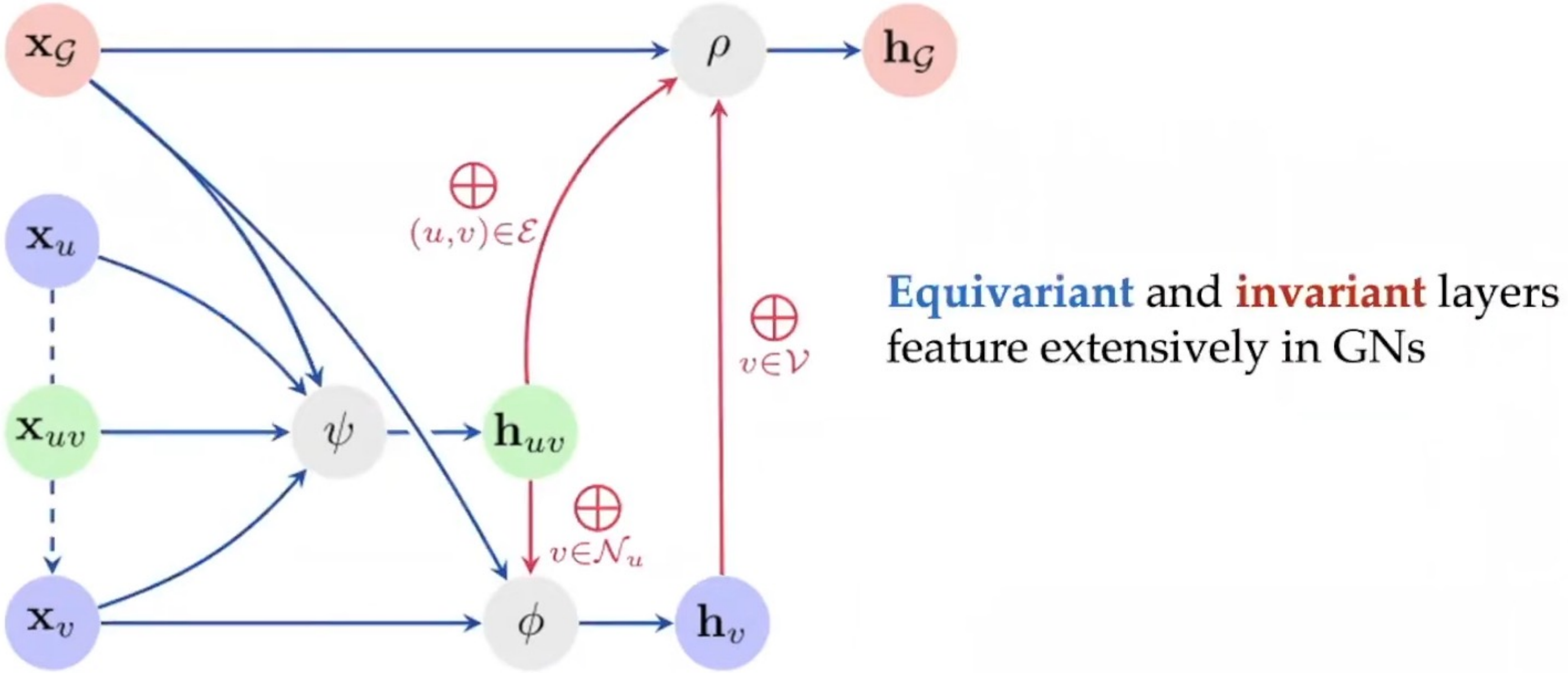


Figure 2: A schematic view of the local graph connectivity in the Encoder, Processor, and Decoder. Left: local spatial and channel information is encoded into an icosahedron node using data from nearby nodes on the input latitude/longitude grid. Center: data on the icosahedron node is further processed using data from nearby icosahedron nodes (including itself, which is not explicitly shown). Right: the output latitude/longitude data is created by decoding data from nearby icosahedron nodes.

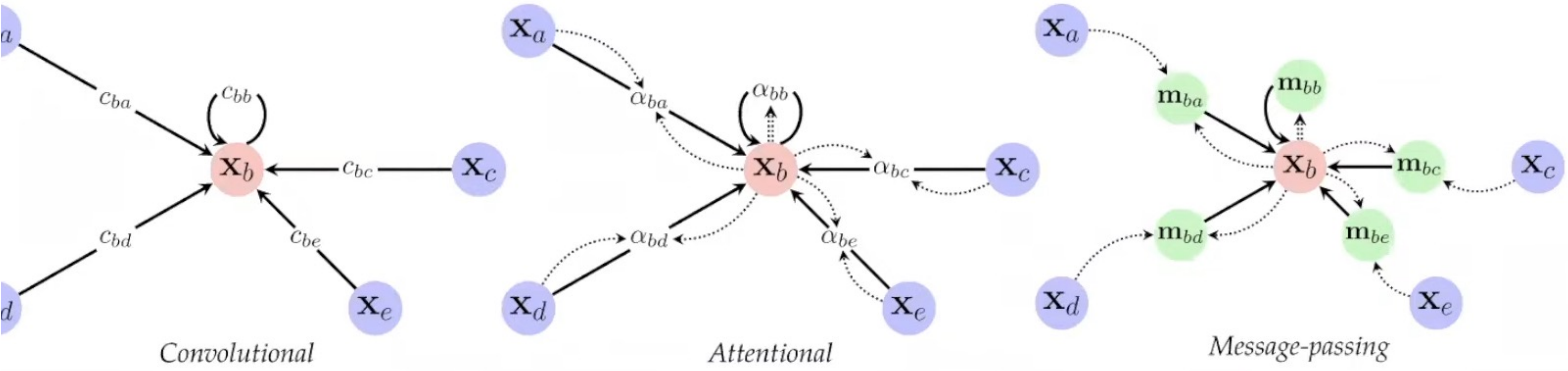
Ryan Keisler, 2022

Ok, but what are GNNs (Graph Neural Networks)?



Petar Veličković Graph Neural Networks: Geometric, Structural and Algorithmic Perspectives Part 2
Cambridge ELLIS Machine Learning Summer School 2022

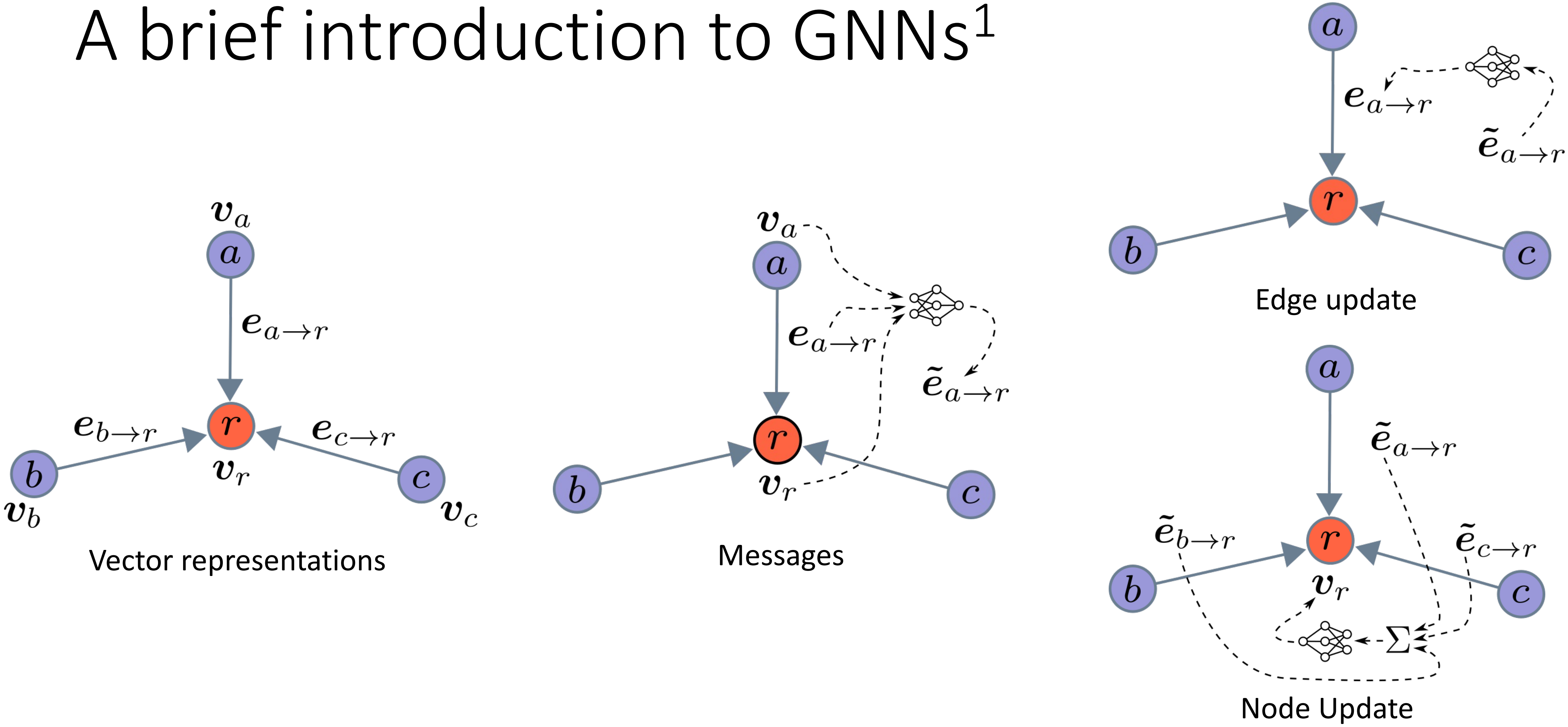
The three "flavours" of GNN layers



$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right) \qquad \mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} a(\mathbf{x}_i, \mathbf{x}_j)\psi(\mathbf{x}_j)\right) \qquad \mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} \psi(\mathbf{x}_i, \mathbf{x}_j)\right)$$

Petar Veličković Graph Neural Networks: Geometric, Structural and Algorithmic Perspectives Part 2
Cambridge ELLIS Machine Learning Summer School 2022

A brief introduction to GNNs[1]



[1] J. Gilmer, et al. (2017). Neural Message Passing for Quantum Chemistry. *ICML*.
P. Battaglia, et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint*.

