

# The future of Data Assimilation in the Era of the AI Revolution

Stephen G. Penny - Head of Weather, Sofar Ocean

SOFAR OCEAN

ECMWF Annual Seminar
18 September 2026

# ECMWF-ESA ML workshop, April 2026, Bologna




## Data Assimilation in the era of the AI revolution - where we stand today

# Outline

Part 1: The Catalyst

Part 2: Reinventing Atmospheric Physics

Part 3: A Grand Unified Theory for DA

Part 4: The Data Assimilation Scientist of the Future

# Part 1:
## The Catalyst

Computing advancements

Modeling advancements

# Computational revolution - “The tail wags the dog”



**1946**
ENIAC

**Charney et al. 1-layer BVE**, 500 hPa, 700 km grid, 270 points, finite-difference

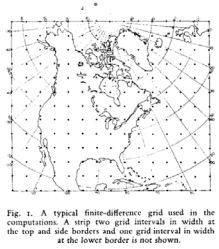


Fig. 1. A typical finite-difference grid used in the computations. A strip two grid intervals in width at the top and side borders and one grid interval in width at the lower border is not shown.



**1976**
Cray 1

**Vector processing** led to transition to spectral models to leverage matrix-vector operations. Global PE models, 10+ layers, physics parameterizations.



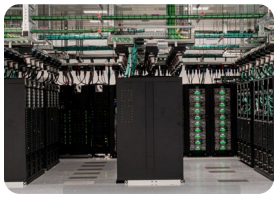
**1991**
Cray Y-MP

Transition to hybrid vector / multi-processor (MPI) -> led to a **complete rewrite** of many NWP codes.



**2002**
IBM p690

Transition to full consumer **multi - CPU clusters**. Commercial off-the-shelf systems began outperforming custom designs.



**2022**
BullSequana

**Mixed CPU/GPU** environments to support both legacy NWP models and emerging AI/ML architectures.

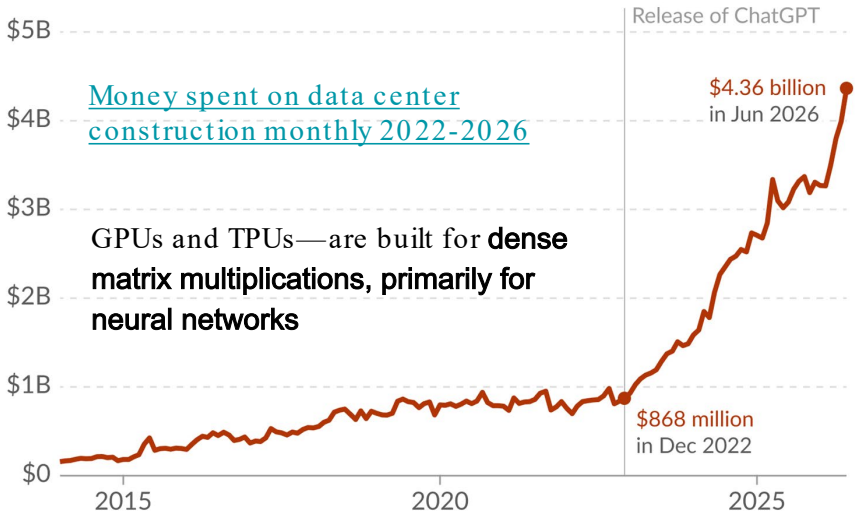
# Part 1:
## The Catalyst

# Computational revolution - “The tail wags the dog”

### Money spent on building data centers in the US has grown 5-fold since late 2022

Our World in Data

Cost of privately-funded construction of data centers each month, including materials and labor. It excludes the cost of IT hardware like servers and storage, which can account for a large share of total investment. This data is expressed in US dollars, adjusted for inflation.




