

Integration of ML in NOAA DA systems

ECMWF Annual Seminar 2026

Daryl Kleist, Deputy Director
NOAA/NWS Office of Modeling and Development

U.S. Department of Commerce
National Oceanic and Atmospheric Administration
National Weather Service

Many contributors (people doing the actual work)

NOAA/NWS OMD: Daniel Holdaway, Russ Treadon, Andrew Collard, Guillaume Vernieres, Azadeh Gholoubi (Axiom), Laura Ko (Lynker), Jeff Whitaker (UCAR), Jack Woollen (Lynker), Xin Jin (Lynker), Miodrag Rancic (Lynker), Brett Hoover (former), Annette Gibbs (Lynker), Ting Lei (Lynker)

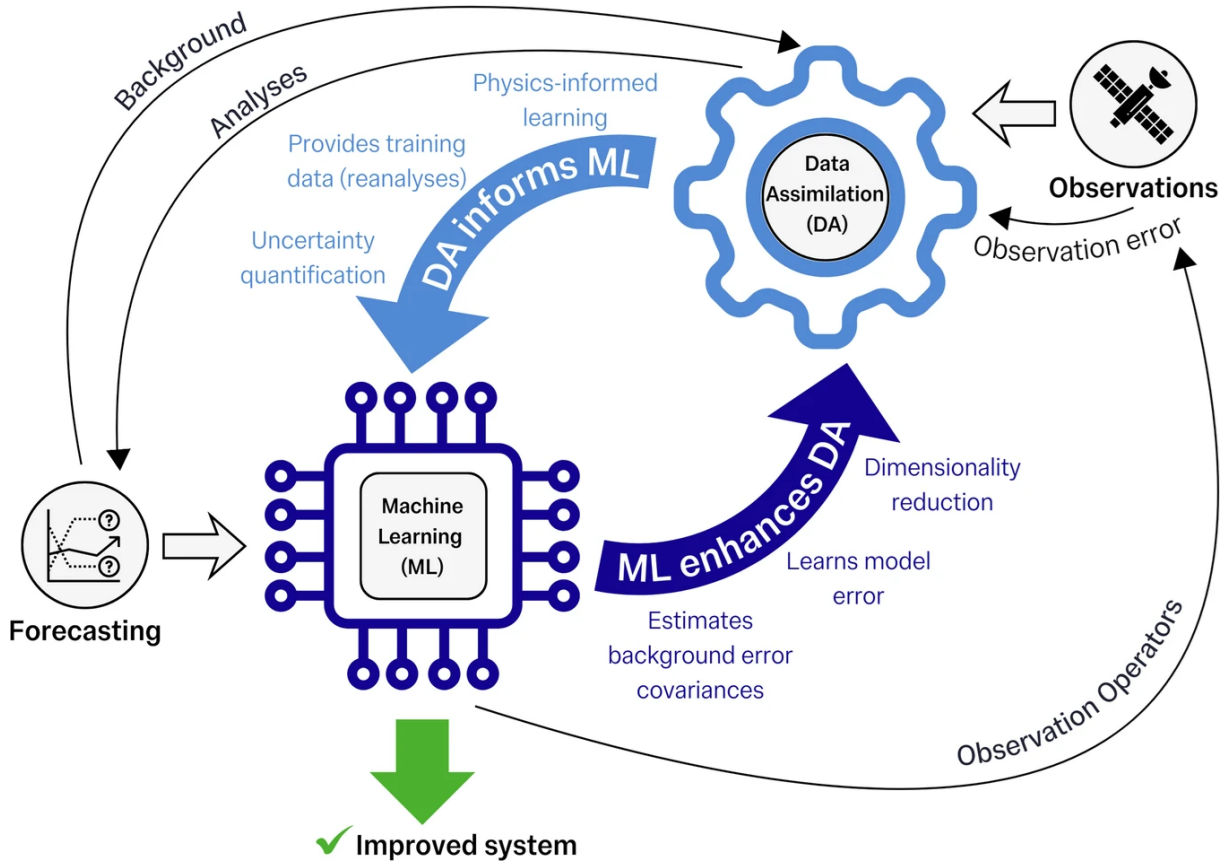
NOAA/OAR PSL: Sergey Frolov, Laura Slivinski, Chong-Chi Tong (CIRES)

Sorcerer: Xiaoxu Tian (previously EMC / OMD)

JCSDA: Ben Johnson, Victor Marchais, Lucas Howard (U.Col)

Convergence?

Arcucci et al., 2026: The convergence of machine learning and data assimilation in Earth system science (10.1038/s44387-026-00107-0).



ML for Operational DA Priorities

From EMC (now OMD) DA Strategy Section 3.6 Summary Table (https://doi.org/10.25923/kqra-6z66)

01 Observations
quality control, data selection, bias correction, superobservations, extraction of maximal information content, anomaly detection and operational monitoring;

02 Forward operator emulation
computational efficiencies, replacement for complex operators;

03 Background error
computational efficiencies, multivariate aspects and coupled assimilation, parameter estimation for error models;

04 Background
dynamic downscaling, bias correction;

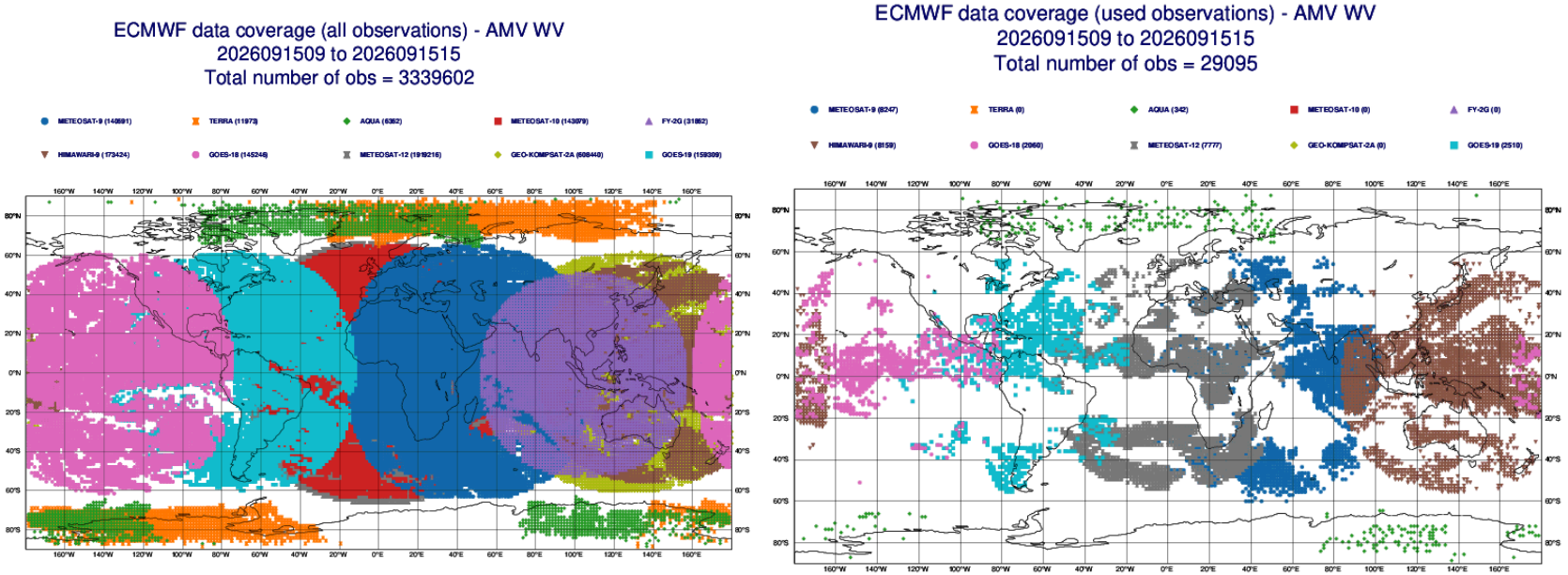
05 Model error
estimation and correction;

06 Emulator exploitation
replacement for TL/AD in 4DVar, efficient creation of ensembles;

07 Hybridization
explicit blending of ML and DA; joint frameworks; pathways to going directly from observations to simulation/emulation.

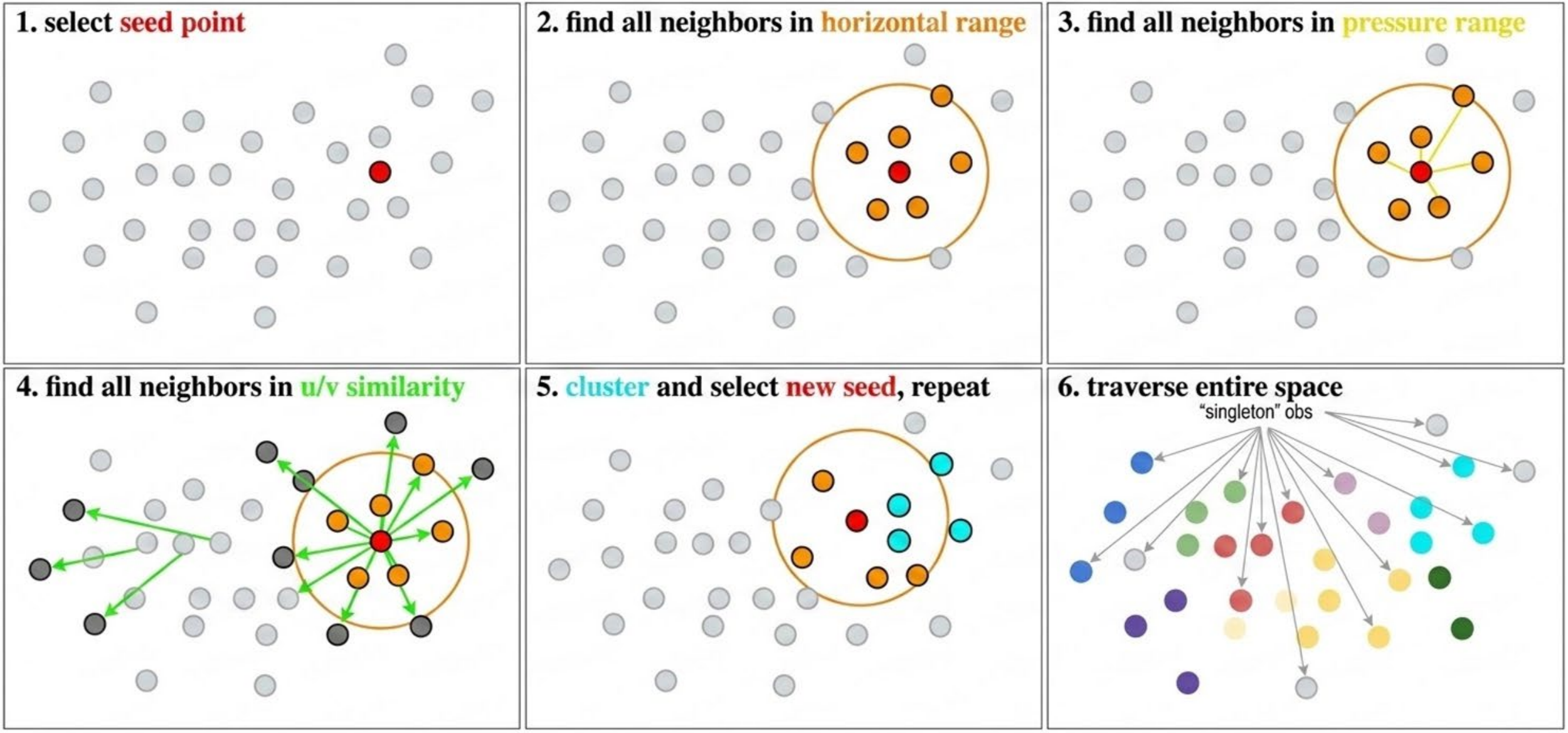
Atmospheric Motion Vectors

- Large volume and more satwinds all the time as new satellites are launched, higher-resolution sensors are used, and new methods are developed to compute AMVs
- AMVs, perhaps more so than other “conventional” wind observations, have large error correlation length scales, typically estimated at 100+ km
- For the sake of timeliness as well as proper accounting of observation error, AMVs need to be filtered by grouping them together in some fashion and then either selecting one “best” satwind (thinning) or averaging AMVs together to form a single observation (super-obbing)