Slide from Joel Oskarsson

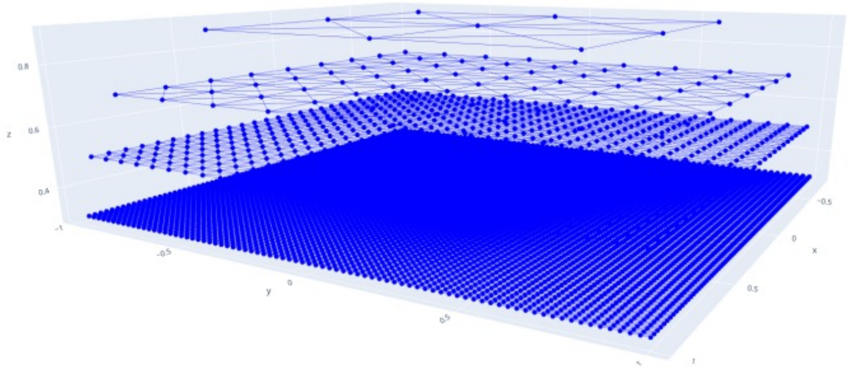
But can we do km-scale forecasting?

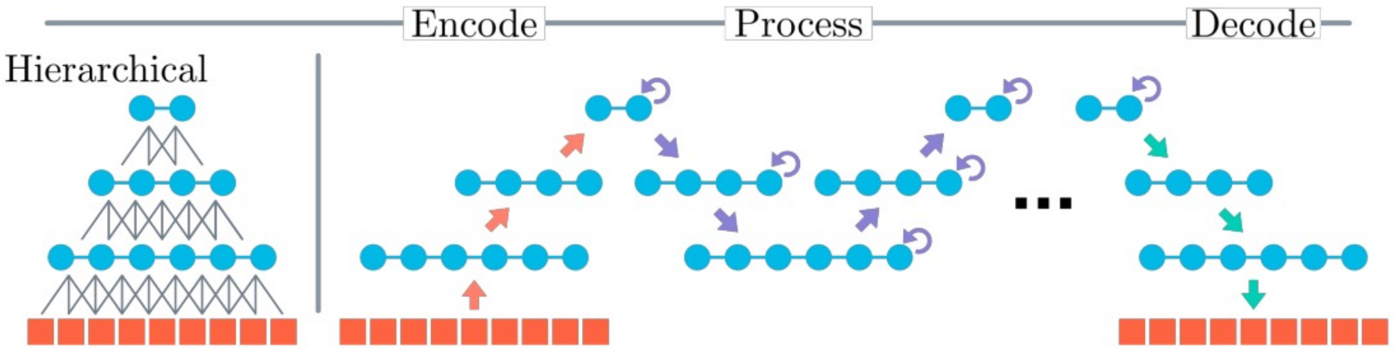
Yes!

Neural-LAM

Hi-LAM: Hierarchical multi-scale graph

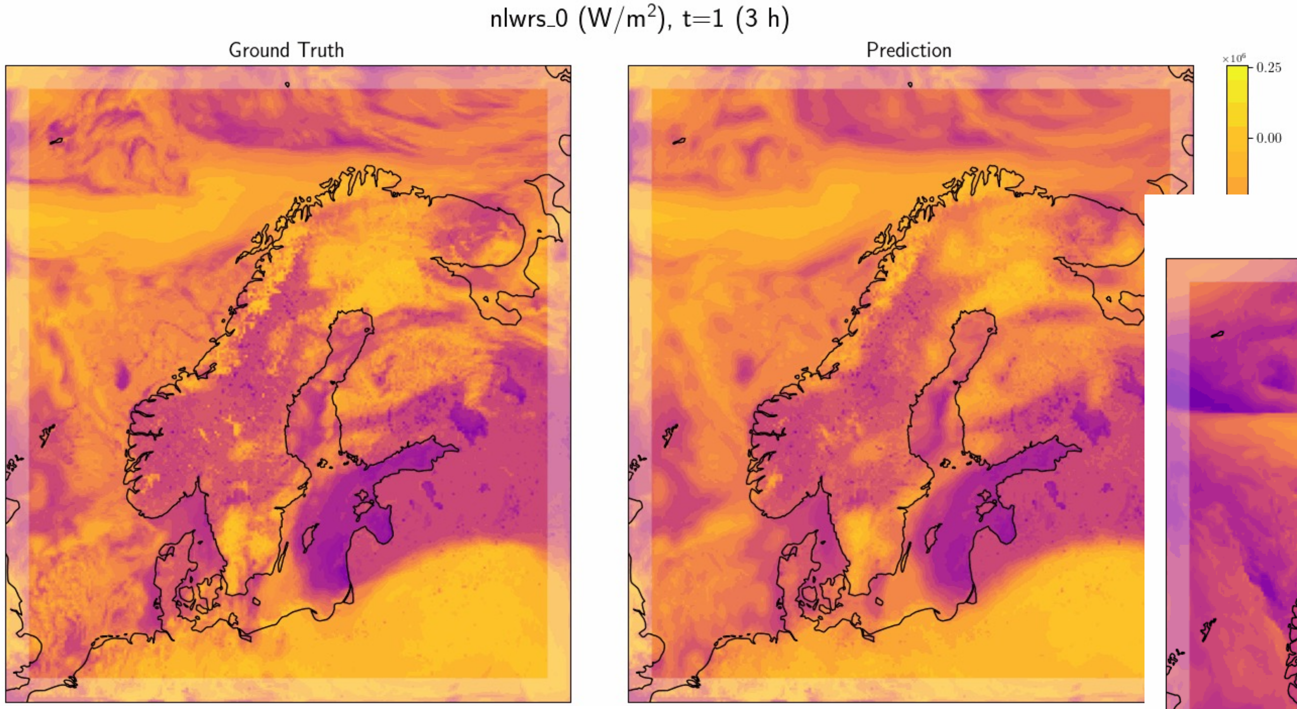
- 4 levels of nodes in mesh graph
  - Intra-level edges
  - Inter-level edges between adjacent levels
- Sequential GNN message passing up and down the hierarchy