**Note:** This data is expressed in constant 2021 US$. CC BY

**Data source:** U.S. Census Bureau (2026); U.S. Bureau of Labor Statistics (2026)



Google central Ohio data center

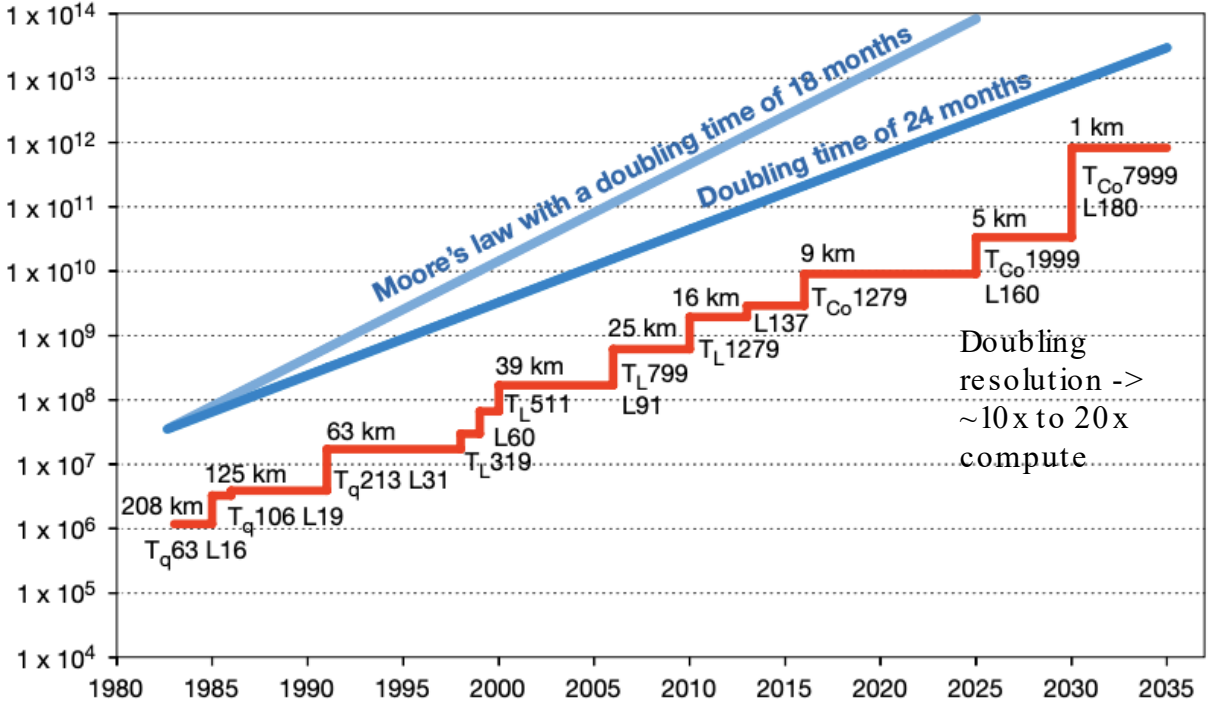


Microsoft Wisconsin data center (2026)