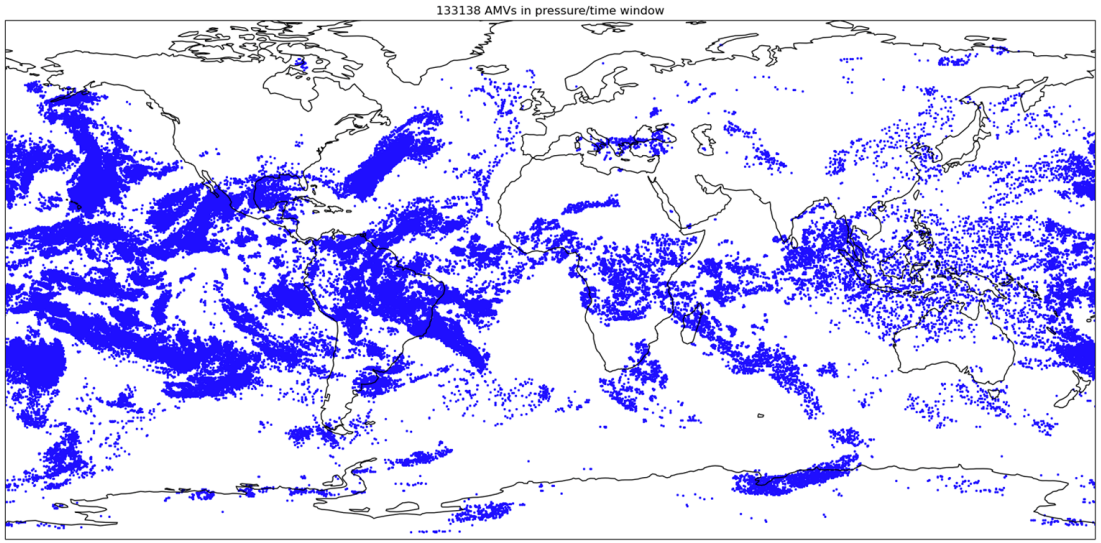
Water Vapor AMVs available (left) and assimilated (right) for 1200 UTC 15 September 2026, from ECMWF Observation Monitoring page

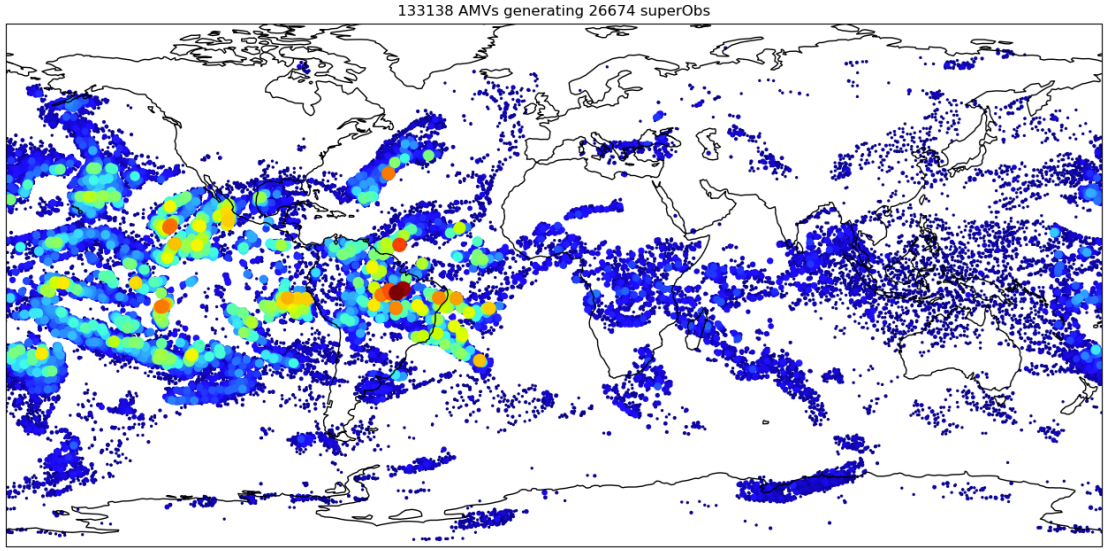
Clustering algorithm

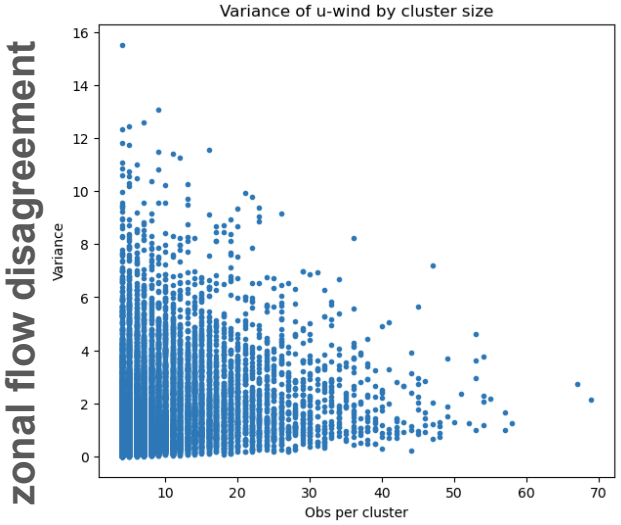


Atmospheric Motion Vectors: Clustering for Superobservations

133,000 AMVs ------ clustering --> 27,000 super observations







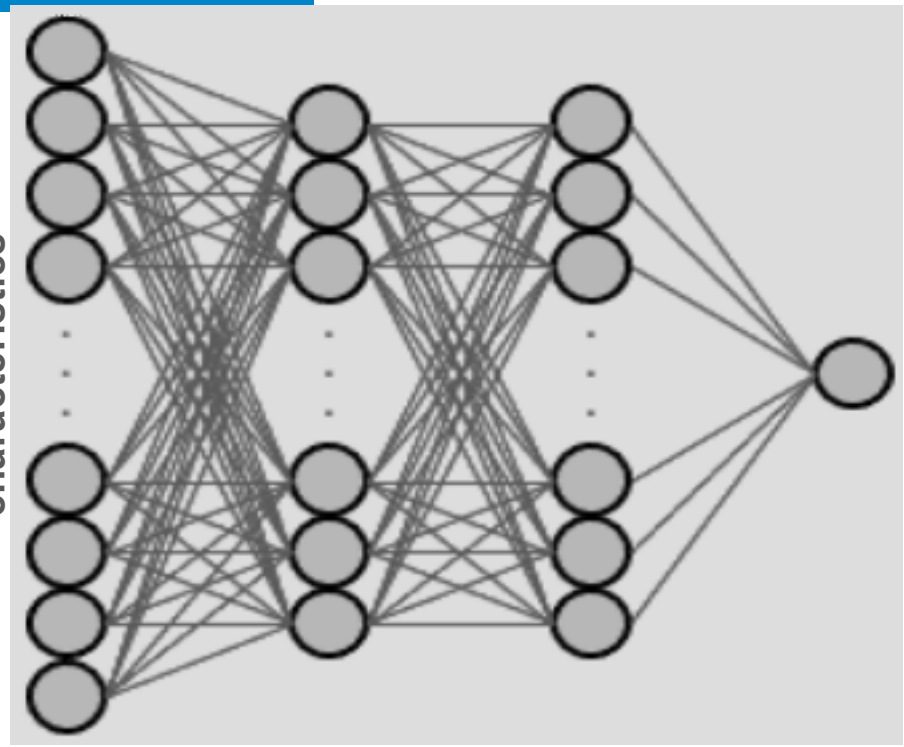
A clustering approach for AMV superobs:

- reducing data volume (operational efficiency)
- combining similar, nearby AMVs (focus on information content)
- addressing spatial error correlation (alternative to thinning)
- producing unified set of observations from disparate AMV sources

Atmospheric Motion Vectors: Clustering for Superobservations

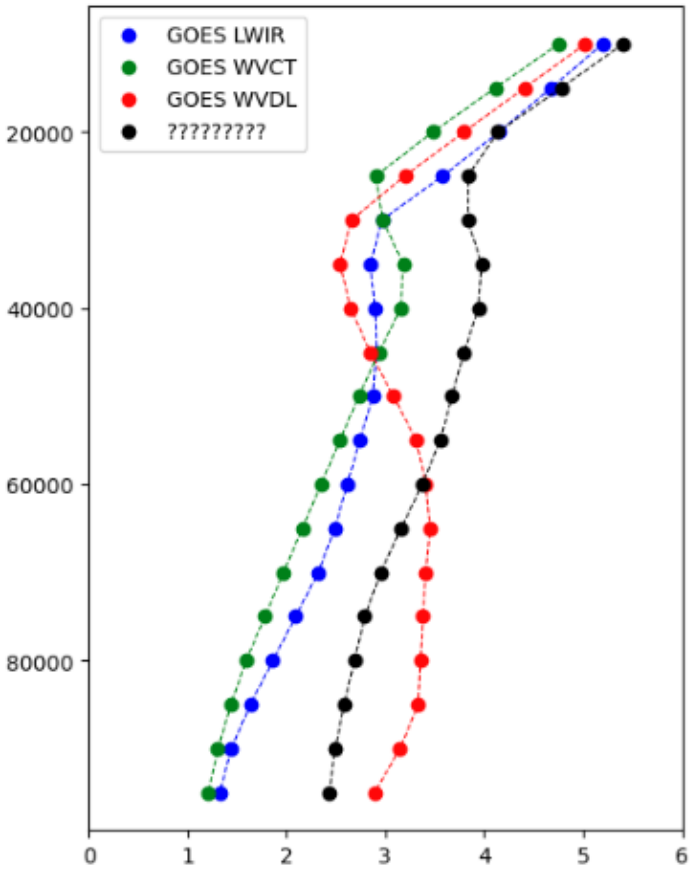
33 inputs predictors:

- **Super-ob** u, v, location in time and space
- **Super-ob complexion**: number of satwinds, number of types, contains type-1, type-2, etc
- **Group characteristics**: variance among group in u, v, location in time and space



fully connected, deep learning neural network

superob error estimate



- Error estimation/assignment is critical for assimilation
- Ability to learn type-dependency on error; differences among GOES types is similar to observed in departure statistics
- With same profile but no ob type information, can predict generic error profile (upper right, black) that appears reasonable

Two GPU implementations of CRTM — one learned, one exact

"CRTM-AI": FAST · PROBABILISTIC

Neural-network emulator of CRTM (L. Howard, V. Marchais, JCSDA)

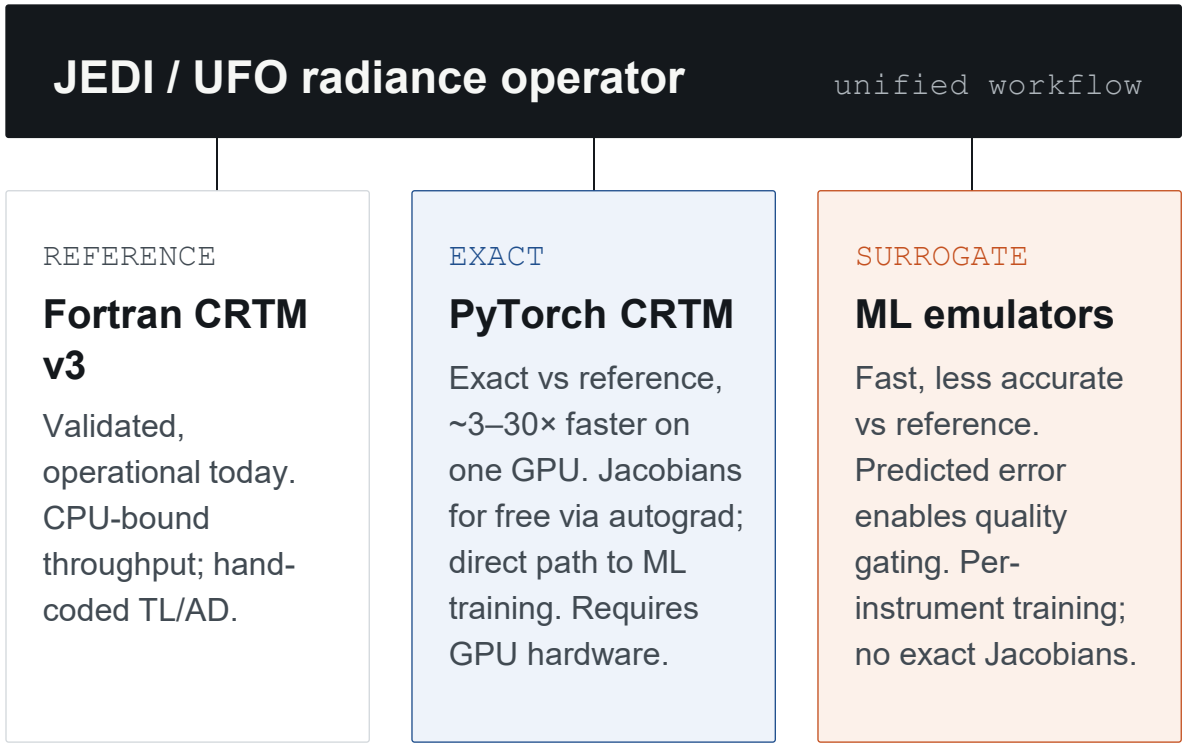
- Motivation: < 1 % of available radiances are assimilated today; the operator's CPU cost is the limit
- Probabilistic network emulates CRTM for GOES ABI (ch 7–16): ~0.3 K RMSE, < 0.1 K clear sky on 9 of 10 IR channels, with a predicted error per channel
- ~10× faster than CRTM in initial tests; the fast end of the design space the PyTorch port brackets
- Howard et al., AIES 2026; AMS 105 / JCSDA Symposium 2025 (with Subramanian, CU Boulder)