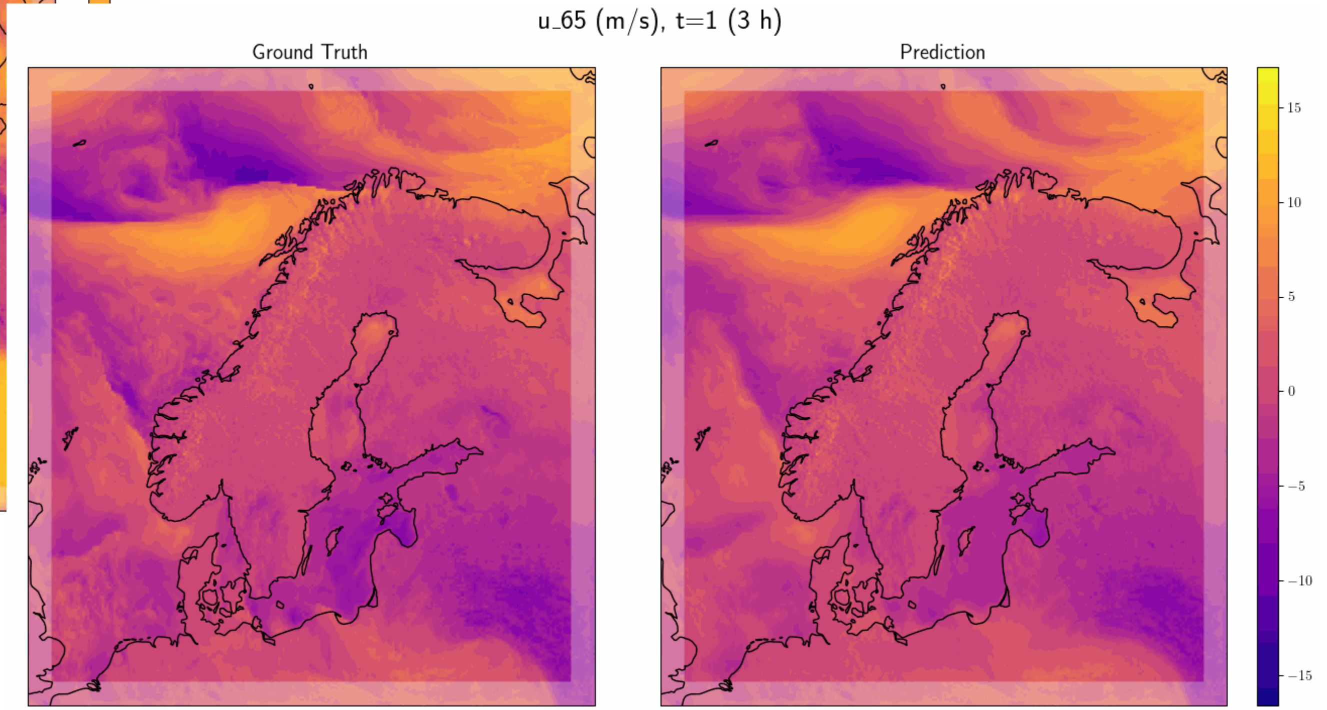




Slide from Joel Oskarsson

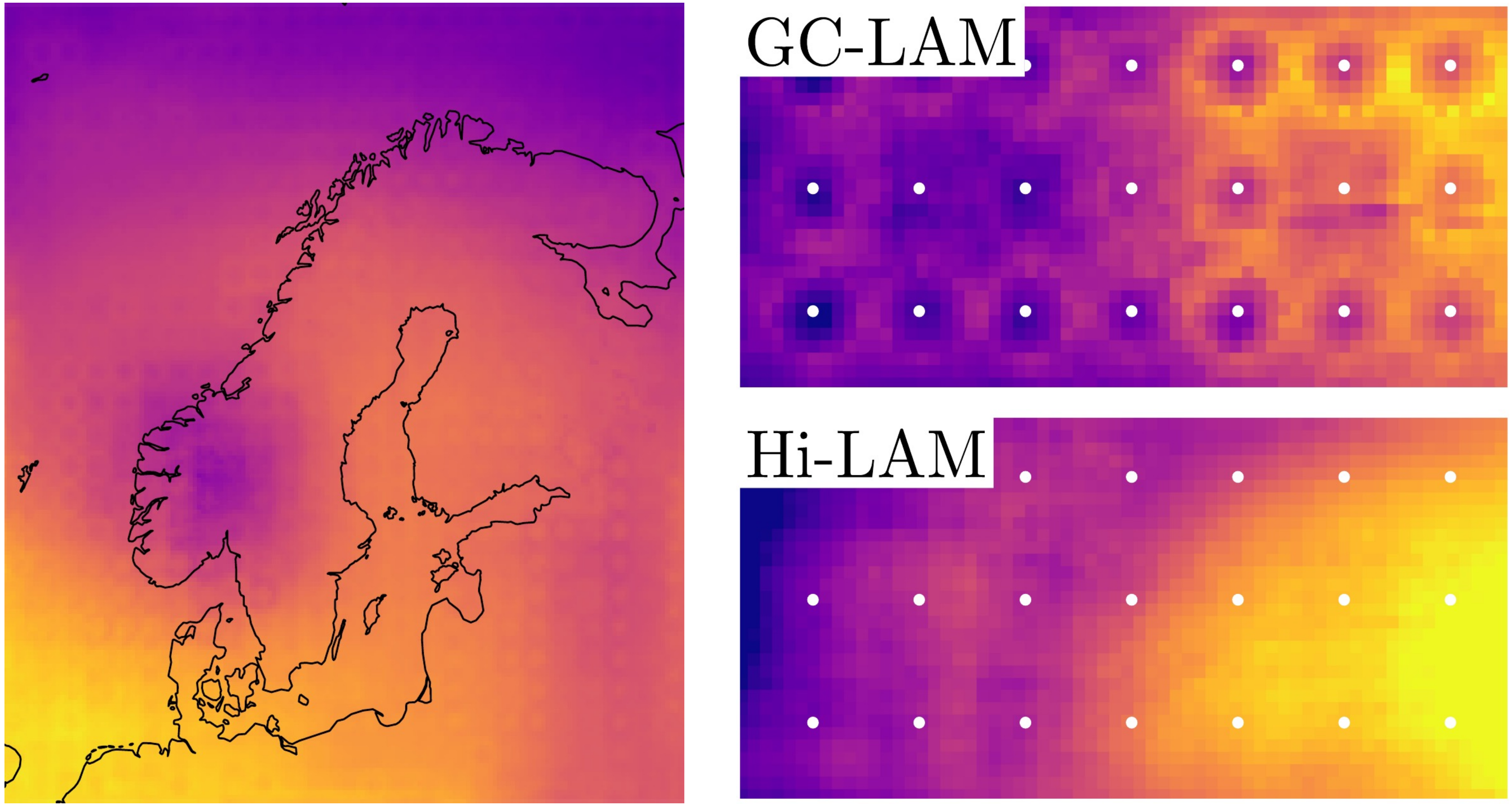
Neural-LAM: Example forecast results





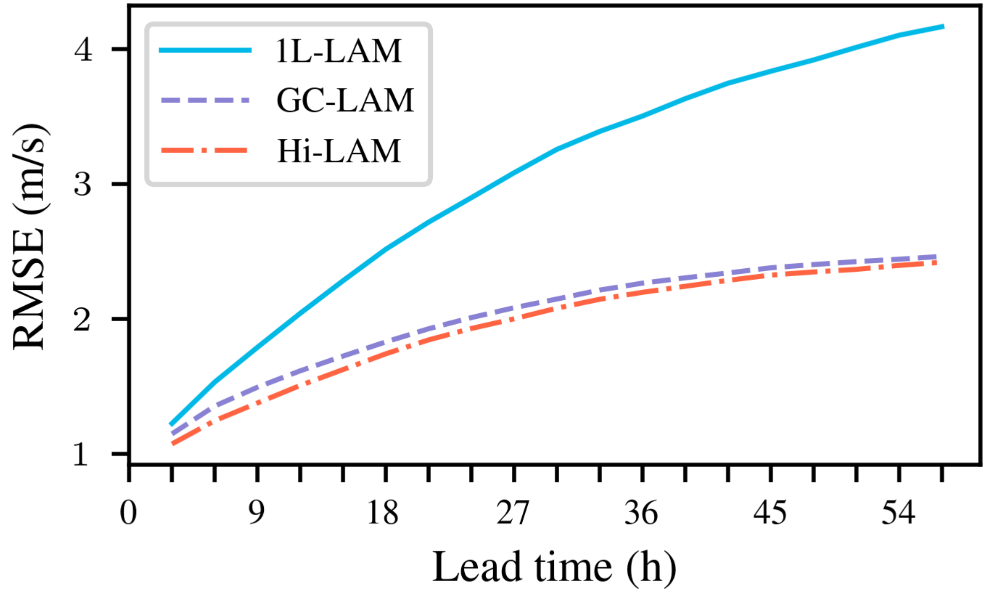
Slide from Joel Oskarsson

Results: Artefacts

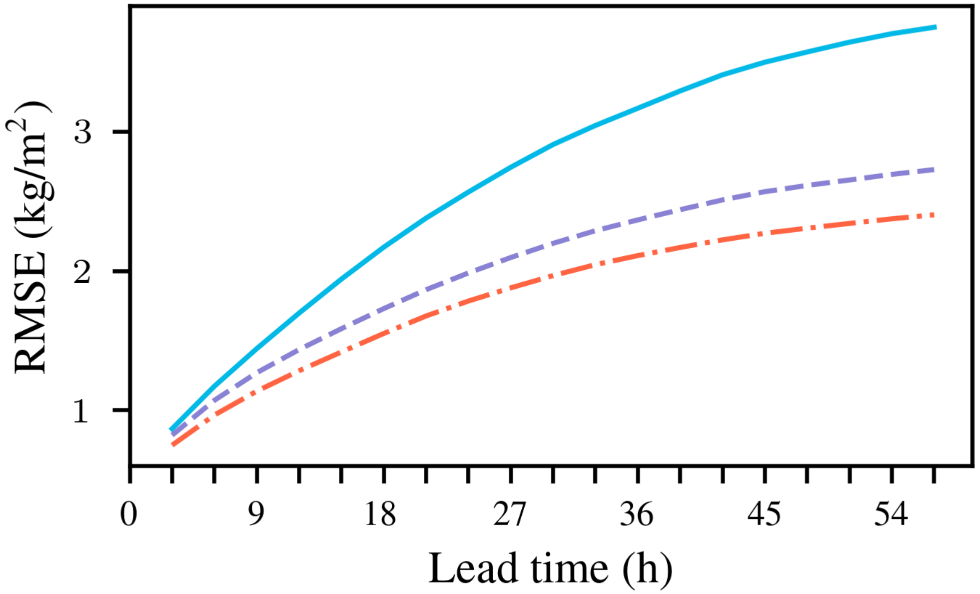


Hierarchical graph appears to avoid near-node artefacts

Results: RMSE



V-component of wind

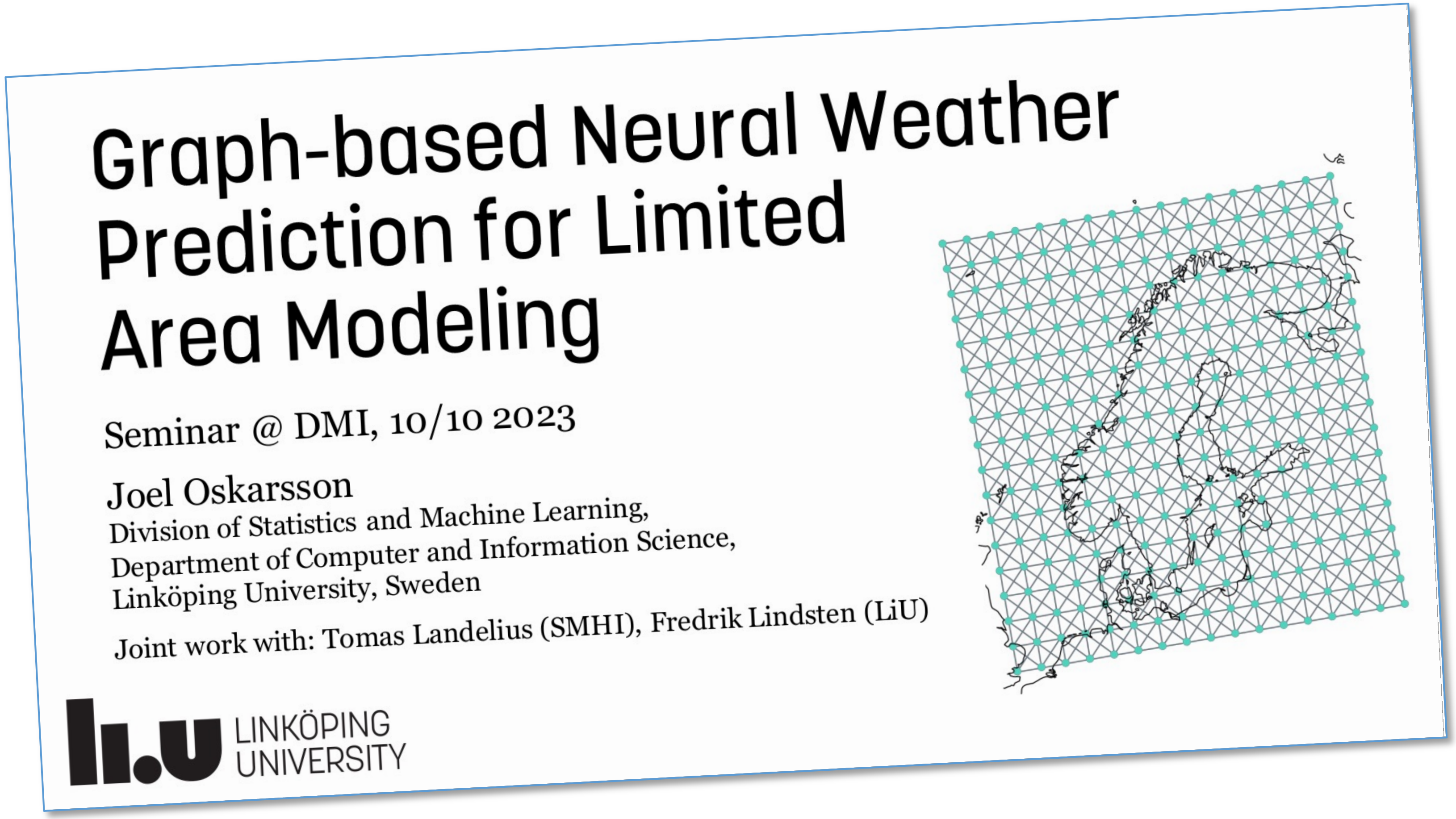


Water vapor

So where are things going?