Part 1:
The Catalyst

# Cost versus model resolution / accuracy

We may not be able to count on the continued pace of CPU improvement

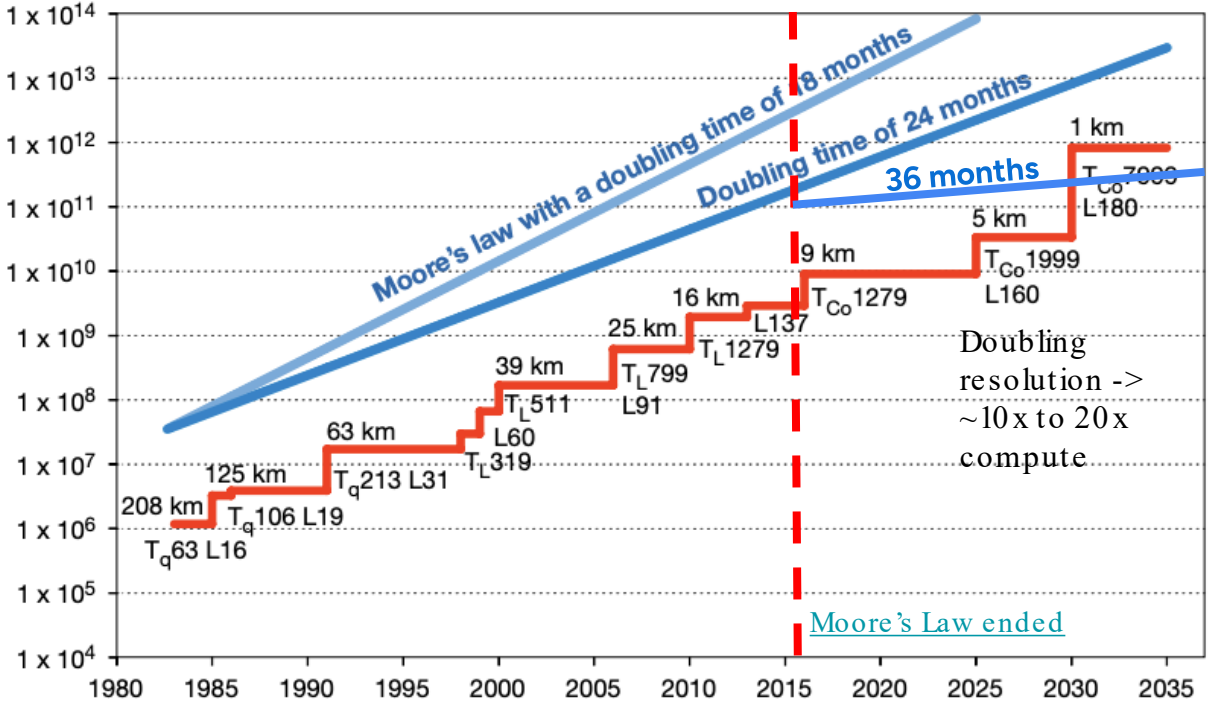



50 Years of ECMWF

Part 1:
The Catalyst

# Cost versus model resolution / accuracy

We may not be able to count on the continued pace of CPU improvement

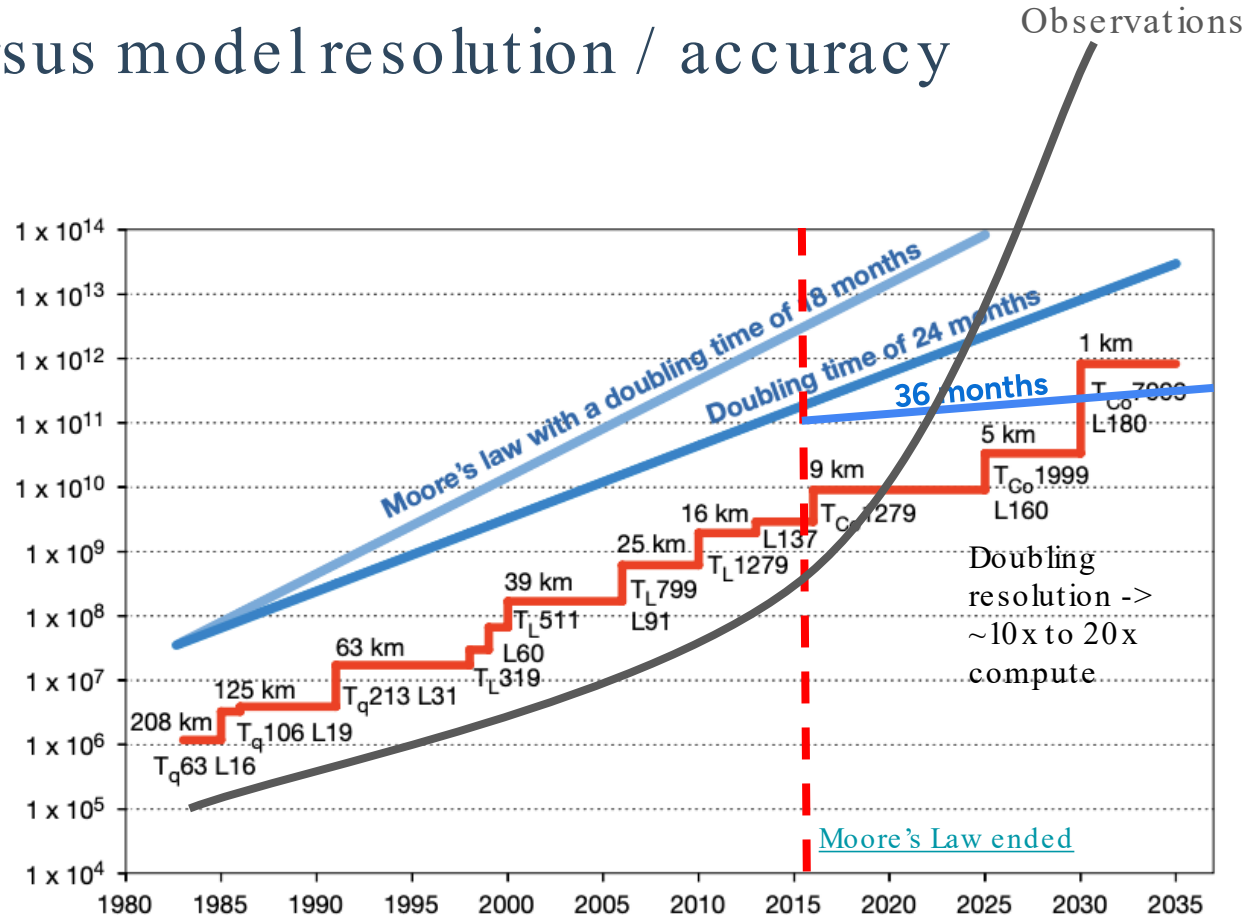



50 Years of ECMWF

Part 1:
The Catalyst

# Cost versus model resolution / accuracy

We may not be able to count on the continued pace of CPU improvement




50 Years of ECMWF

Part 1:
The Catalyst

# Traditional DA was already hitting walls

### The Memory Wall

Storing and moving state vectors gets progressively harder with high-resolution models.

The Bottleneck: Compute is cheap, but memory and network communication limit real performance.

### Increasing challenge: Utilize full volume of streaming observations

$35+ Billion spent on active satellite observing system.

Data Thinning: Traditional DA throws most of this rich data away because:
(a) physics-based models cannot resolve it, and
(b) DA methods cannot extract enough added value from them to warrant the cost.

### A missed opportunity: DA "Amnesia"

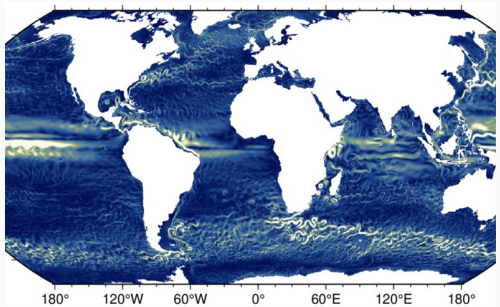
Traditional DA generally operates in the present—for today's current state.