"CRTM-TORCH": EXACT · DIFFERENTIABLE

GPU-native PyTorch port of CRTM (X. Tian)

- Matches CRTM v3.1.3 to ~10^{-11} K (brightness temperature and Jacobians, float64)
- One A100: > 150,000 profiles/s clear sky, ~13,000 all sky; 30-core Fortran: ~4,800 all sky
- All-sky Jacobians validated (IR and MW); found two bugs in the Fortran adjoint
- Abstract submitted, AMS 2027 (Johnson and Tian)

One UFO observation operator, three switchable CRTM backends

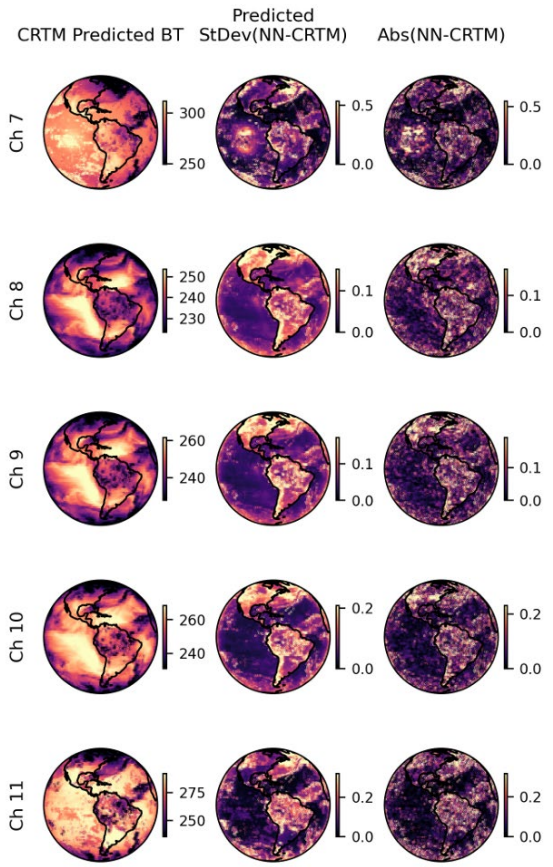


JEDI INTEGRATION AND EVALUATION

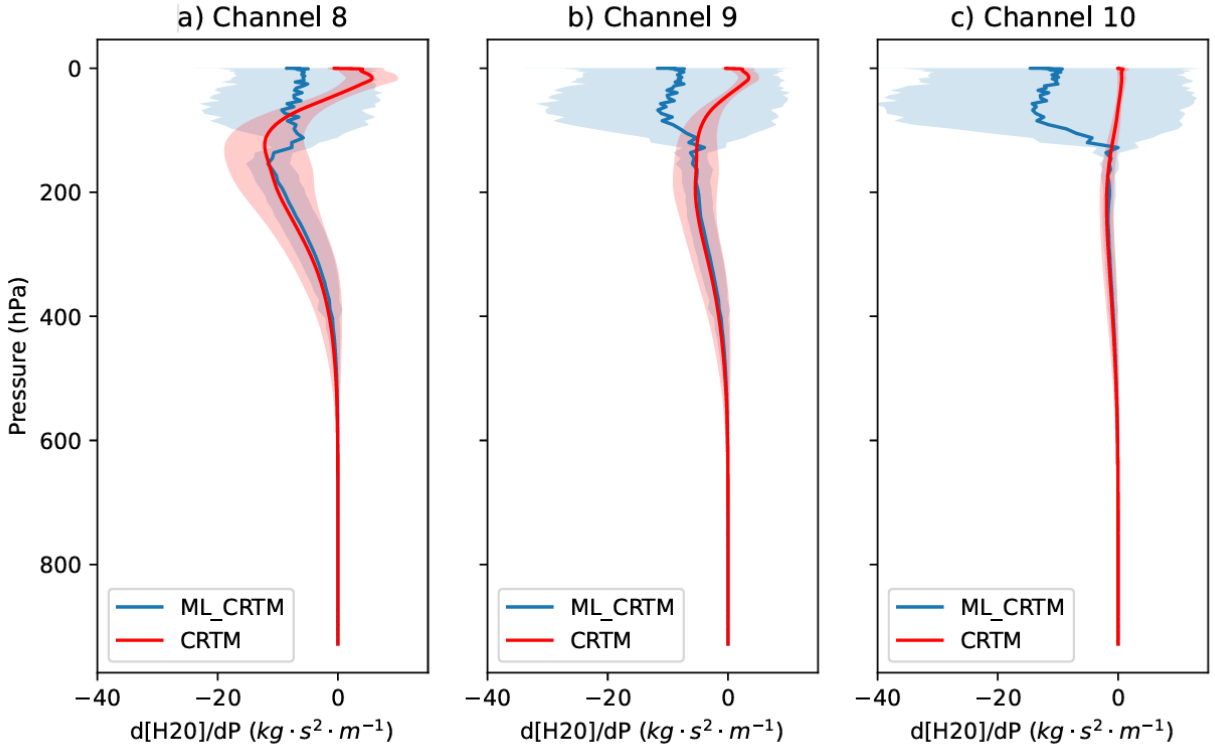
- Integration and evaluation of PyTorch CRTM directly in JEDI and UFO along side Traditional CRTM
- Backend selected at run time within a single observation workflow; same inputs, same diagnostics
- Codebase open-sourced with the publications and official releases

CRTM · GPU ACCELERATION AND MACHINE LEARNING

Howard et al., 2026: Probabilistic Emulation of the Community Radiative Transfer Model (arXiv:2504.16192v1).



LEFT: GOES ABI Predicted BT, predicted error standard deviation, and actual absolute error (left to right) for each channel. (From preprint Figure 4).



ABOVE: Mean Jacobians plus or minus a standard deviation for the upper-level (channel 8, panel a, mid-level (channel 9, panel b) and lower-level (channel 10, panel c). Preprint Figure 6.

AI Balance Operators for Coupled DA

What's the idea?

- Balance operators needed between ocean and sea ice
- Use a non-linear regression to predict one variable using a select number of other variables (lower-triangular).
- Assert the Jacobian structures are not too noisy from AI.
- Enables strongly coupled ocean-sea ice data assimilation with diagonal correlation operator.
- Added a "TorchBalance" operator to the JEDI-SABER repo.

*Ocean-Seaice Parametric **B***

$$\mathbf{B} = \mathbf{K} \cdot \mathbf{D} \cdot \mathbf{C} \cdot \mathbf{D} \cdot \mathbf{K}^{\mathrm{T}}$$

Physics (vertical displacement)

Physics (steric height)

Physics (geostrophy)

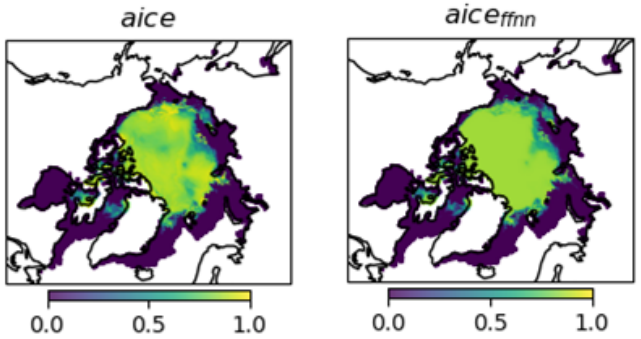
ML or Physics

$$\mathbf{K} = \begin{array}{c} \\ \delta T \\ \delta S \\ \delta \eta \\ \delta u \\ \delta v \\ \delta h_i \\ \delta h_s \\ \delta c \end{array} \begin{array}{c} \begin{array}{cccccccc} \delta T & \delta S & \delta \eta & \delta u & \delta v & \delta h_i & \delta h_s & \delta c \end{array} \\ \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial S}{\partial T} & I & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial \eta}{\partial T} & \frac{\partial \eta}{\partial S} & I & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial u}{\partial T} & \frac{\partial u}{\partial S} & \frac{\partial u}{\partial \eta} & I & 0 & 0 & 0 & 0 \\ \frac{\partial v}{\partial T} & \frac{\partial v}{\partial S} & \frac{\partial v}{\partial \eta} & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ \frac{\partial c}{\partial sst} & \frac{\partial c}{\partial sss} & 0 & 0 & 0 & \frac{\partial c}{\partial h_i} & \frac{\partial c}{\partial h_s} & I \end{bmatrix} \end{array}$$

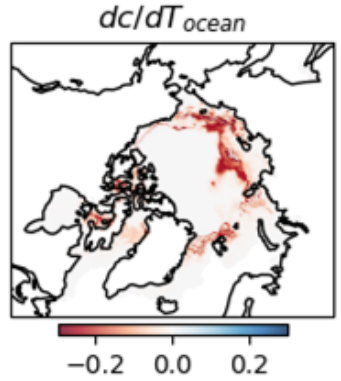
AI Balance Operators for Coupled DA

- **Seaice emulator:** Approximated using a simple feed-forward neural network (**FFNN**), implemented with **LibTorch** in C++ to facilitate interfacing with **JEDI**.
- The training and validation data from **coupled UFS** backgrounds
- **Balance:** The Jacobian of the emulator is computed using **LibTorch's** automatic differentiation tools.
- Implemented in **JEDI/SOCA**

Ice concentration emulator results

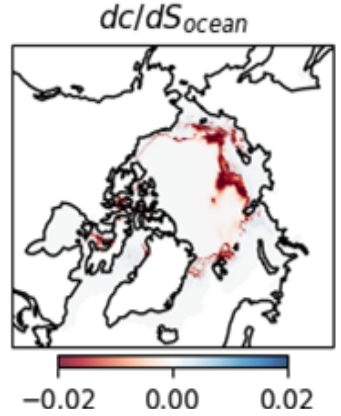


Example of some of the elements of the Jacobian (right)



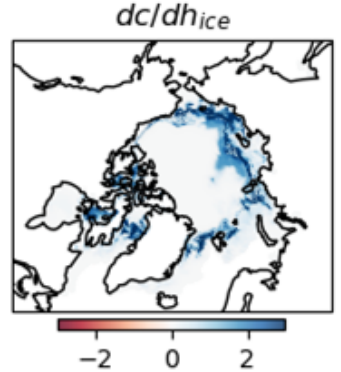
$$\delta c = \frac{\partial c}{\partial T_o}\delta T_o + \ldots$$

A **positive** SST increment within the marginal ice zone results in a **negative** contribution to the sea ice concentration.



$$\delta c = \frac{\partial c}{\partial S_o}\delta S_o + \ldots$$

A **positive** SSS increment within the marginal ice zone results in a **negative** contribution to the sea ice concentration.



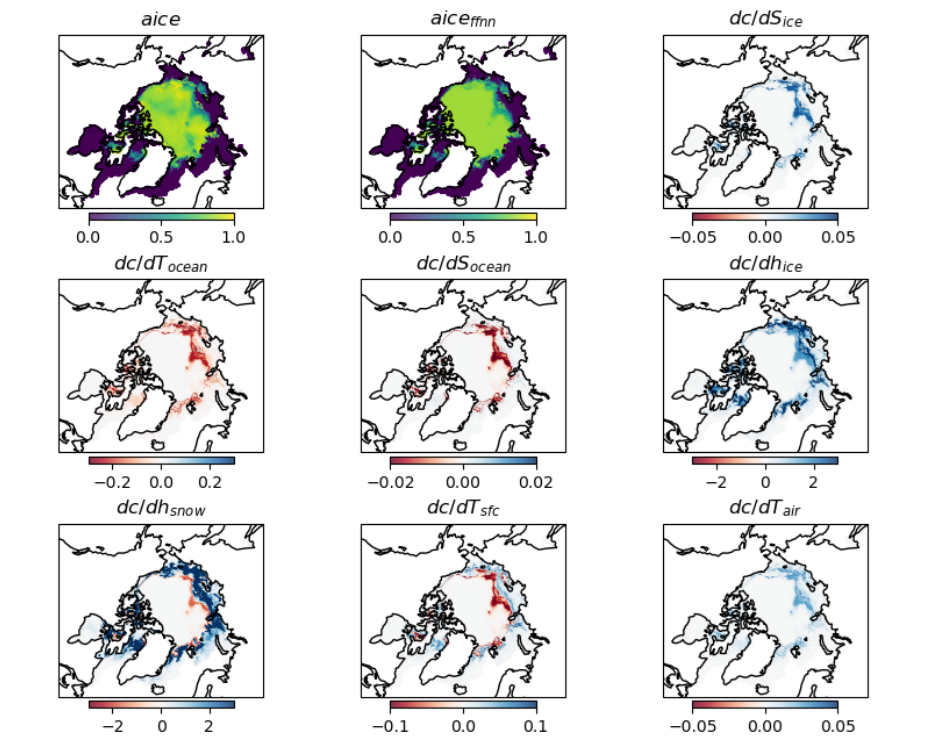
$$\delta c = \frac{\partial c}{\partial h_i}\delta h_i + \ldots$$

A **positive** ice thickness increment within the marginal ice zone results in a **positive** contribution to the sea ice concentration.