Next step: LAM machine learning weather model

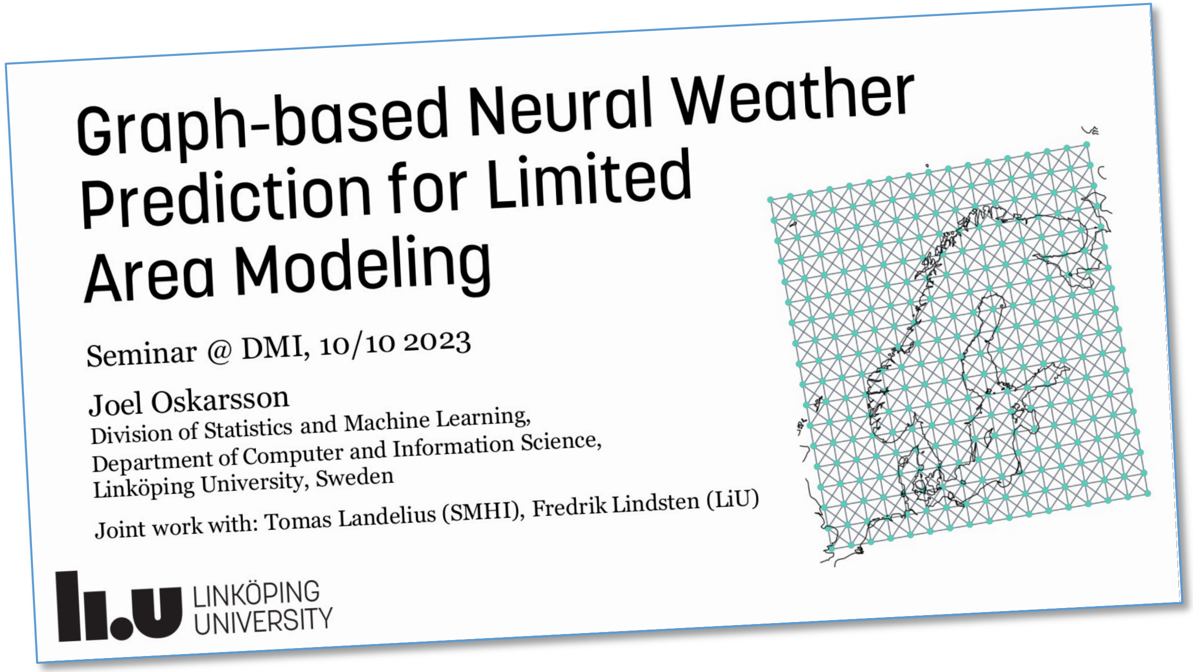
national km-scale data-driven weather model



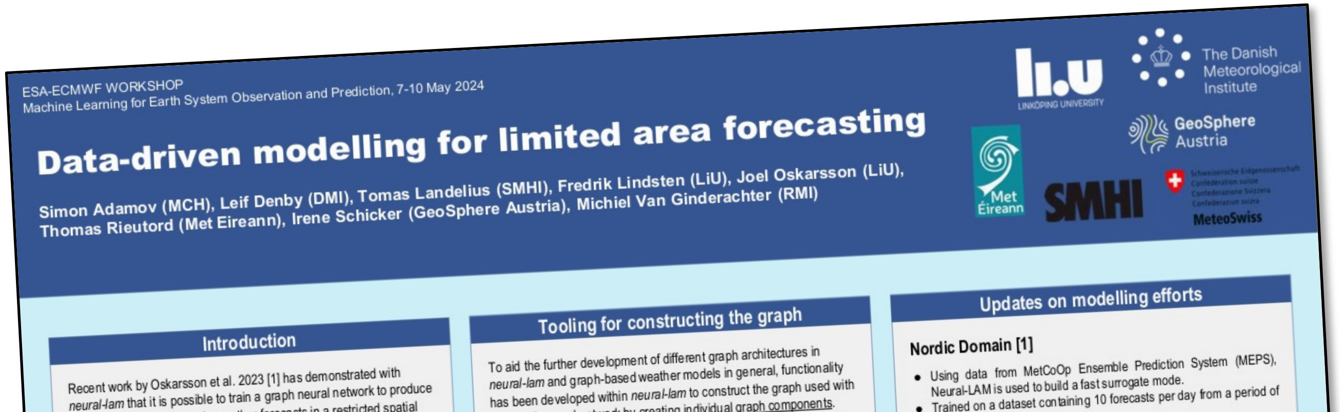
See Oskarsson et al 2023, https://arxiv.org/abs/2309.17370

Next step: LAM machine learning weather model

national km-scale data-driven weather model



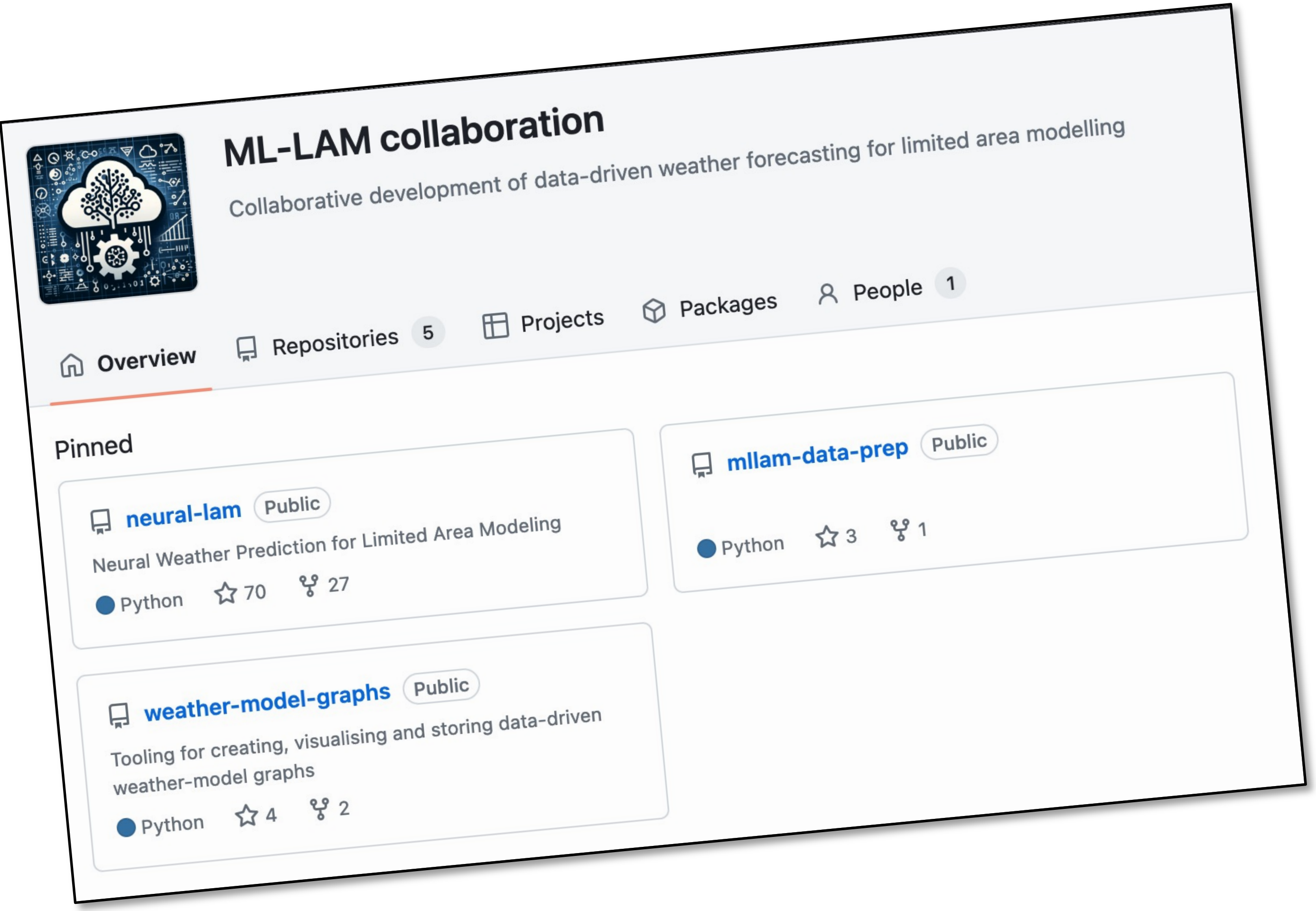
- Started collaboration together with SMHI, MetEireann, Geosphere Austria, RMB-B, SMHI, FMI og MeteoSwiss on further development of Neural-LAM
- Converted 30yr 2.5km DANRA (northern Europe) reanalysis from GRIB to zarr format for preparing training data



Presented at ESA-ECMWF workshop at Esrin, Rome in May

Next step: LAM machine learning weather model

national km-scale data-driven weather model



ML LAM development plan

People

- LCD: Leif Denby, lcd@dmi.dk, DMI (google: leifdenby@gmail.com)
- TL: Landelius Tomas, Tomas.Landelius@smhi.se, SMHI
- JO: Joel Oskarsson, joel.oskarsson@liu.se, https://joeloskarsson.github.io/, LiU
- TR: Thomas Rieutord, thomas.rieutord@met.ie, Met Éireann
- IrS: Irene Schicker, irene.schicker@geosphere.at, GeoSphere Austria (formerly known as ZAMG)
- FB: Fabrizio Baordo, fab@dmi.dk, DMI
- MVG: Michiel Van Ginderachter, michiel.vanginderachter@meteo.be (RMI-B)
- DY: Daniel Yazgi, daniel.yazgi@smhi.se, SMHI
- LH: Leila Hieta, leila.hieta@fmi.fi, FMI
- SK: Santeri Karppinen, santeri.karppinen@fmi.fi, FMI
- MJC: Mat Chantry matthew.chantry@ecmwf.int ECMWF
- ADS: Simon Adamov, simon.adamov@meteoswiss.ch, ETH/MeteoSwiss
- KAH: Kasper Hintz, kah@dmi.dk, DMI
- VC : VIncent Chabot vincent.chabot@meteo.fr, MF
- CB : Clément Brochet clement.brochet@meteo.fr, MF
- MP: Mikko Partio mikko.partio@fmi.fi FMI
- JLC: Jose-Luis Casado jcasador@aemet.es AEMET
- TG: Teresa García mtgarciag@aemet.es AEMET
- OF: Oliver Fuhrer oliver.fuhrer@meteoswiss.ch MeteoSwiss
- LR : Laure Raynaud laure.raynaud@meteo.fr, MF
- SA : Sara Akodad sara.akodad@meteo.fr, MF
- MP : Matthieu Plu matthieu.plu@meteo.fr, MF
- PAA: Andreas Pauling andreas.pauling@meteoswiss.ch MeteoSwiss
- TW: Tobias Wicky tobias.wicky@meteoswiss.ch MeteoSwiss
- SKC: Simon Kamuk Christiansen skc@dmi.dk, DMI
- FG: Frank Guibert frank.guibert@meteo.fr, MF
- RP: Roland Potthast Roland.Potthast@dwd.de, DWD
- ABR: Basavaraj Rajkumar Amogi basavaraj.amogi@wsu.edu WSU
- TB: Tom Beucler, tom.beucler@unil.ch, UNIL
- PG: Pascal Gfäller, pascal.gfaeller@geosphere.at, GeoSphere Austria

github organisation: https://github.com/mllam/

development doc: https://bit.ly/mllam-plan

Probabilistic Weather Forecasting with Hierarchical Graph Neural Networks

Joel Oskarsson
Linköping University
joel.oskarsson@liu.se

Tomas Landelius
Swedish Meteorological and Hydrological Institute
tomas.landelius@smhi.se