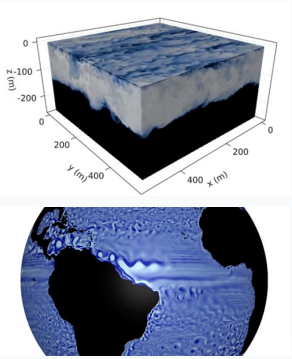
Lost Value: The DA systems have limited ability to extract patterns from past decades of historical observations, simulations and reanalysis.

# How is traditional modelling keeping up with the hardware?

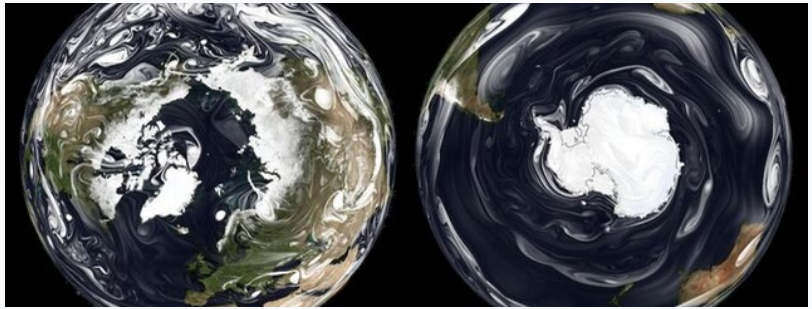
Modernized python/julia/etc. refactors and completely new builds:

## Oceananigans / CliMA -Ocean ↗

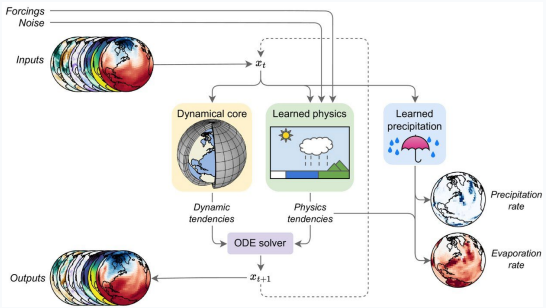


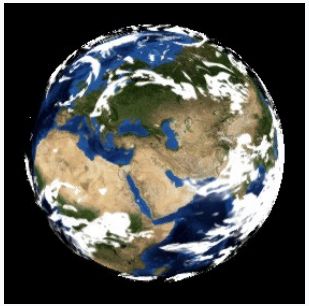

## SpeedyWeather.jl ↗



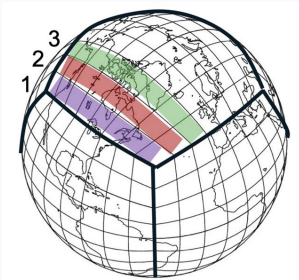
## Dinosaur / NeuralGCM ↗




## CliMA -Atmos ↗



## Pace (Fv3) ↗



Part 1:
The Catalyst

# Summary so far



## GPU Transition

The transition to GPU-dominated compute is inevitable for next-generation weather prediction systems.



## Historical Adaptation to hardware

NWP models have historically adapted to available computer architectures and hardware paradigms.



## Scaling Limits

Modeling and Data Assimilation are hitting physical scaling walls as model grid resolutions and observing networks expand.