AI Balance Operators for Coupled DA

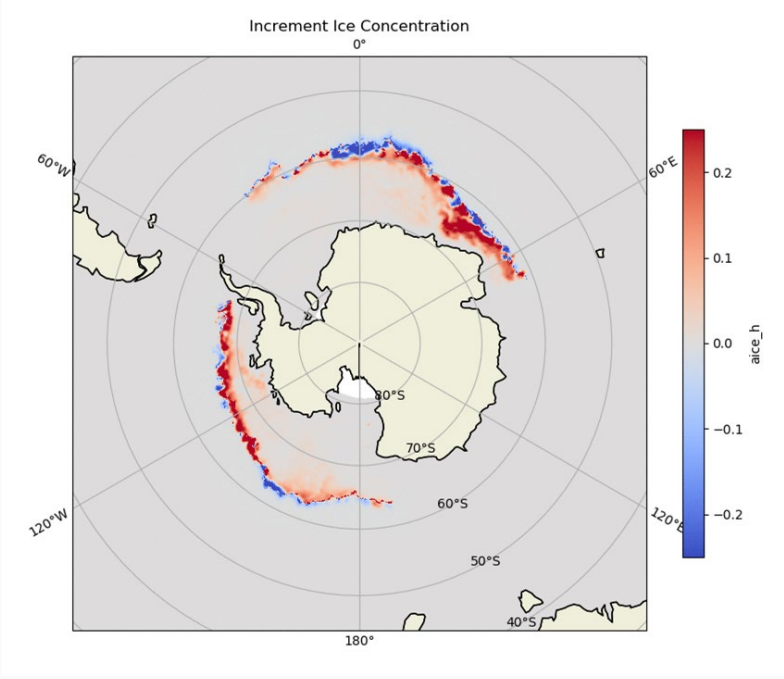
Sea Ice-Ocean Balance

Linearized a sea ice concentration emulator (TorchBalance).

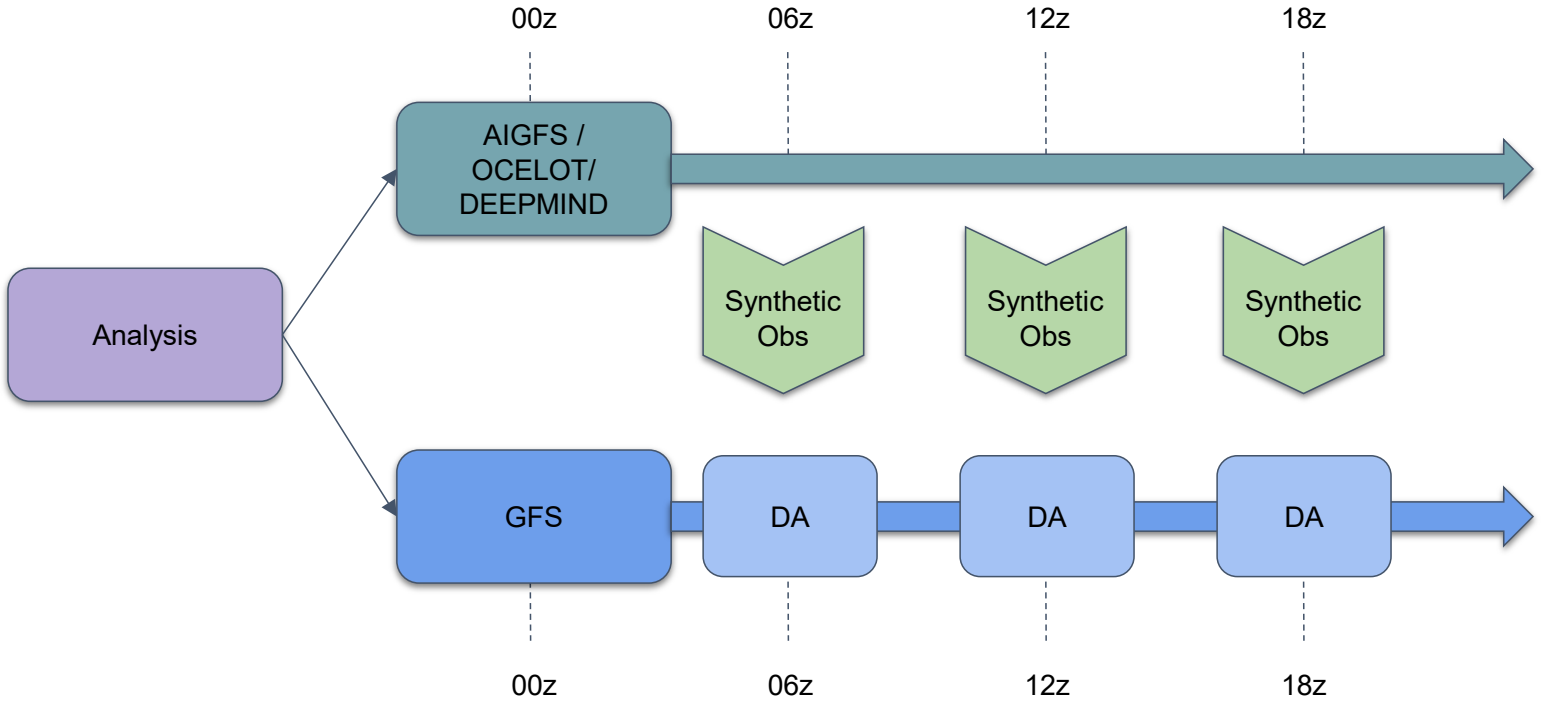


Analysis Projection

SST increment from assimilating ice concentration. Uses the Jacobian matrix (left) to project increments onto SST.



JEDI to Assimilate AI-generated observations

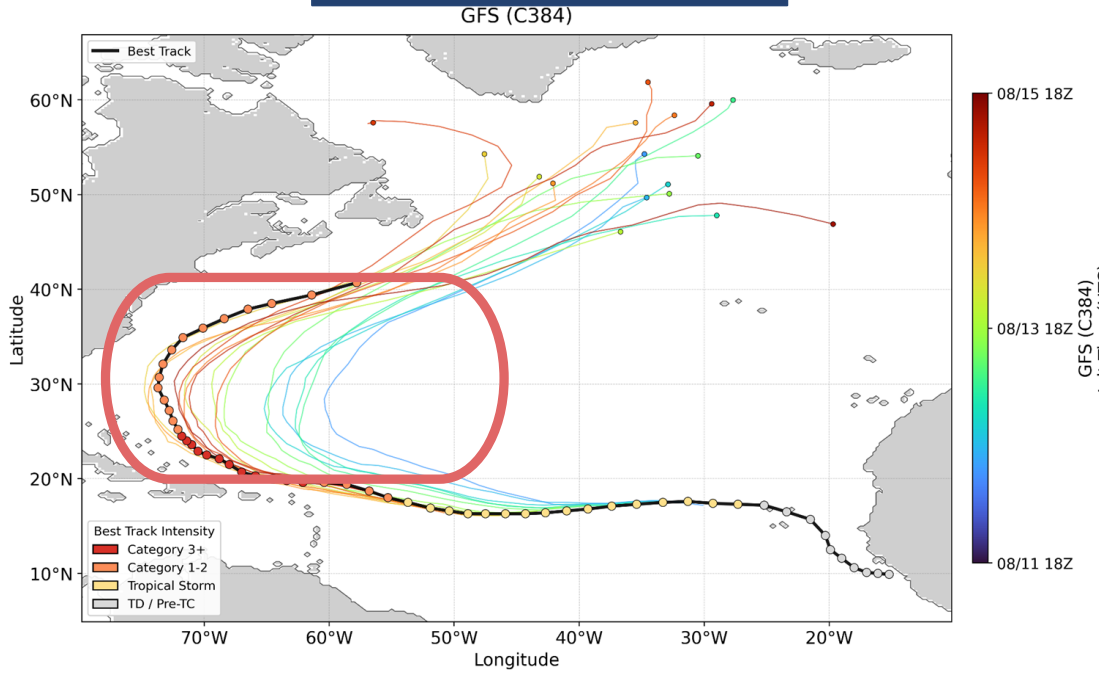


What's the idea?

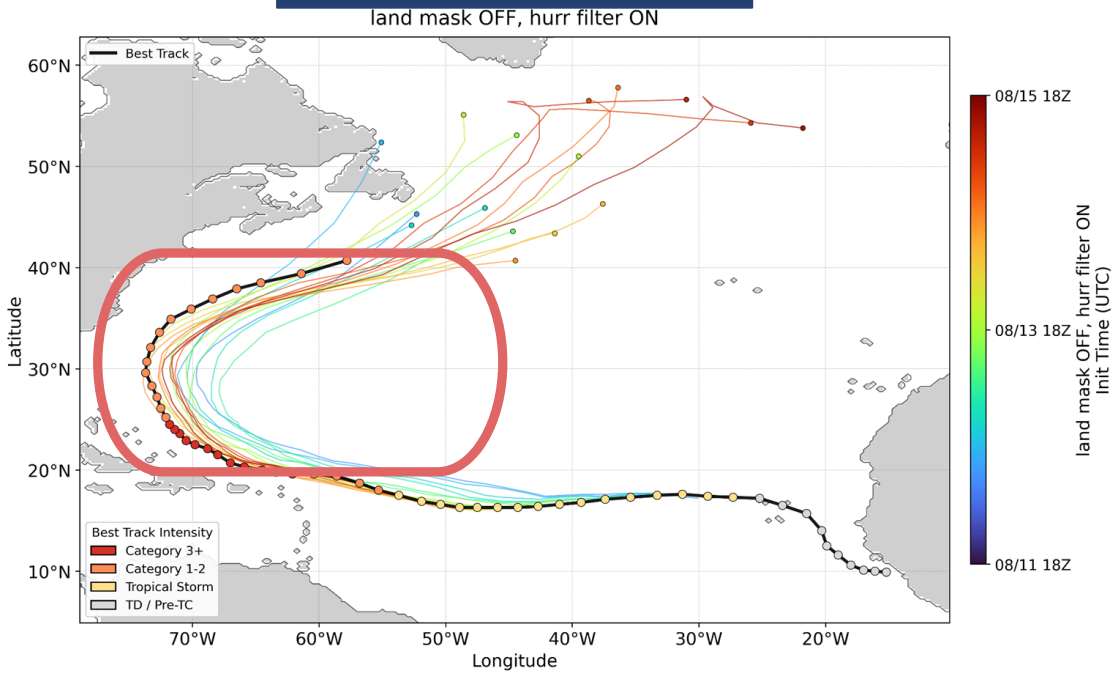
- AI models run faster than physical models and produce accurate large scales.
- AI models produce obs-space predictions (DOP, DeepMind TCvitals etc)
- JEDI runs in-core data assimilation.
- The data assimilation field knows something about producing balanced nudgings.
- Can DA be used to nudge to a combination of AI model output in a single operation?