Marc Peter Deisenroth
University College London
The Alan Turing Institute
m.deisenroth@ucl.ac.uk

Fredrik Lindsten
Linköping University
fredrik.lindsten@liu.se

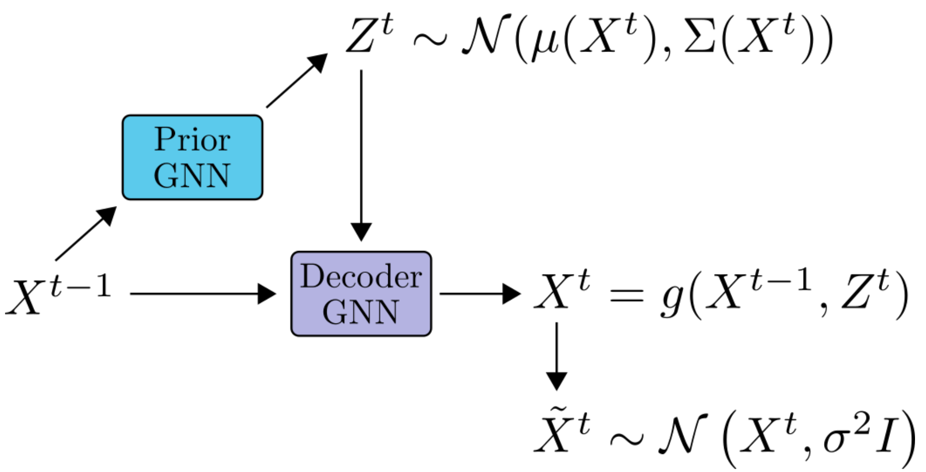
Abstract

In recent years, machine learning has established itself as a powerful tool for high-resolution weather forecasting. While most current machine learning models focus on deterministic forecasts, accurately capturing the uncertainty in the chaotic weather system calls for probabilistic modeling. We propose a probabilistic weather forecasting model called Graph-EFM, combining a flexible latent-variable formulation with the successful graph-based forecasting framework. The use of a hierarchical graph construction allows for efficient sampling of spatially coherent forecasts. Requiring only a single forward pass per time step, Graph-EFM allows for fast generation of arbitrarily large ensembles. We experiment with the model on both global and limited area forecasting. Ensemble forecasts from Graph-EFM achieve equivalent or lower errors than comparable deterministic models, with the added benefit of accurately capturing forecast uncertainty.

Latent variable model

$$p(X^t|X^{t-1}) = \int p(X^t|Z^t, X^{t-1})p(Z^t|X^{t-1})dZ^t$$

Integrated with hierarchical GNN



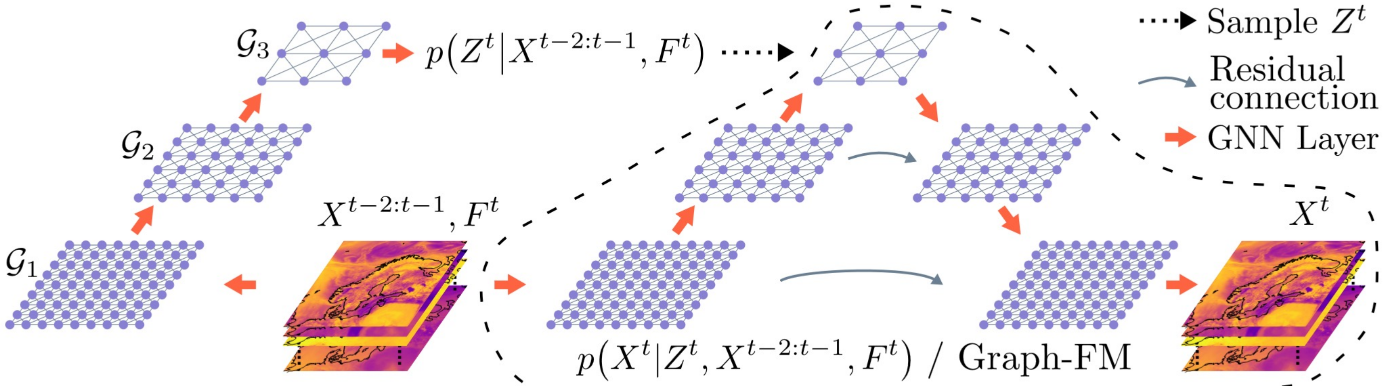
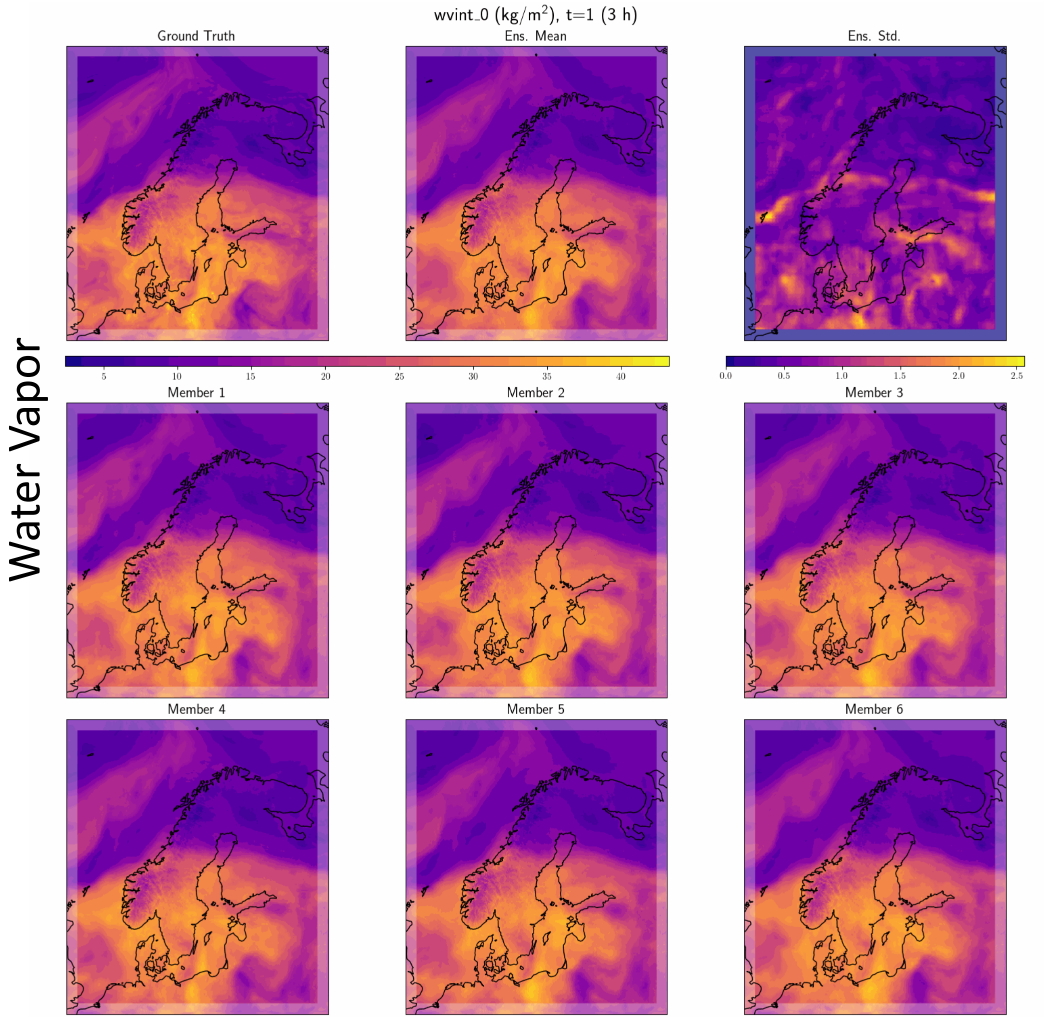
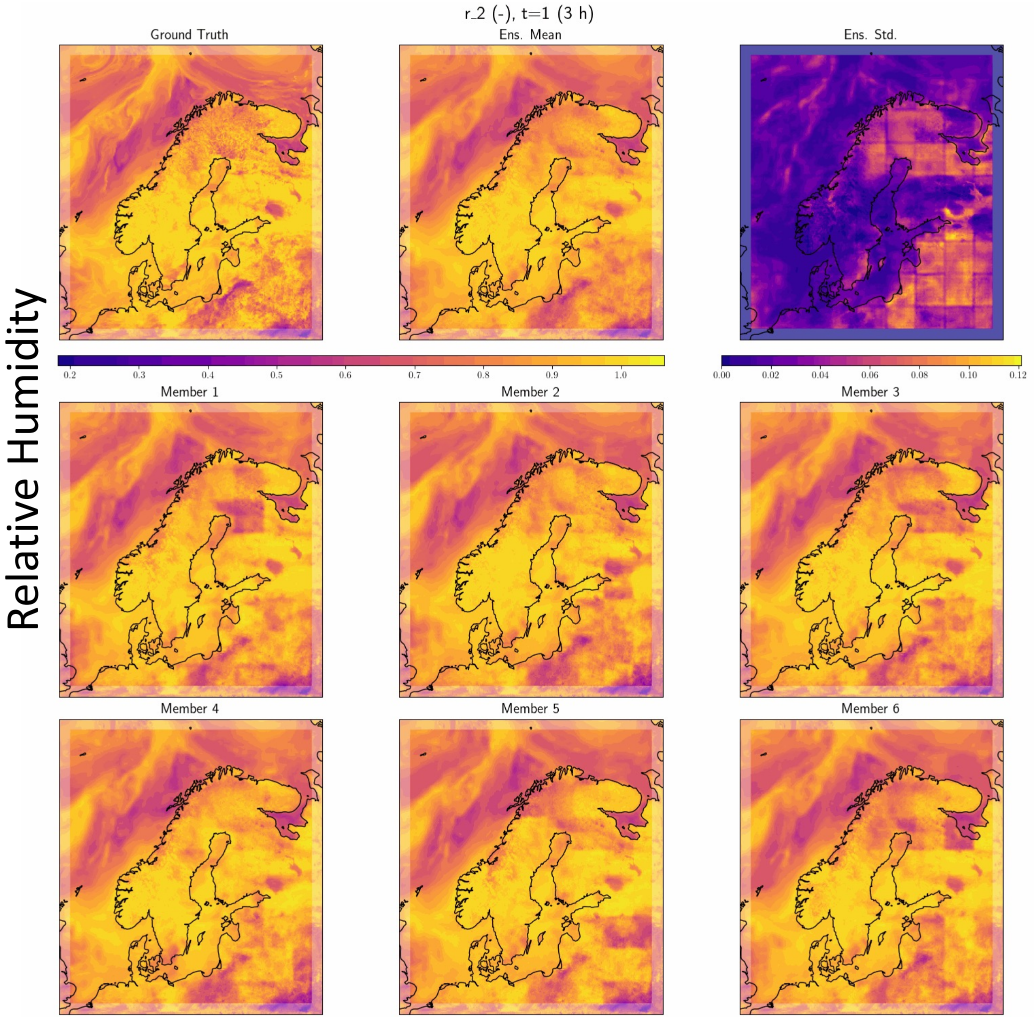


Figure 1: Overview of our Graph-EFM model, with example data and graphs for a Limited Area Model. The corresponding overview for the global setting is given in fig. 5 in appendix C.