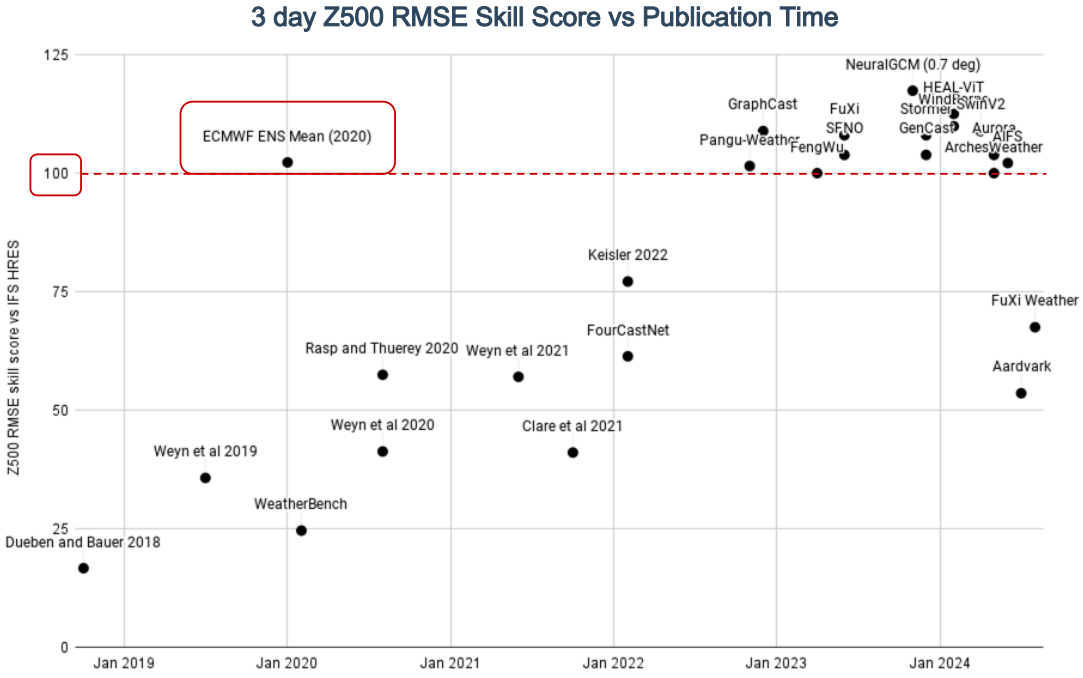
# Part 1:
## The Catalyst

# MLWP rapid development over recent years

→ In the last 6-7 years, MLWP models have advanced rapidly

→ In the last 2-3 years, they seem to have plateaued

→ These models all depend on NWP inputs

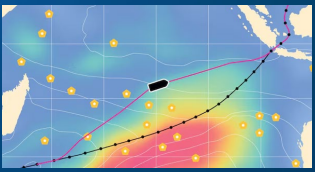



https://raspstephan.github.io/blog/ai-weather-progress/# Last updated November 2024

# Part 1:
## The Catalyst



Sofar Global Spotter Buoy Network

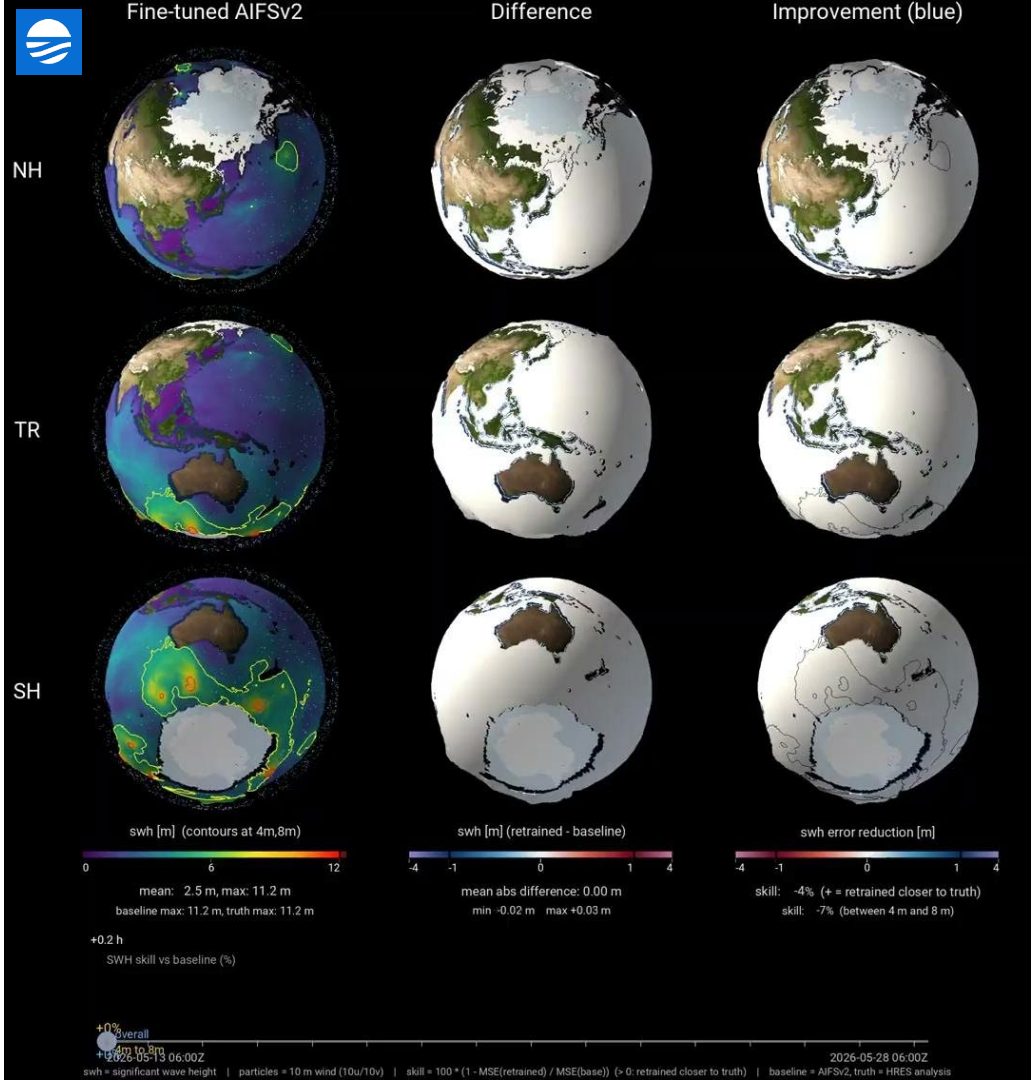


Sofar Wayfinder Route Optimization

# Fine-tuning AR MLWP models

We have fine-tuned AIFSv2 to improve 15-day wave height forecasts **up to 50% in the critical 4m -8m** decision range for ship routing

But it still requires conventional NWP generated initial conditions (ICs).