Assimilate AIGFS and Deepmind

Baseline



DA Nudging



Assimilating synthetic surface pressure observations generated from AIGFS as well as TCVITALS from Deepmind corrects the rightward bias in the track for Hurricane Erin

Jacobians of AIML Models

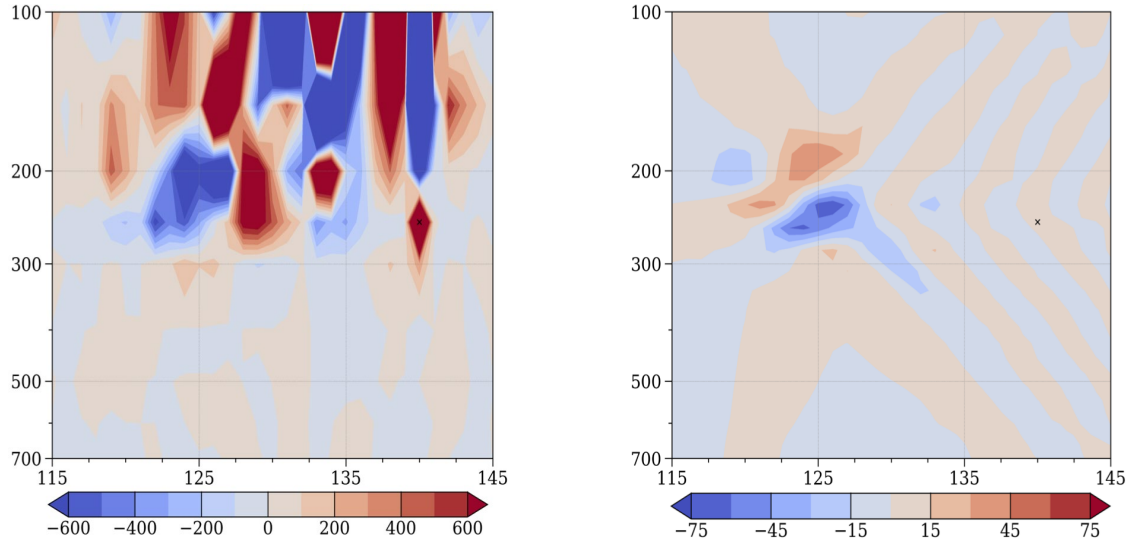
What's the idea?

- 4DVar, FSOI etc needs the model's TL and adjoint to be physically sensible.
- Ensemble DA needs a physically consistent perturbation response.
- Do these constraints hold for off-the-shelf AI/ML models?

Key Findings:

- **4DVar/FSOI:** corrupted gradients; analysis increments with unphysical structure far from the observation.
- **Ensemble DA:** consistent with Slivinski et al. (2025): ML models accumulate noise and diverge in cycled EnKF. TL/AD analysis offers a mechanism.
- Roll-out training does not help: the loss is in state space, not perturbation evolution.
- **Conclusion:** ML models may need better linearization properties before operational DA use.

AD sensitivity of T(×, 6 h) to initial q, section along 33°N
Colorbars: ±600 vs ±75.



GraphCast

MPAS Adjoint

Tian, Holdaway, & Kleist, 2026: Evaluating Machine Learning Weather Models for Data Assimilation: Fundamental Limitations in Tangent Linear and Adjoint Properties (10.1029/2025GL119402).

Fully Differentiable Physical Model

What's the idea?

- Machine Learning libraries such as PyTorch offer a plethora of highly advanced techniques for machine portability, GPU acceleration and code that is fully differentiable.

- We have an effort to port UFS (MPAS + Physics) to GPUs. Can we achieve fully a differentiable model?
  - Full model 4DVar/FSOI with physically consistent gradients.
  - Gradient limiters for pesky physics routines.
  - Long window 4DVar with nonlinear minimizers (e.g. re-analysis applications).
  - NeuralGCM-like approach to model improvement.

(See presentation by Xiaoxu Tian at the ECMWF HPC Workshop)

An initial port of MPAS to PyTorch has been completed. Work is now underway to port the physics package and establish the links to JEDI

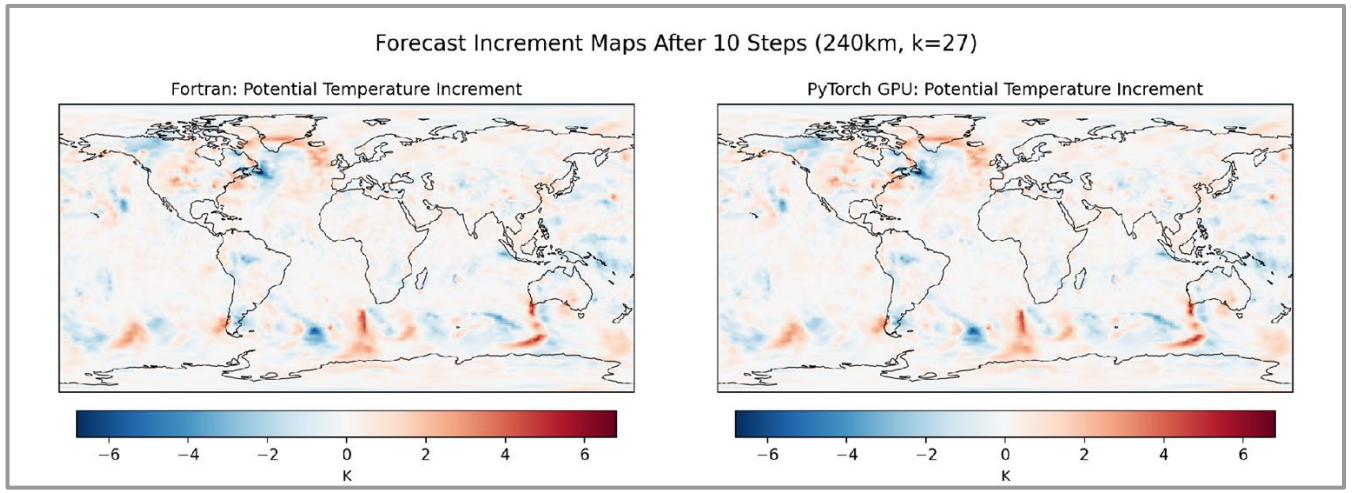
Concurrently working towards GPU models in JEDI as a subprocess and adding nonlinear minimizers using the JEDI building blocks.

Fully Differentiable Physical Model

Single GPU vs single CPU performance for MPAS dynamical core. Early results show promise.

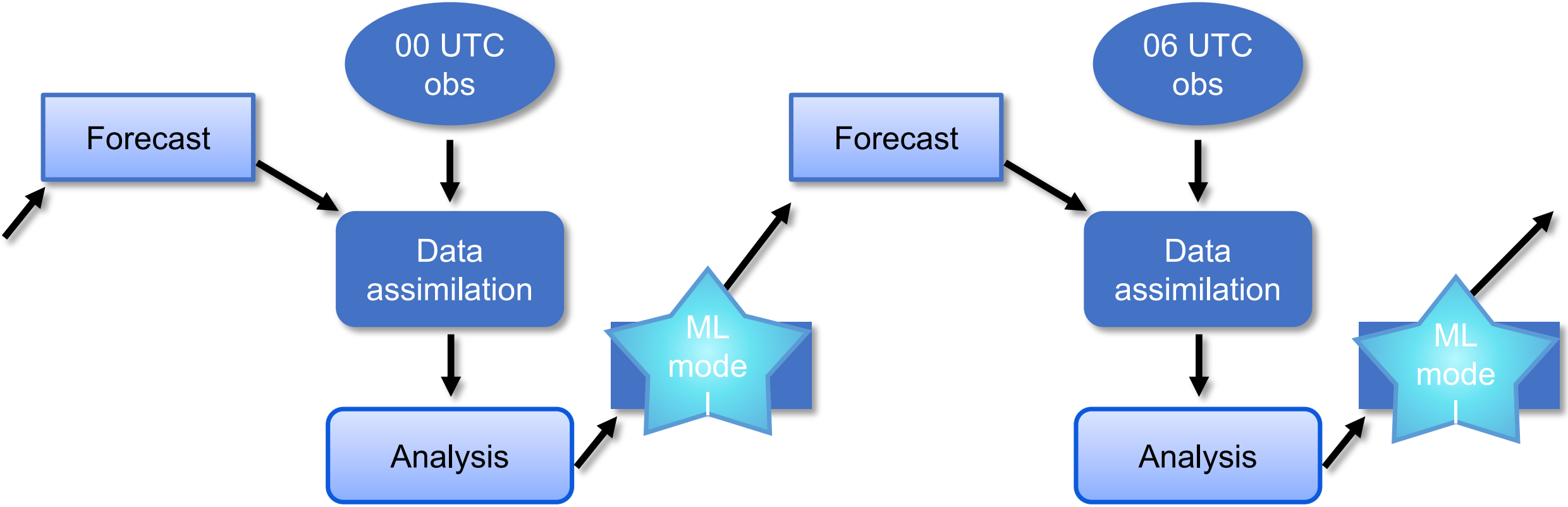
Mesh	Mode	Fortran CPU s/step	PyTorch A100 s/step	Speedup	Peak GPU (GiB)
120 km	**split**	3.793	0.459	**8.27x**	13.0
120 km	**coupled**	1.419	0.229	**6.20x**	12.7
60 km	**split**	16.389	1.738	**9.43x**	52.3
60 km	**coupled**	6.117	0.886	**6.91x**	50.1

Higher resolution will require cross-GPU node compute due to memory requirements. OMD, GSL and NVIDIA will work on this together.

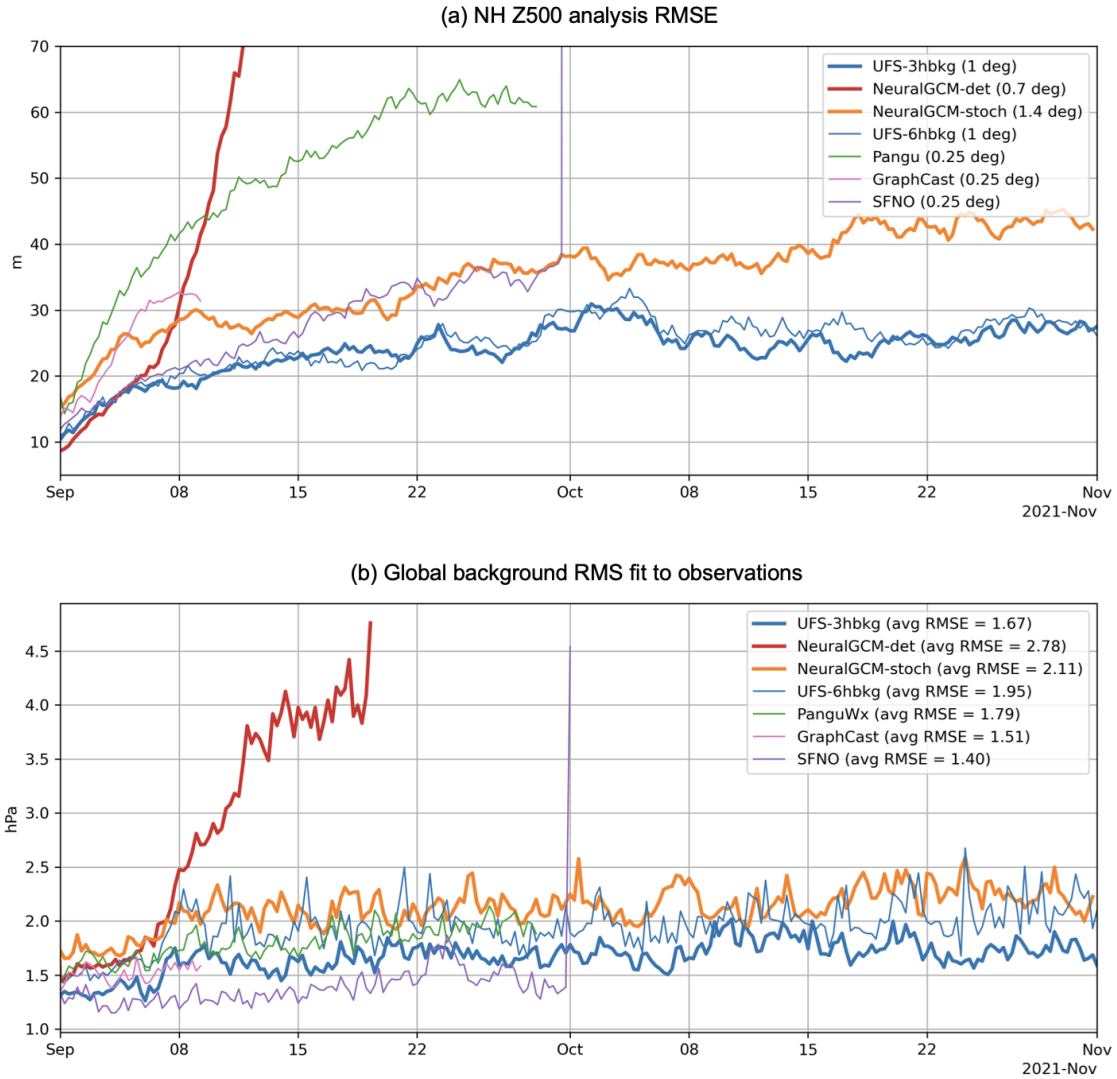


Can physical models be replaced by ML models in a cycling DA system for NWP?

Slivinski et al., 2025: Assimilating Observed Surface Pressure Into ML Weather Prediction Models (10.1029/2024GL114396).