- Training
  - Maximize variational bound (ELBO)
  - CRPS fine-tuning

Prel. Results: Ensemble forecasts





Many ongoing ML-weather projects in Europe!

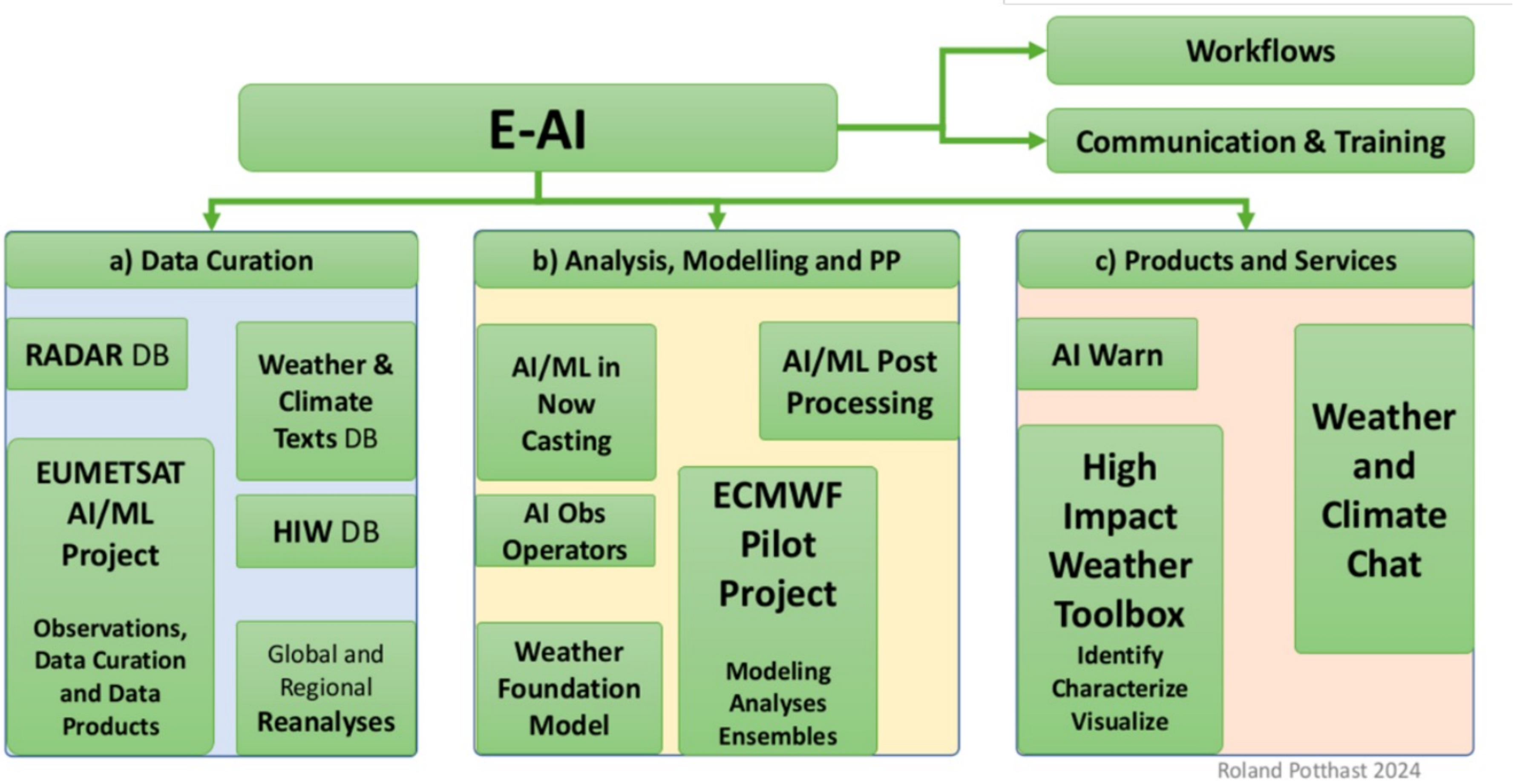


Figure 1 sketch of the E-AI EUMETNET initiative

ECMWF ML Pilot project

Machine Learning pilot project kick-off workshop

ECMWF | Reading | 3-7 June 2024

"The ECMWF ML Pilot project is a Member-led project funded by ECMWF (ECMWF/C/107(23)12 Rev.2) with the objective to foster European collaboration on machine learning (ML) for weather forecasting with a focus on the whole forecasting chain (model, analysis, uncertainty estimation, MLops platforms) and high resolution/limited area modelling, as well as training activities. The project is part of a new EUMETNET optional programme on AI and Machine Learning (E-AI), reflecting the strong initiative and motivation of European NMHS to collaborate and advance on these topics."

ECMWF

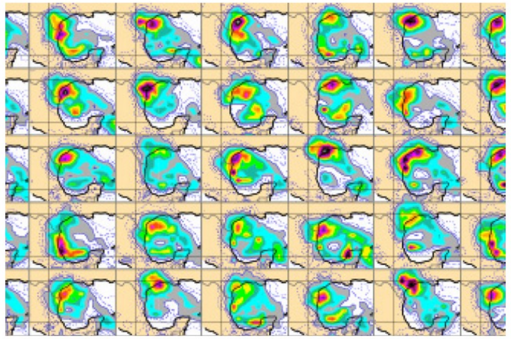
Search site... | Help | Log in

Home | About | Forecasts | Computing | Research | Learning | Publications

Who we are | What we do | Jobs | **Media centre** | Suppliers | Location

AIFS Blog

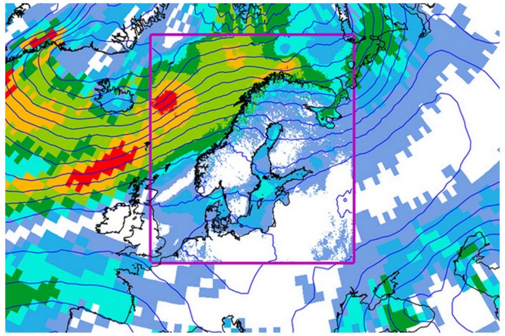
- News
- In focus
- **AIFS blog**
- Science blog
- Key facts and figures
- Media resources
- Videos



Enter the ensembles

21 June 2024

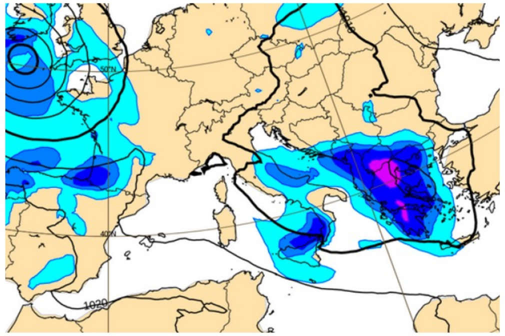
We introduce a first version of an ensemble AIFS, explain how it works, show some early results and explain where you can view charts.