Part 1:
The Catalyst

# Observation-only ML modeling: a useful toolset

**Empirical nonlinear autoregressive spatiotemporal interpolation** :

- **Empirical:** Relies on observed historical data and statistical regularities rather than explicit physical equations or pre-defined probability distributions
- **Nonlinear:** Uses flexible architectures (like deep neural networks) to map complex, non-linear atmospheric interactions (like fronts or multi-sensor dependencies) that a linear model would smooth away.
- **Autoregressive:** Uses past states of the observations to predict or condition the estimation of future states.
- **Spatiotemporal:** It simultaneously ingests and maps data distributed across both three-dimensional space and continuous or stepped time.
- **Interpolation:** The ultimate functional goal of the entire system is to take sparse, irregular observations and fill the gaps—both across the spatial grid and forward into the temporal window.

Potential: Infrastructure for rapid, unique, and customizable forecasts in well observed regions.

Danger: Cannot resolve unobserved areas/variables - leading to hallucinations that may seem realistic.

Part 1:
The Catalyst

# Need for a Better Approach

## Current MLWP Status Quo

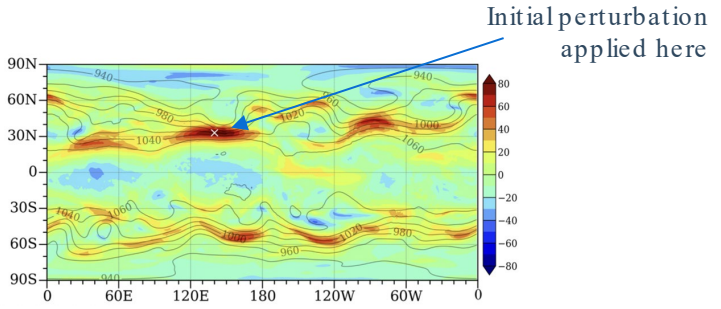
- Postprocessing Role: Current MLWP models act primarily as "autoregressive (AR) postprocessing" of NWP analyses.
- Limited Scope: Good RMS performance when forecasting individual, isolated fields, but no guaranteed consistency between them.
- Loss of Memory: ML models quickly "forget" initial conditions (ICs) and drift back to the model's training distribution.

## Core Requirements for Modern DA

- Dynamical Integrity: Must capture flow dependence, accurate perturbation responses, attractor convergence, and instability-driven error growth.
- Physical Integrity: Must explicitly incorporate physical laws and conserved quantities.
- Statistical Integrity: Must adhere to Bayes Theorem, with well estimated priors and posteriors, and robust error estimation.
- -> Structure Functions: Produce state-dependent structure functions that accurately transfer observation information across spatiotemporal domains.

# Part 1:
## The Catalyst

# Why AR MLWP Models Fail at DA: The Illusion of Physics

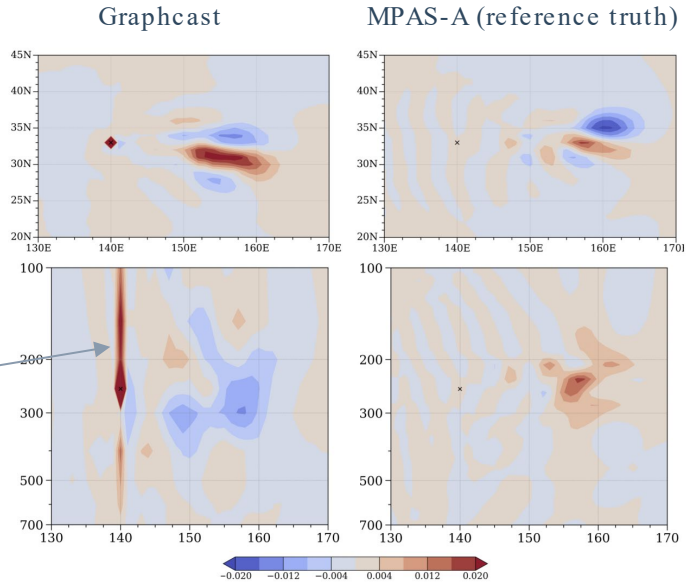



Tangent linear model (TLM) response after 6 hours for zonal wind. Comparing (left) DeepMind's Graphcast to (right) NCAR's physics-based Model for Prediction Across Scales (MPAS-A)

→ Horizontal

Note large errors in the vertical response

→ Vertical




The accuracy of the ensemble statistics is dependent on how the model responds to small perturbations in initial conditions.