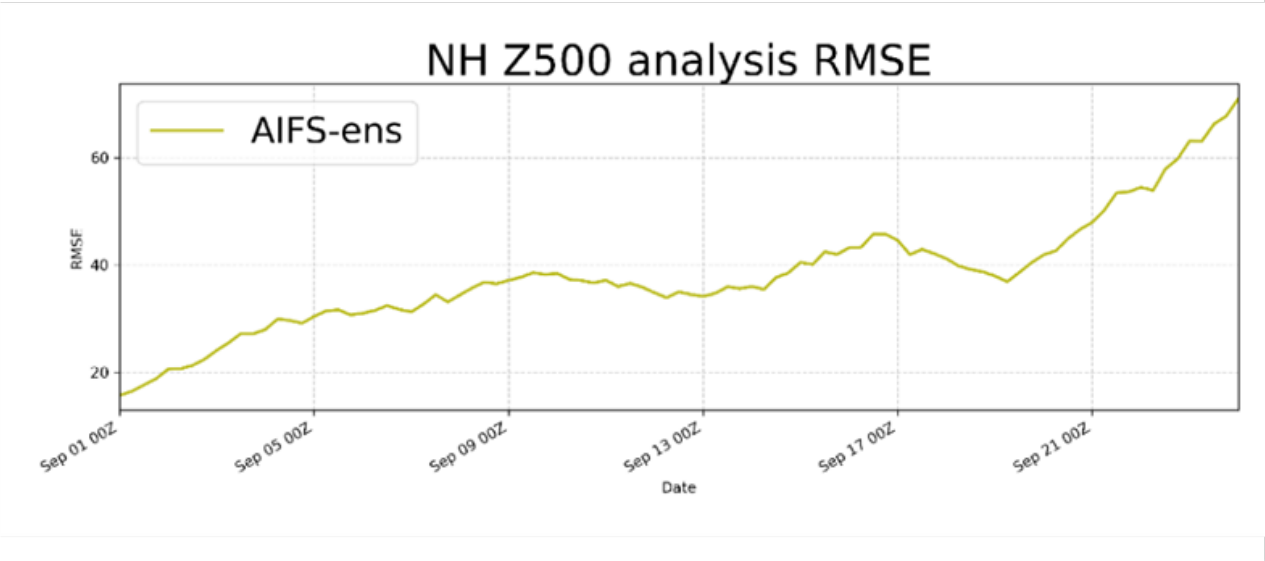
Can physical models be replaced by ML models in a cycling DA system for NWP?

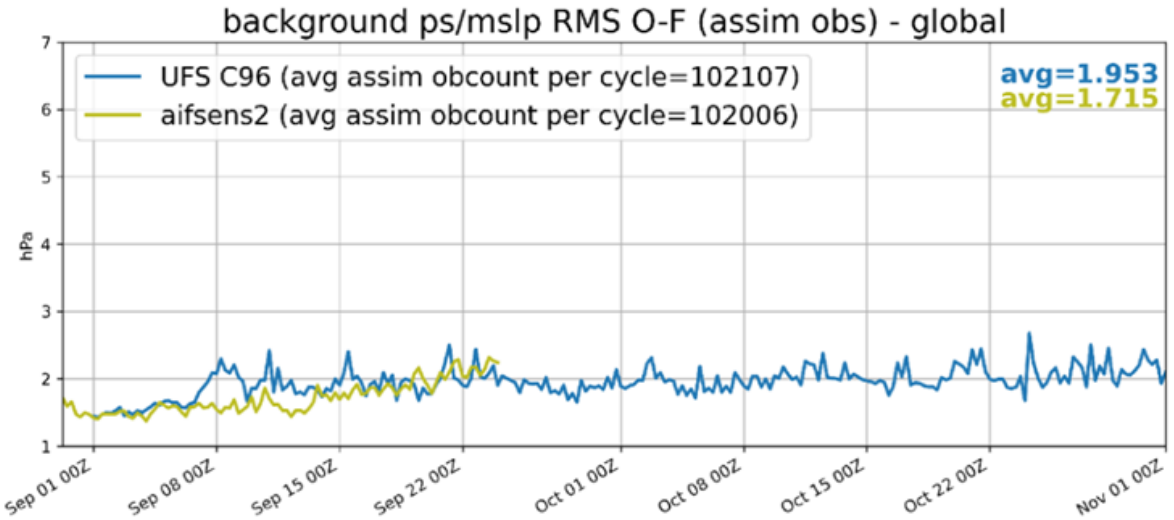


Unobserved fields become unphysical, leading to model crashes, in pure ML models (top panel) even while MSLP fields remain realistic at observation locations (lower panel).

Likely due to inaccurate representation of background error covariances from the ensemble (e.g. the models do not correctly represent cross-variable relationships).

Can physical models be replaced by ML models in a cycling DA system for NWP?





AIFS-ens does not crash during cycling, but performance of unobserved fields is poor (left panel).

Possibly due to the noise added in AIFS-Ens: may be destroying some cross-variable relationships present in the model (requires further investigation).

ML-based DA Example: HealDA

HealDA takes observations as input, and outputs an analysis field that can be used to initialize ML models such as Aurora, FourCastNetv3.

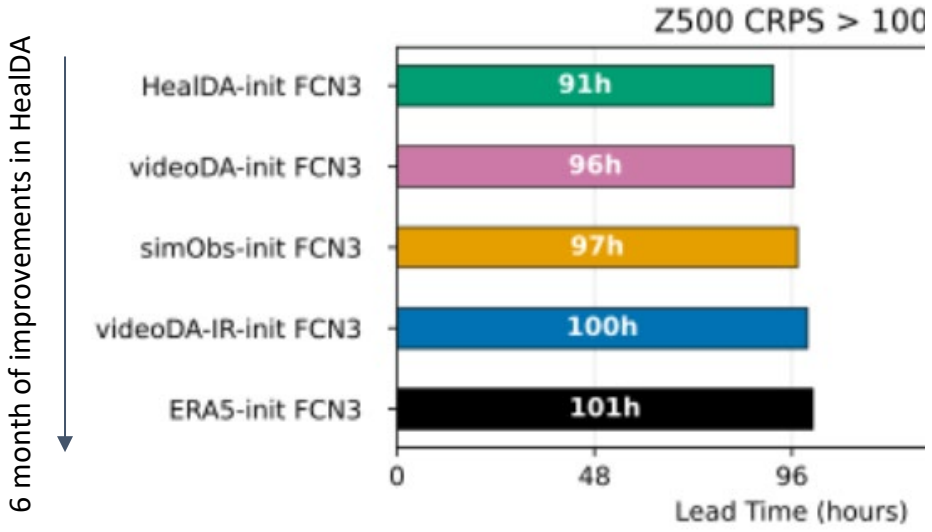


Image courtesy of A. Gupta (NVIDIA)

- HealDA was developed by NVIDIA in collaboration with NOAA and MITRE.
- HealDA skill is rapidly advancing and approaching the skill of ERA5
- Could we use the computational efficiency of HealDA to enable hourly update of the early delivery (AIGFS) forecast?

Gupta et al., 2026: HealDA: Highlighting the importance of initial errors in end-to-end AI weather forecasts (arxiv.org/abs/2601.17636).

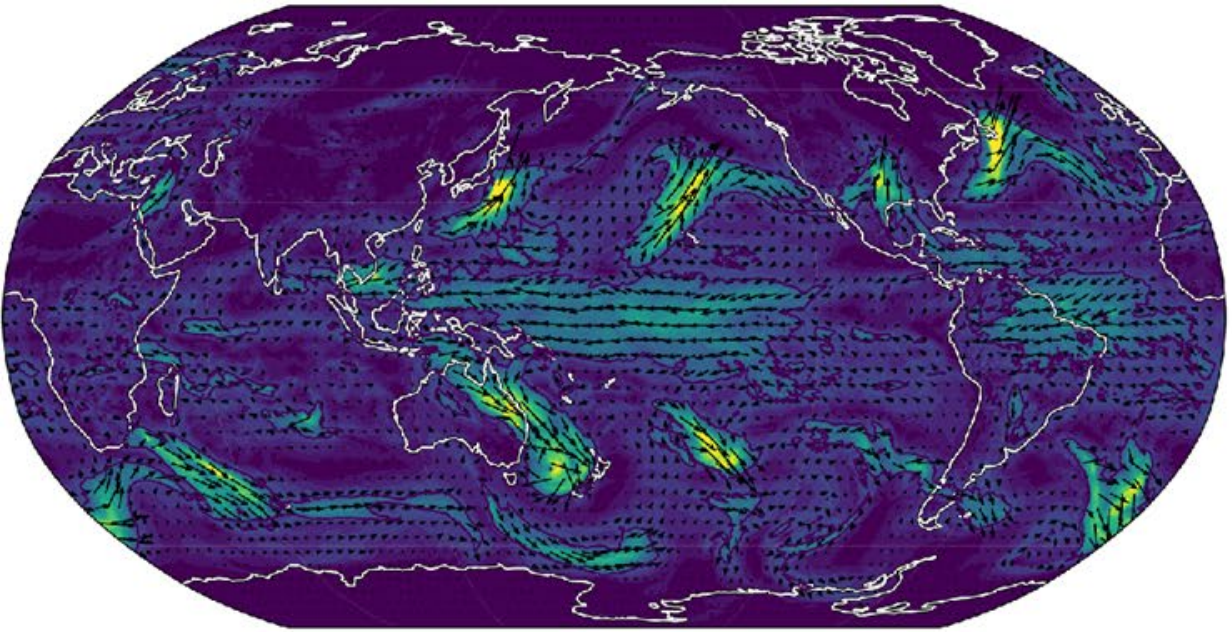
AIGFSv2dev (NOAA) vs. Aurora

	AIGFSv2	Aurora
Framework/architecture	Anemoi graph transformer	3D Swin Transformer + Perceiver encoder/decoder
Model size	~105.6M trainable parameters	~1.3B parameters
Horizontal grid	Global 0.25° (1440 x 721)	Global 0.25° (1440 x 720)
Pressure levels	1000, 925, 850, 700, 600, 500, 400, 300, 250, 200, 150, 100, 50; 13 levels	Same 13 pressure levels
3D prognostic variables	u, v, t, q, gh (geopotential height)	u, v, t, q, z (geopotential)
Surface prognostic variables	u10m, v10m, t2m, d2m, skt, msl, sp	u10m, v10m, t2m, msl
Training data	Pretrained: ERA5 1979-2024 Fine tuned: GDAS Apr 2021-Mar 2025	Pretrained: ERA5 1979-2020 HRES 0.25° fcst 2016-2020 GFS 0.25° anal/fcst 2015-2020 CMCC/ECMWF climate simulation 1950-2014 Fine tuned: IFS HRES anal 2016-2021

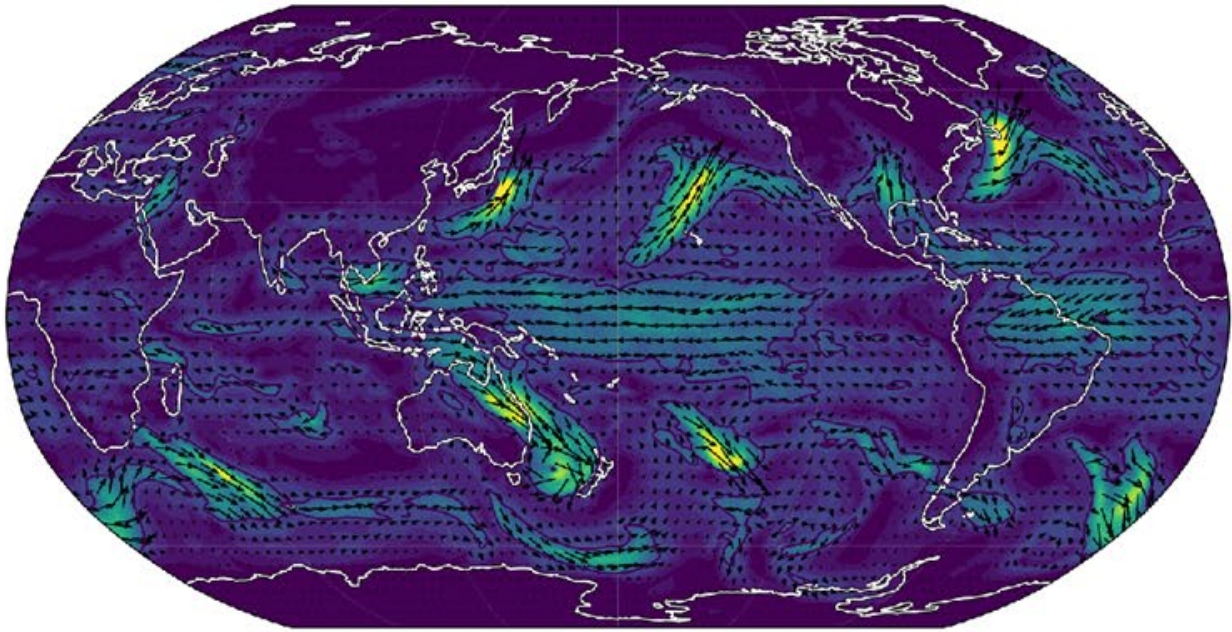
5-Day IVT Forecasts, init: 00Z Apr 2, 2025