Data-driven regional modelling

23 April 2024

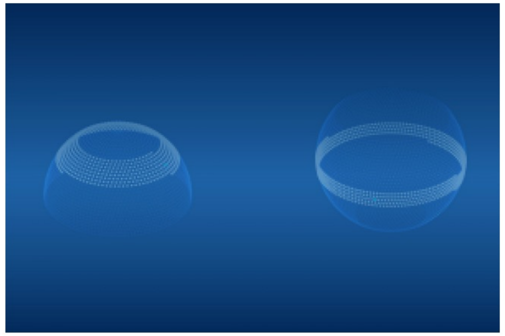
With colleagues from MET Norway, we



It's rain(ing) data

4 March 2024

We provide an update on the AIFS, including



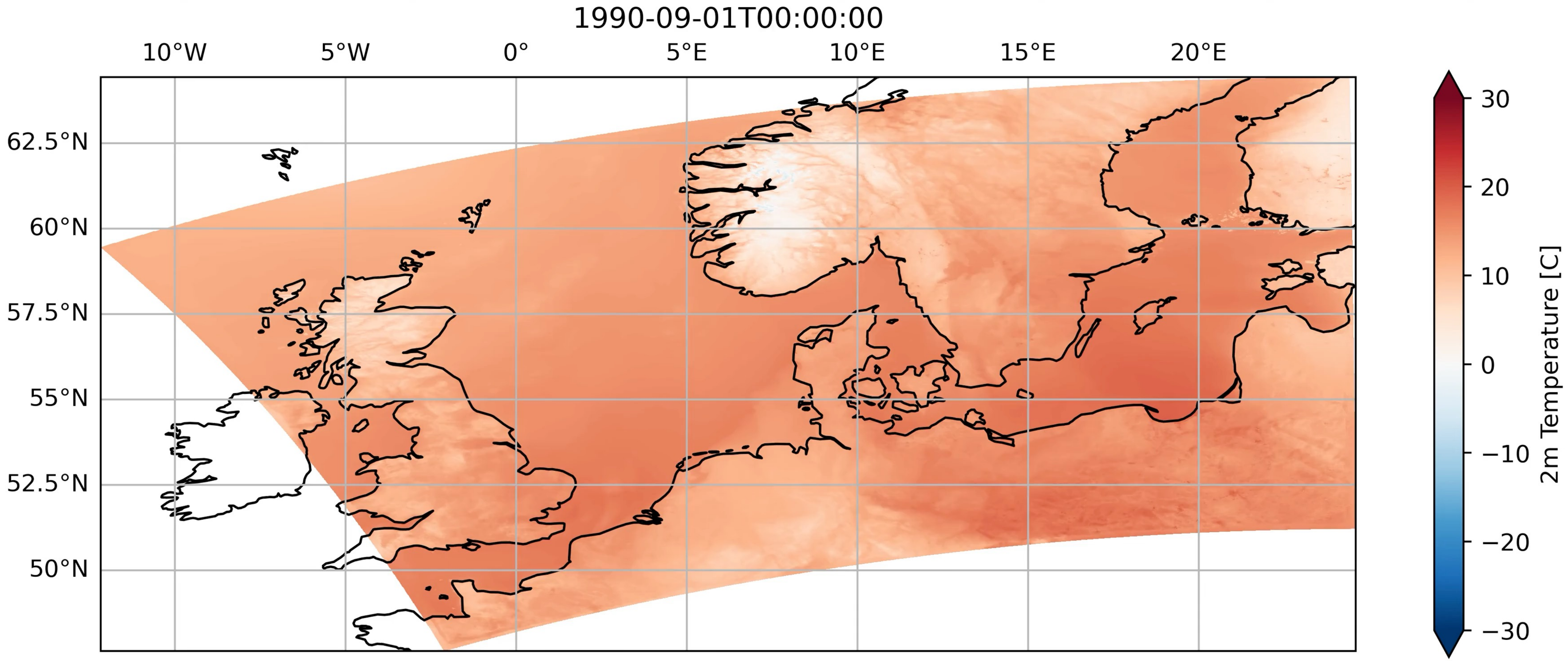
First update to the AIFS

16 January 2024

We have introduced a new version of the

https://www.ecmwf.int/en/about/media-centre/aifs-blog

DANRA (2.5km reanalysis, 30 years)



First year of DANRA 2m temperature: https://www.youtube.com/watch?v=NNpPamhi2co

DMI NWP Group ML Roadmap

v0.1.2

	2024 Q1	Q2	Q3	Q4	2025 Q1	Q2	Q3
2m temperature							
Neural-LAM forecast	DANRA reanalysis in zarr, code refactor	Training data prep and first training	Global model ICs + BCs, real-time setup	Domain transfer learning + validation	Refactor VAE arch & COMEPS ensemble in zarr	real-time ensemble inference	DMI surface obs archive in zarr, train/valid on DMI archive
10m wind							
Neural-LAM forecast	DANRA reanalysis in zarr, code refactor	Training data prep and first training	Global model ICs + BCs, real-time setup	Domain transfer learning + validation	Refactor VAE arch & COMEPS ensemble in zarr	real-time ensemble inference	DMI surface obs archive in zarr, train/valid on DMI archive
Surface precipitation							
LDCast nowcast			RadKlim & DMI radar archive in zarr, code refactor	Train in Seamless	Real-time inference from DMI radar obs		
Neural-LAM forecast	code refactor	DANRA forecast in zarr	RadKlim & DMI radar archive in zarr	Train/validation on RadKlim/DMI archive	Real-time inference from DMI radar obs		
Surface Irradiance							
SHADECast nowcast		SARAH-3 reanalysis in zarr	baseline with solarSTEPS and SARAH-3, SHADECast refactored	MSG derived surface irradiance emulation	real-time inference from MSG retrievals	Comparison of Neural-LAM and SHADECast	Refactor into Seamless
Neural-LAM forecast	code refactor	DANRA forecast & surface obs in zarr	SARAH-3 reanalysis in zarr	Train/validation on DANRA and SARAH-3	Real-time inference from global ICs + BCs		
Lee-wave rotor risk							
LeeWaveNet		Code refactored & containerised	real-time inference on NWP model output				

We're hiring!

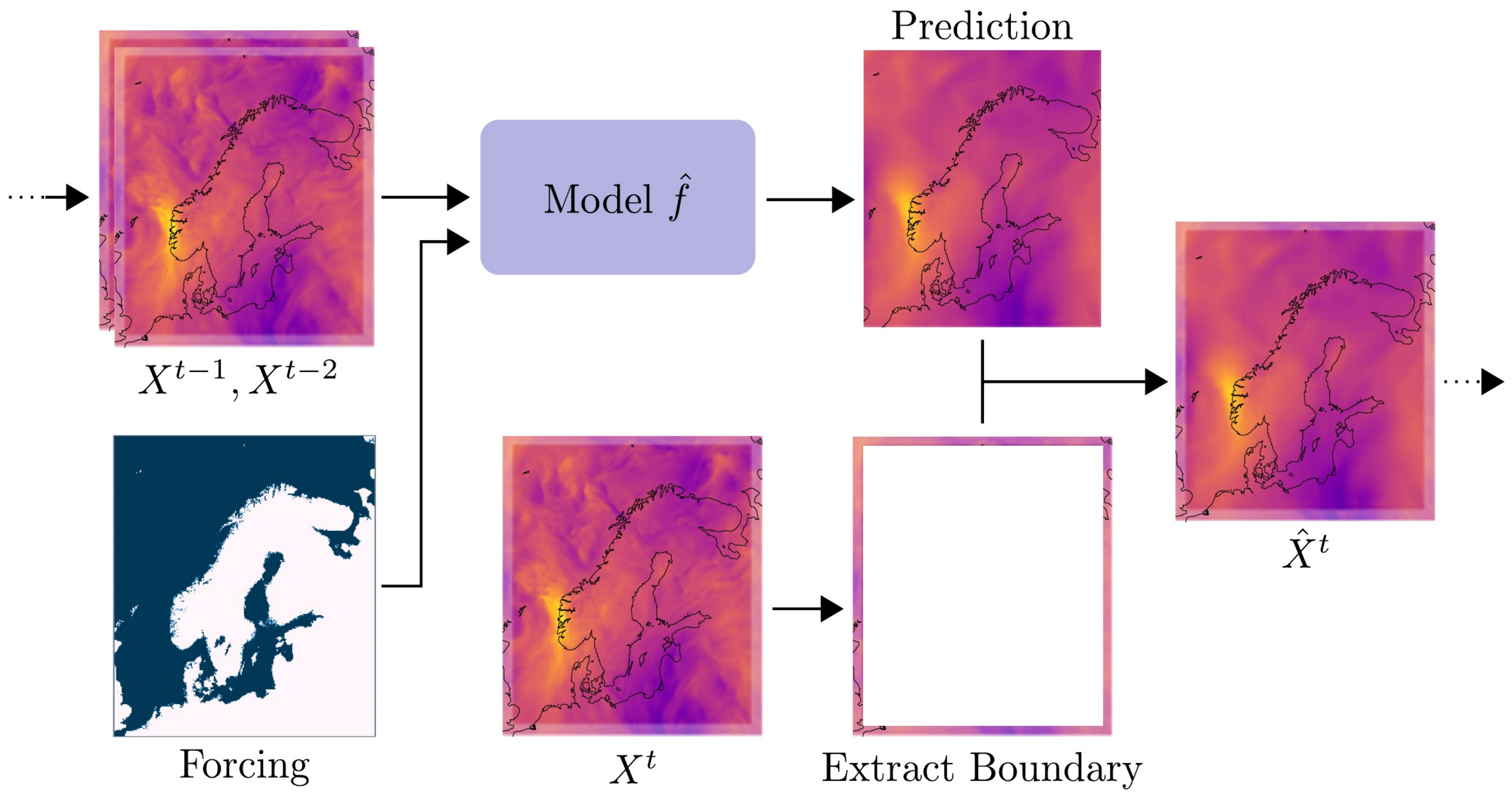
Where will things go from here?

- Forecasting directly from observations
  - AtmoRep: Transformer-based synop predictions from satellite radiances
  - Aarkwark weather: Convolution-based synop -> analysis -> forecast
- Using the latent-space:
  - Forecasting, enquery, physical constraints, combining heterogenious data
- Increased focus on observations
- Imposing physical structure in architecture, loss, etc
- Km-scale forecasting
  - Convection, precipitation

Best courses to get start (in my opinion) - both free:
https://fast.ai: *"Practical Deep Learning for Coders"* and *"From Deep Learning Foundations to Stable Diffusion"*

github.com/leifdenby lcd@dmi.dk @leifdenby@mas.to

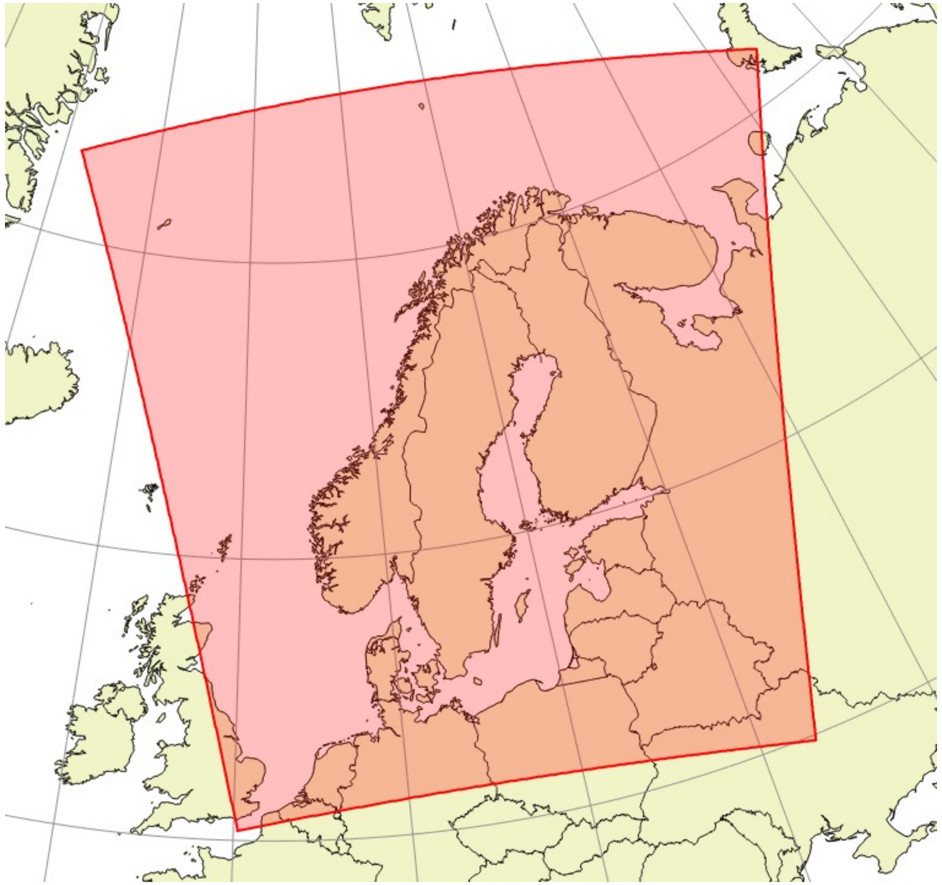
Boundary forcing



MetCoOp Ensemble Prediction System (MEPS)