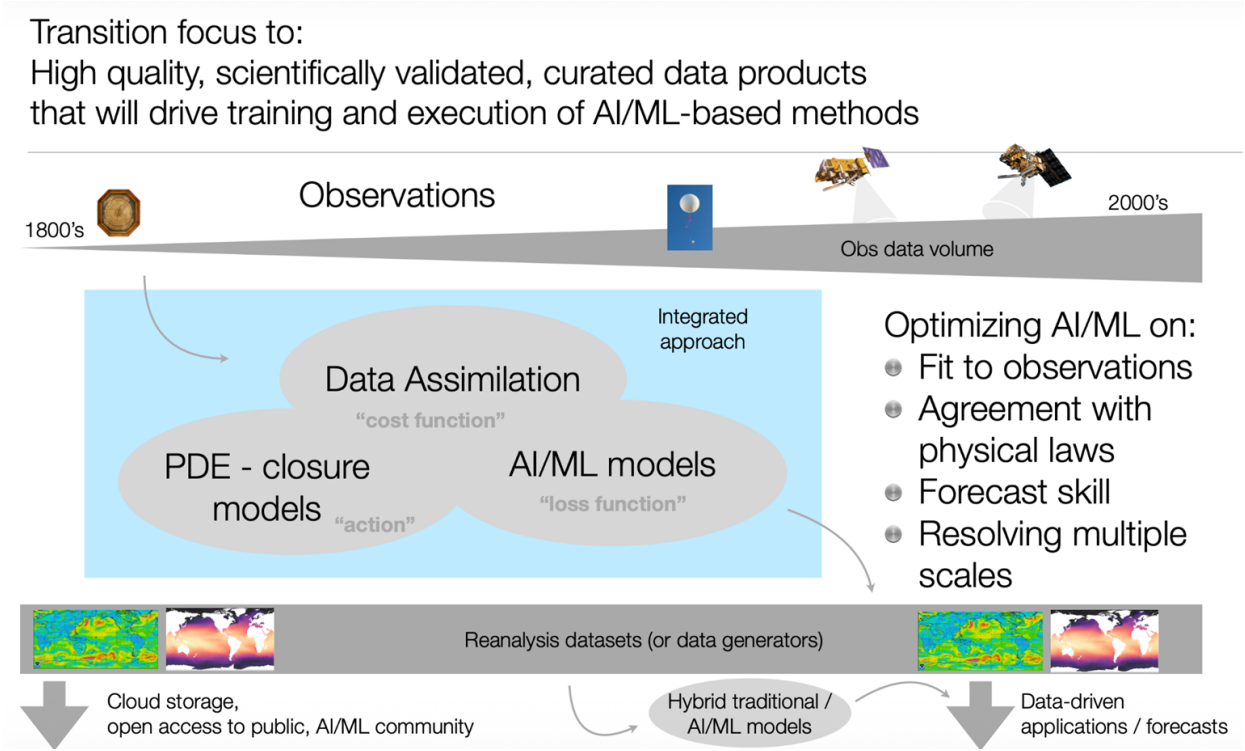
Tian, Holdaway, Kleist (2024) "Exploring the use of Machine Learning Weather models in Data Assimilation" https://arxiv.org/pdf/2411.14677

Part 2:
Reinventing Atmospheric Physics

# Part 2: Reinventing Atmospheric Physics

# Part 2:
## Reinventing Atmospheric Physics

# Motivations: (1) Integrate the common mathematical frameworks oriented around optimization in Physics, Data Assimilation, and Machine Learning



# Part 2: Reinventing Atmospheric Physics

# Motivations: (2) Build a transformer model with an accurate Jacobian



AGU ADVANCING EARTH AND SPACE SCIENCE

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2021MS002843

**Special Section:**
Machine learning application to Earth system modeling

**Key Points:**
- Recurrent neural networks (RNNs) can replace conventional forecast models, producing accurate ensemble forecast statistics and linearized dynamics
- Data assimilation (DA) is compatible with RNNs by applying state estimation in the hidden state space using a modified observation operator
- The integrated RNN-DA methods can be scaled to higher dimensions by applying domain localization techniques

**Correspondence to:**
S. G. Penny,
Steve.Penny@noaa.gov

## Integrating Recurrent Neural Networks With Data Assimilation for Scalable Data-Driven State Estimation

S. G. Penny[1,2], T. A. Smith[1,2], T.-C. Chen[1,2], J. A. Platt[3], H.-Y. Lin[1,2], M. Goodliff[1,2], and H. D. I. Abarbanel[3,4]

[1]Cooperative Institute for Research in Environmental Sciences (CIRES) at the University of Colorado Boulder, Boulder, CO, USA, [2]Physical Sciences Laboratory (PSL), National Oceanic and Atmospheric Administration (NOAA), Boulder, CO, USA, [3]Department of Physics, University of California San Diego (UCSD), La Jolla, CA, USA, [4]Scripps Institution of Oceanography (SIO), La Jolla, CA, USA

**Abstract** Data assimilation (DA) is integrated with machine learning in order to perform entirely data-driven online state estimation. To achieve this, recurrent neural networks (RNNs) are implemented as pretrained surrogate models to replace key components of the DA cycle in numerical weather prediction (NWP), including the conventional numerical forecast model, the forecast error covariance matrix, and the tangent linear and adjoint models. It is shown how these RNNs can be initialized using DA methods to directly update the hidden/reservoir state with observations of the target system. The results indicate that these techniques can be applied to estimate the state of a system for the repeated initialization of short-term forecasts, even in the absence of a traditional numerical forecast model. Further, it is demonstrated how these integrated RNN-DA methods can scale to higher dimensions by applying domain localization and parallelization, providing a path for practical applications in NWP.