Integrated Vapor Transport
Valid: 2025-04-02T00 UTC Forecast hour: +000 h

GDAS / AIGFSv2dev Forecast



HealDA-v2 / Aurora Forecast



Thanks to Aayush for assistance setting up HealDAv2 and Earth2Studio



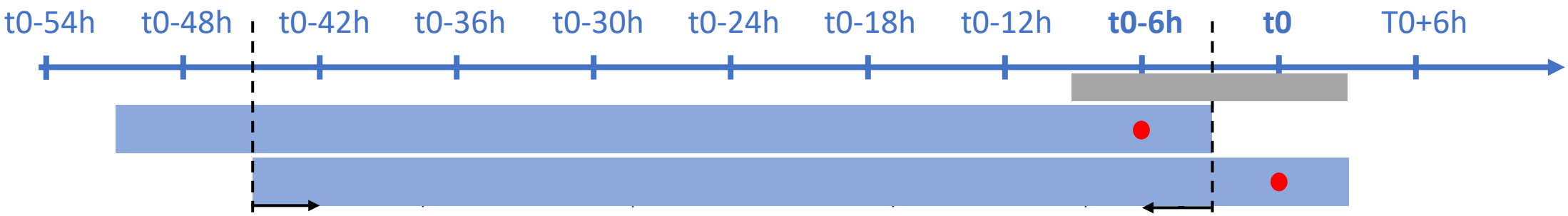
Forecast Performance against ERA5 Analysis

	Experiment	Initialization	Observation Coverage	Forecast Model
v	GFS	GFS analysis (@t0 and t0-6h)	t0-9h to t0+3h (early obs cutoff)	GFS (FV3 dynamic core)
v	GDAS + AIGFSv2dev	GDAS analysis (@t0 and t0-6h)	t0-9h to t0+3h (late cutoff)	AIGFSv2 dev
v	GDAS + Aurora	GDAS analysis (@t0 and t0-6h)	t0-9h to t0+3h (late cutoff)	Aurora v1
v	HealDA + Aurora	HealDA analysis, obs only (@t0 and t0-6h)	t0-51h to t0+3h (repeated use of 42h obs in-between?)	Aurora v1

NWP vs. ML

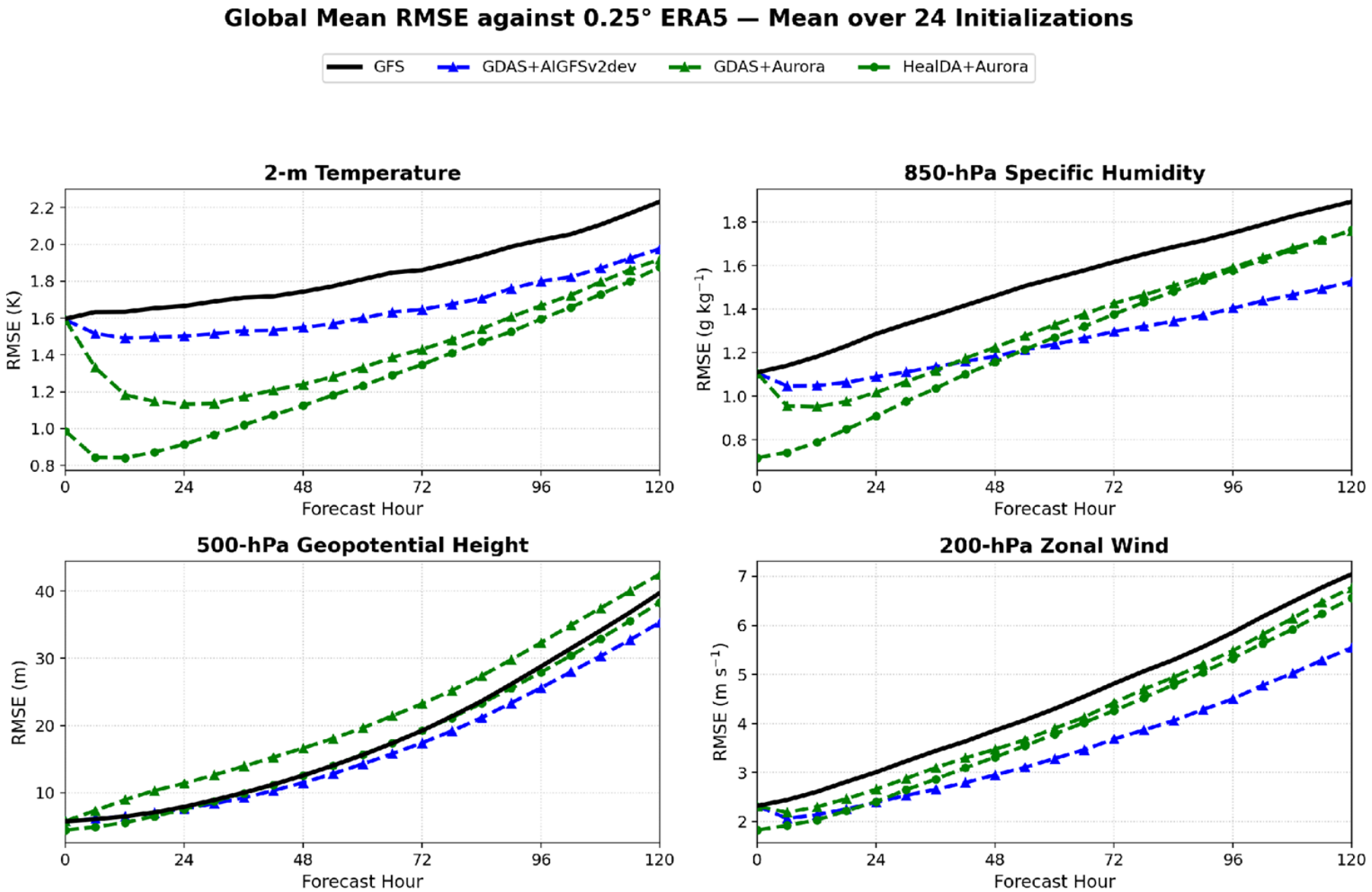
AIGFS vs. Aurora (Different MLs)

GDAS vs. HealDA (Different ICs)



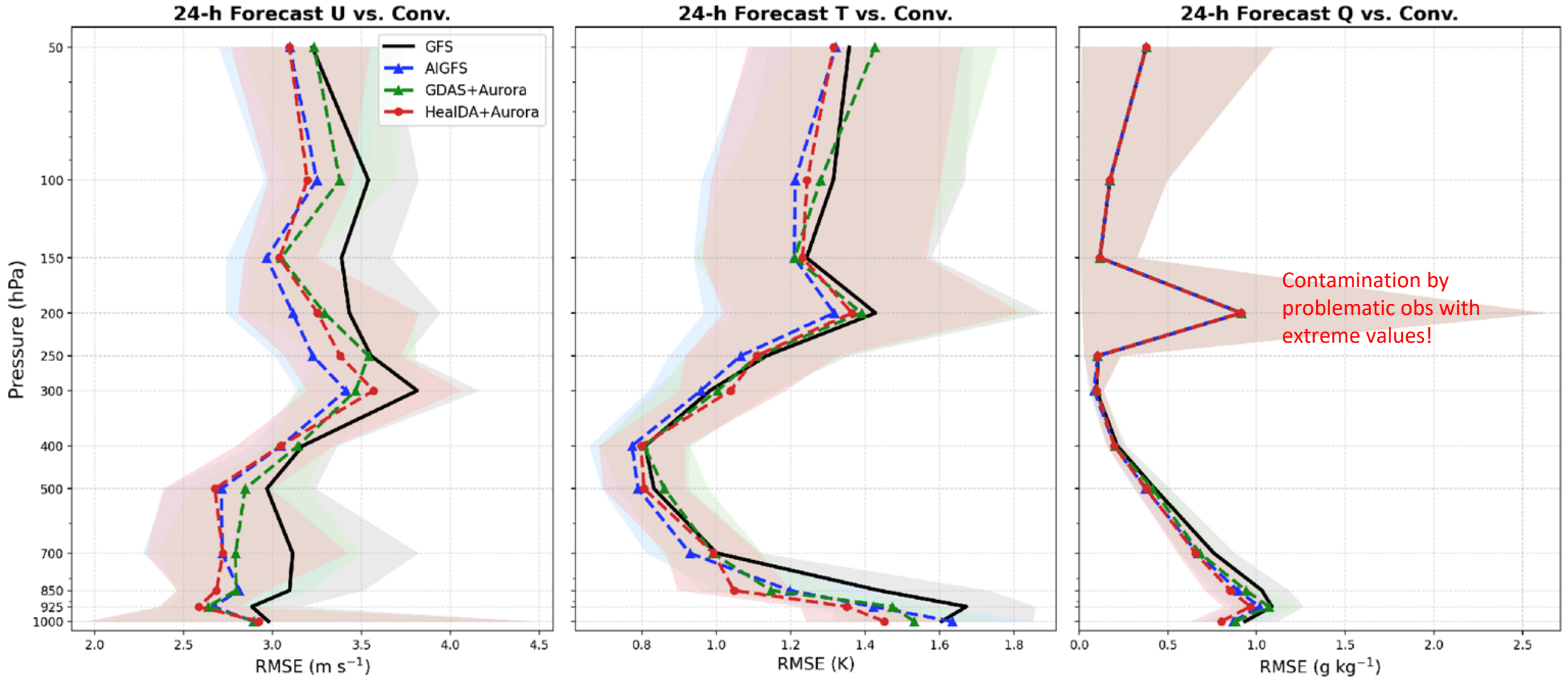
Forecast Performance against ERA5 Analysis

IC samples: from 00Z April 2 to 18Z April 30, 2025, in every 30 hrs



Forecast Performance against Observations

(±1h obs matching window)



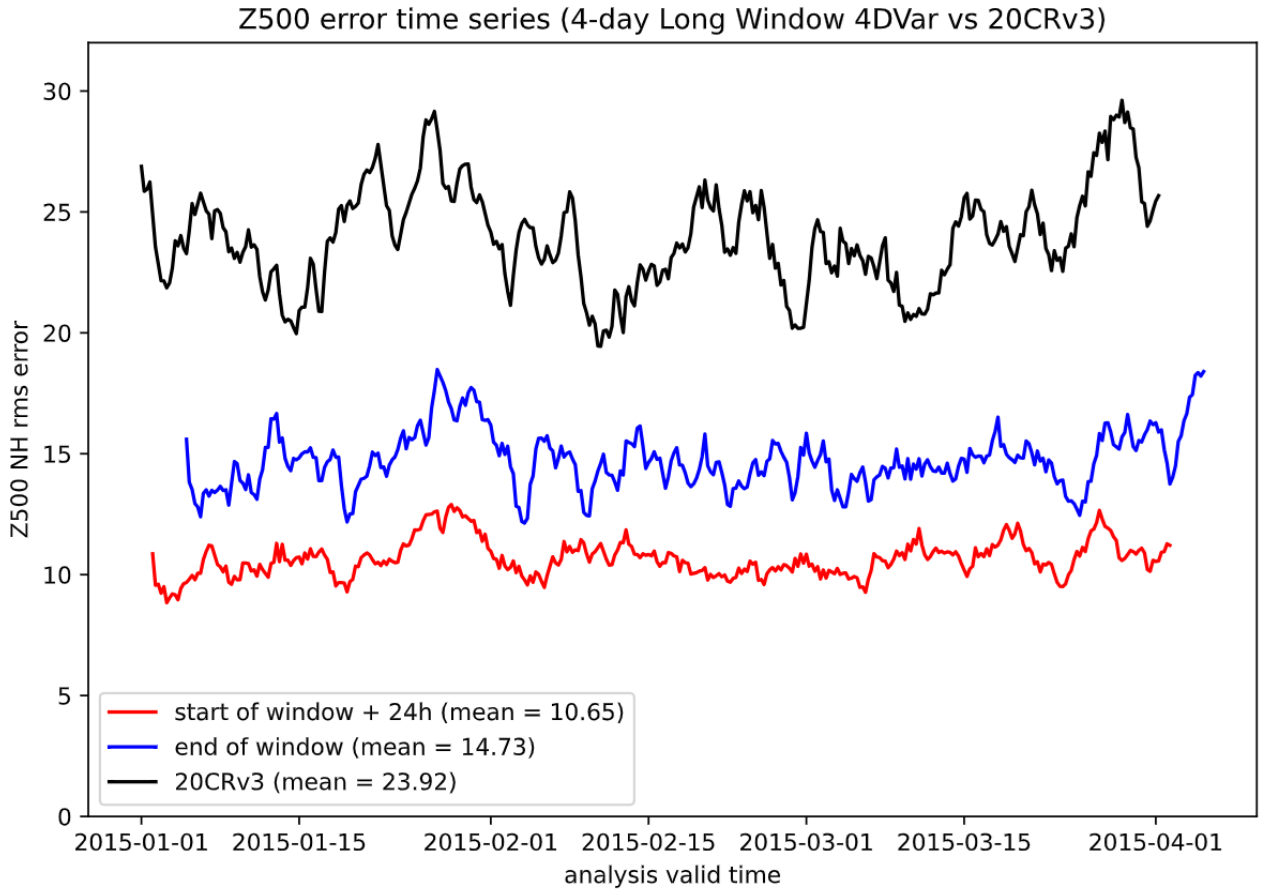
Mean profiles from available forecast initializations; shaded bands denote 95% bootstrap confidence intervals.
Specific humidity RMSE is shown in g kg^{-1}.