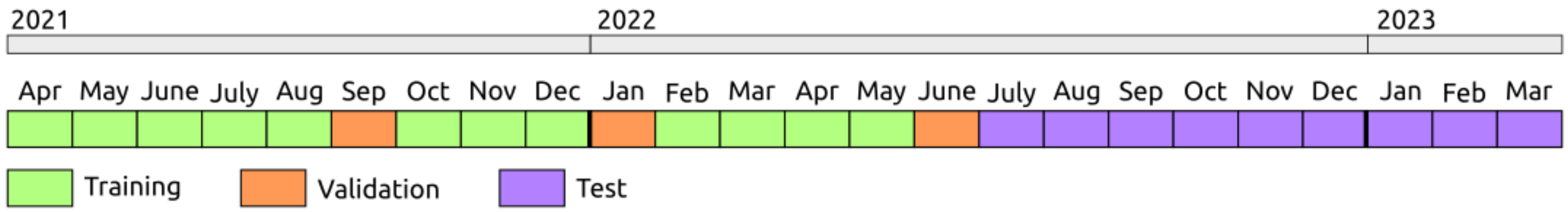
- 960×1080 (2.5 km) x 65 vertical
- Non-hydrostatic dynamics
- IFS HRES and IFSENS boundaries
- 66h ensemble forecast run hourly

- Idea: Emulate with fast deep learning model



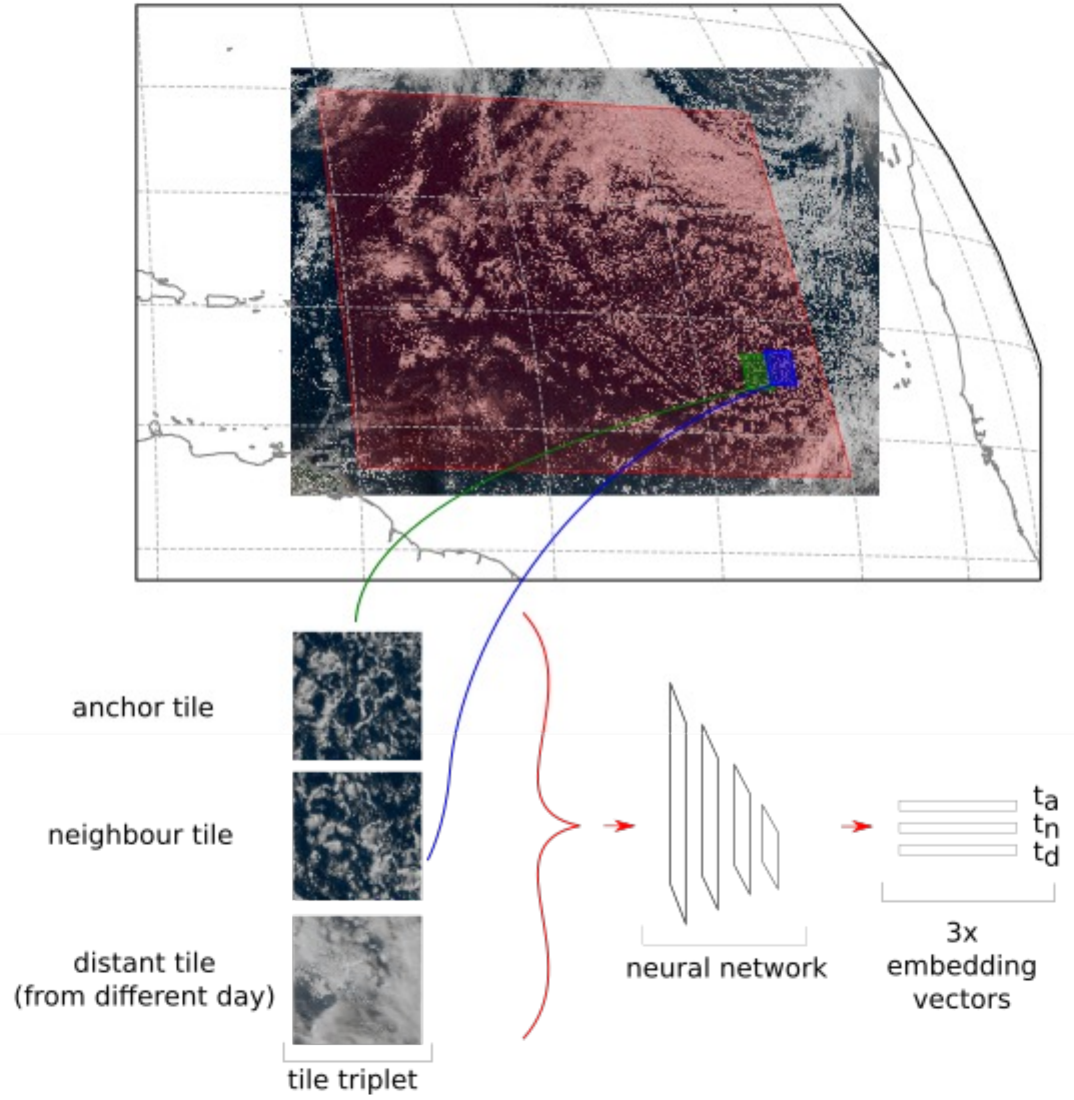
Dataset

- Subset of atmospheric variables used:
  - Pressure (surface, MSL)
  - Geopotential (500, 1000 hPa)
  - Wind (lev 65, 850 hPa)
  - Temperature (2m, lev 65, 500, 850 hPa)
  - Relative humidity (2m, lev 65)
  - Total water vapor column
  - Net short- and longwave solar radiation
- Spatial down-sampling ×4 (10 km)
- Additional forcing inputs:
  - TOA radiation, time, land/water mask
  - Forecast as boundary forcing
- 10 forecasts per day from ~2 years
- 3h time-steps



Model training

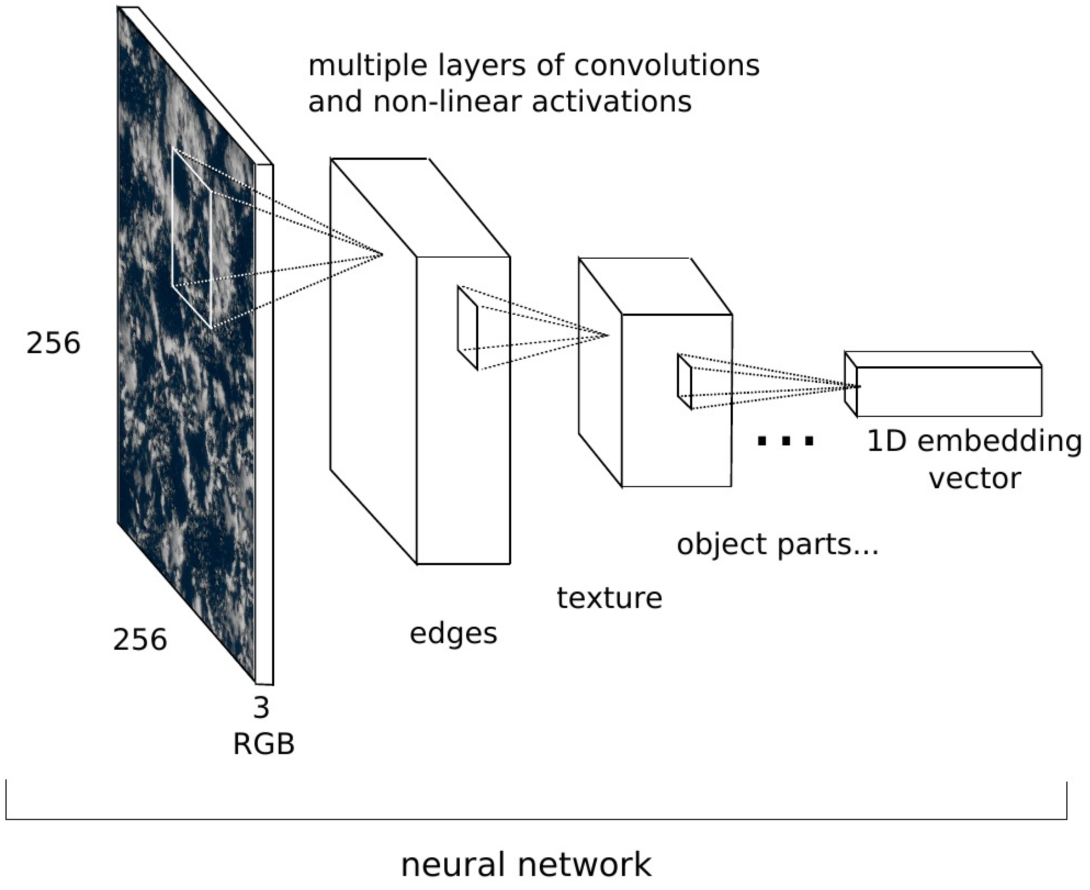
- Every training example consists for three tiles (triplet) the *anchor* (t_a), *neighbour* (t_n) and *distant* (t_d) tiles.
- Use loss function which optimises for *anchor* and *neighbour* tiles being close in embedding space and *distant* tile being far away (measured by Euclidian distance):



$$L(t_a, t_n, t_d) = max(0, ||f_\theta(t_a) - f_\theta(t_n)||_2 - ||f_\theta(t_a) - f_\theta(t_d)||_2 + m)$$

See Denby 2020 (10.1029/2019GL085190) for details

Using convolutional network to produce embedding



- Training done with `pytorch` (and `pytorch-lightning`)
- Use pre-trained Resnet34 (transfer learning)
- Replace last layer by fully connected layer
- Currently using N_d=100 embedding length
  - Experimenting with shorter embedding vectors

- Each layer contracts information from a finite part of image into a single value
  - These are composited over multiple layers to produce more complex features