Long-Window 4DVar with NeuralGCM

Hakim et al., 2026: Long-window 4DVar for reanalysis using a differentiable weather model (arxiv.org/abs/2608.11515).

What's the idea?

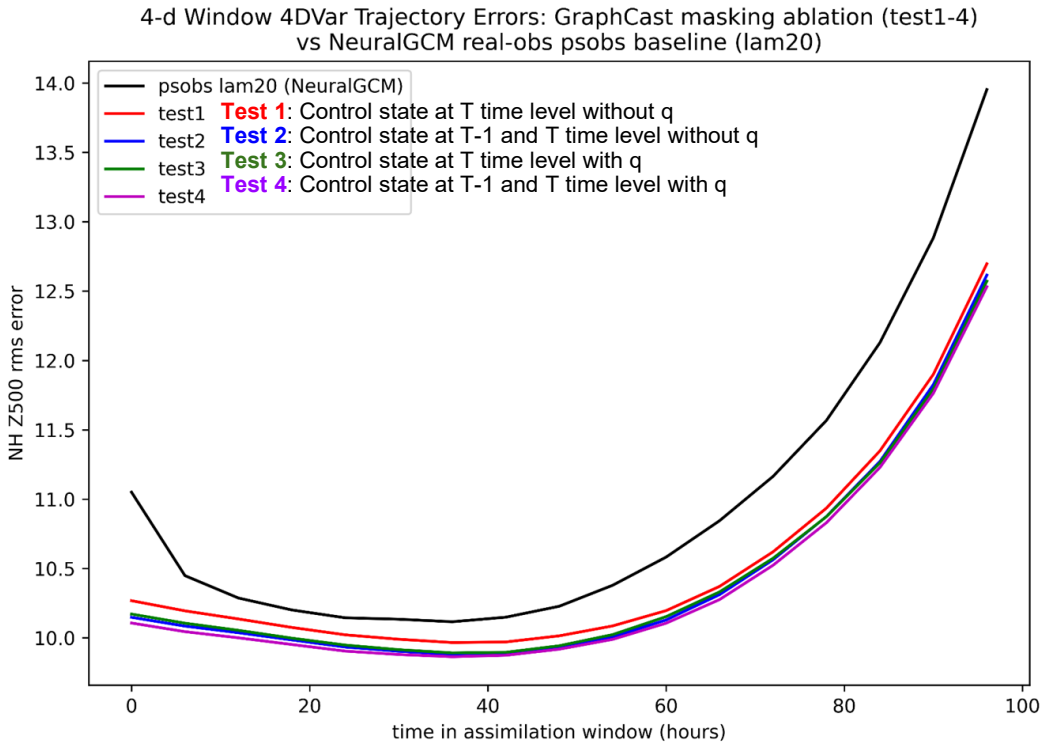
- Machine Learning models are fast and fully differentiable.
- This can enable fast long-window 4DVar with fully nonlinear minimization techniques. Potential reanalysis application to enable better analysis in sparse observation eras.



- Stable cycling for 90+ days.
- Errors for smoothing estimate (red) less than 50% EnKF based 20CRv3.
- Errors for filtering estimate (end of window, blue) still ~38% less.

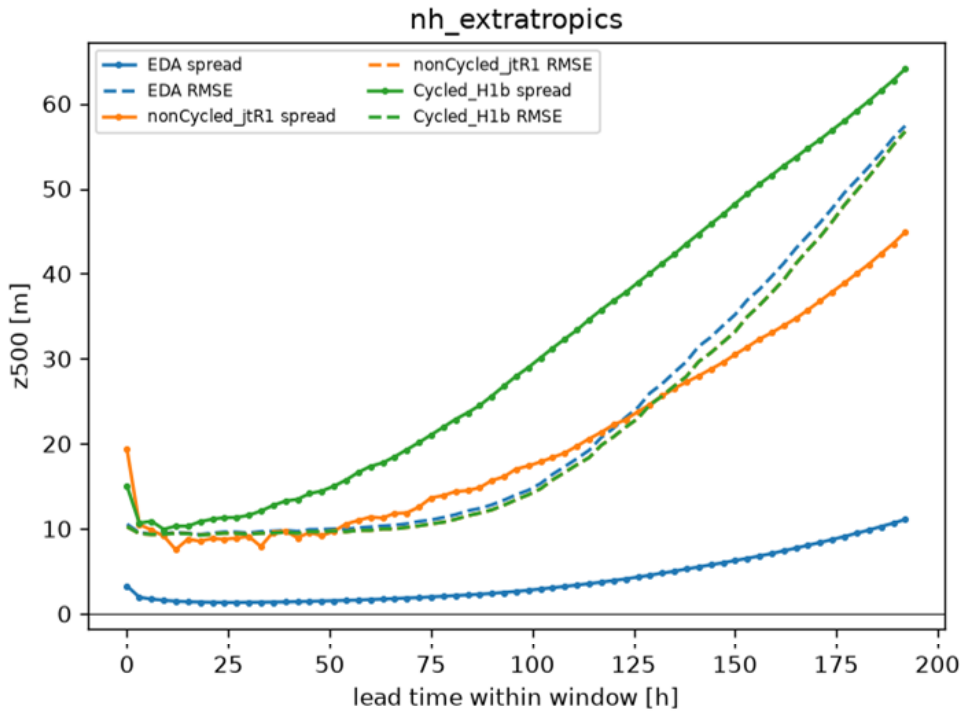
Orbit: Sun-synchronous / 27-day

Switch from 3-degree NeuralGCM to ¼ GraphCast



- Currently in the process of switching from a 3-degree NeuralGCM model to ¼ degree GraphCast and AIFS.
- Early results suggests that we can improve the quality of the analysis by switching to higher-resolution Ai-based model.
- Currently working on the stability of the cycling system.

Extending deterministic 4DVAR to ensemble framework



- Can we extend traditional deterministic 4DVAR by searching for optimal perturbations that satisfy afCRPS cost function?
- Early results suggest:
  - Traditional EDA can be under dispersive.
  - 4DCRPS (centered on 4DVAR control) can produce well calibrated results.
  - Cycling 4DCRPS is proving a bit more difficult (possibly due to sparse observations here).

OCELOT

Gholoubi et al., 2026: OCELOT: Direct Atmospheric Forecasting from Heterogeneous Earth Observations Using a Graph–Transformer Hybrid Model (arxiv.org/pdf/2607.14196).



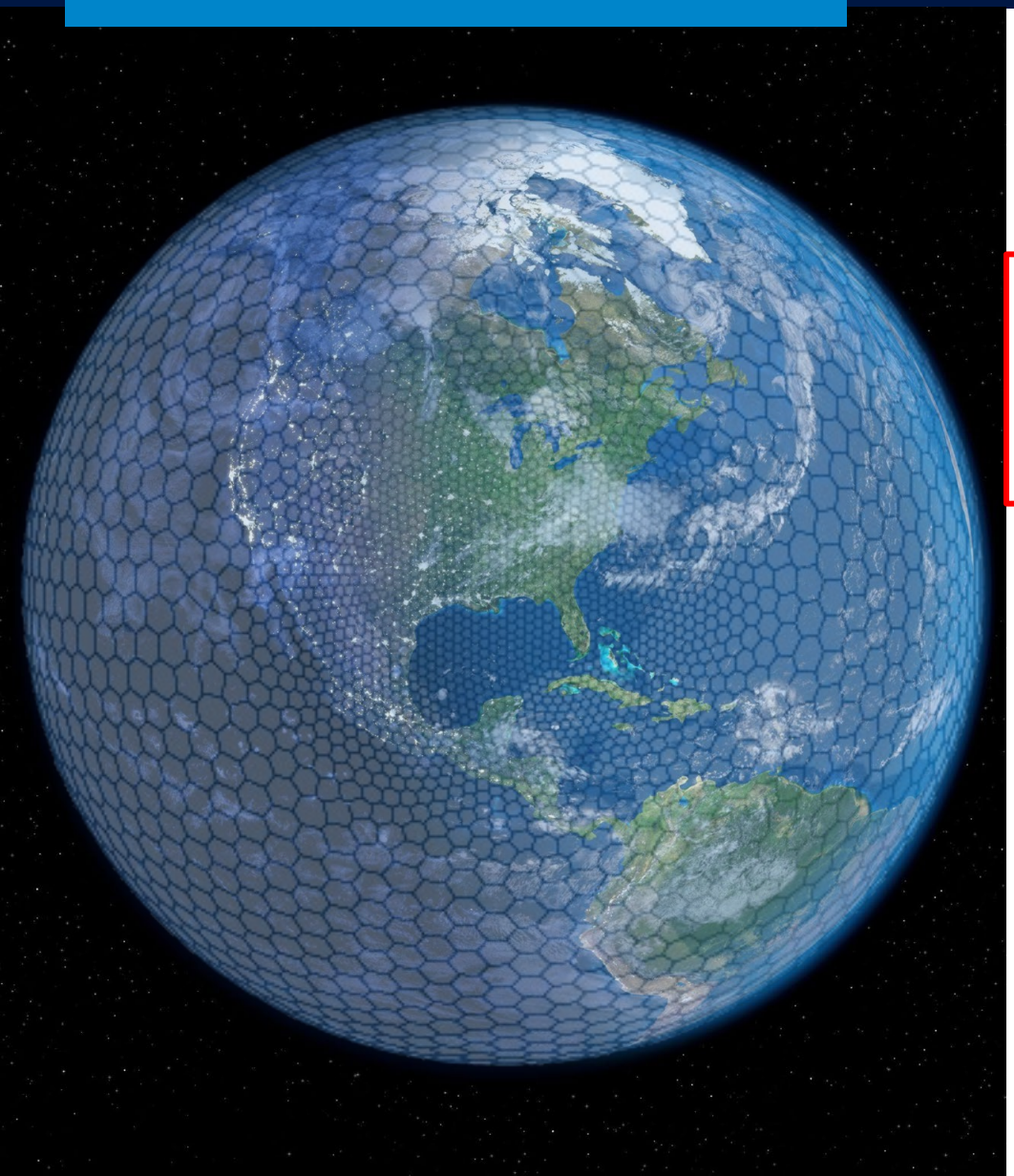
Inspired by GraphDOP (ECMWF):

- Showed it is possible to generate skilful weather forecasts using only Earth observations — no physics-based background state or (re)analysis.
- Introduced a latent-space that forecasts in observation space, dynamically constructing graphs from satellite and conventional data.
- **OCELOT builds on this concept with NOAA datasets + modular GNN**

Concluding Remarks

- ML proving to be beneficial within construct of “traditional / classical” DA: efficiency, better use of observations, background error, etc.
  - Not shown: additional examples for relatively simple quality control, bias correction, etc.
- Examples of things not working as expected (at least initially): cycling with AIWP models, TL/AD of AIWP models, etc.
- Transformative DA paradigms (or replacements) are rapidly accelerating: where is this all heading and where do we best allocate (finite) resources?

NOAA NWP Moonshot



Goal 1. A global, fully-coupled and convection-allowing Earth system prediction system based on the MPAS dynamical core.

Goal 2. Hybrid ensemble forecasts combining physical and data-driven models, underpinned by high resolution Earth system observations, reanalyses, and continuous data assimilation.

Goal 3. Modernize the NWS modeling infrastructure and governance to optimize forecast accuracy and efficiency in the operational production suite.

Goal 4. Seamless creation of probabilistic forecast guidance and integration with tools and cloud-based services to increase access, reliability and trustworthiness of tailored data to meet forecaster & societal demands.