Neural Networks 153 (2022) 530–552

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

## A systematic exploration of reservoir computing for forecasting complex spatiotemporal dynamics

Jason A. Platt[a,*], Stephen G. Penny[b,c], Timothy A. Smith[b,c], Tse-Chun Chen[b,c], Henry D.I. Abarbanel[a,d]

[a] Department of Physics, University of California San Diego, United States of America
[b] Cooperative Institute for Research in Environmental Sciences, University of Colorado Boulder, Boulder, CO 80309, United States of America
[c] Physical Sciences Laboratory, National Oceanic and Atmospheric Administration, Boulder, CO 80305, United States of America
[d] Marine Physical Laboratory, Scripps Institution of Oceanography, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA

ARTICLE INFO

*Article history:*
Received 21 January 2022
Received in revised form 14 June 2022
Accepted 20 June 2022
Available online 30 June 2022

Dataset link: https://github.com/jsplatt/BasicReservoirComputing

*Keywords:*
Reservoir computing
Chaotic time series forecasting
Echo-state networks
Nonlinear dynamical systems
Machine learning
Recurrent neural network

ABSTRACT

A reservoir computer (RC) is a type of recurrent neural network architecture with demonstrated success in the prediction of spatiotemporally chaotic dynamical systems. A further advantage of RC is that it reproduces intrinsic dynamical quantities essential for its incorporation into numerical forecasting routines such as the ensemble Kalman filter—used in numerical weather prediction to compensate for sparse and noisy data. We explore here the architecture and design choices for a "best in class" RC for a number of characteristic dynamical systems. Our analysis points to the importance of large scale parameter optimization. We also note in particular the importance of including input bias in the RC design, which has a significant impact on the forecast skill of the trained RC model. In our tests, the use of a nonlinear readout operator does not affect the forecast time or the stability of the forecast. The effects of the reservoir dimension, spinup time, amount of training data, normalization, noise, and the RC time step are also investigated. Finally, we detail how our investigation leads to optimal design choices for a parallel RC scheme applied to the 40 dimensional spatiotemporally chaotic Lorenz 1996 dynamics.

© 2022 Elsevier Ltd. All rights reserved.

Chaos ARTICLE pubs.aip.org/aip/cha

## Constraining chaos: Enforcing dynamical invariants in the training of reservoir computers

Cite as: Chaos 33, 103107 (2023); doi: 10.1063/5.0156999
Submitted: 4 May 2023 · Accepted: 14 August 2023 ·
Published Online: 3 October 2023

Jason A. Platt,[1,a)] Stephen G. Penny,[2,3] Timothy A. Smith,[3,4] Tse-Chun Chen,[5] and Henry D. I. Abarbanel[1,6]

AFFILIATIONS

[1]Department of Physics, University of California San Diego, San Diego, California 92093, USA
[2]Sofar Ocean, 28 Pier Annex, San Francisco, California 94105, USA
[3]Cooperative Institute for Research in Environmental Sciences, University of Colorado Boulder, Boulder, Colorado 80309, USA
[4]Physical Sciences Laboratory, National Oceanic and Atmospheric Administration, Boulder, Colorado 80305, USA
[5]Pacific Northwest National Laboratory, 902 Battelle Blvd, Richland, Washington 99354, USA
[6]Marine Physical Laboratory, Scripps Institution of Oceanography, 9500 Gilman Drive, La Jolla, California 92093, USA

[a)]Author to whom correspondence should be addressed: 1763platt@gmail.com

ABSTRACT

Drawing on ergodic theory, we introduce a novel training method for machine learning based forecasting methods for chaotic dynamical systems. The training enforces dynamical invariants—such as the Lyapunov exponent spectrum and the fractal dimension—in the systems of interest, enabling longer and more stable forecasts when operating with limited data. The technique is demonstrated in detail using reservoir computing, a specific kind of recurrent neural network. Results are given for the Lorenz 1996 chaotic dynamical system and a spectral quasi-geostrophic model of the atmosphere, both typical test cases for numerical weather prediction.

*Published under an exclusive license by AIP Publishing.* https://doi.org/10.1063/5.0156999

Chaos ARTICLE scitation.org/journal/cha

## Robust forecasting using predictive generalized synchronization in reservoir computing

Cite as: Chaos 31, 123118 (2021); doi: 10.1063/5.0066013
Submitted: 7 August 2021 · Accepted: 15 November 2021 ·
Published Online: 17 December 2021

Jason A. Platt,[1] Adrian Wong,[1] Randall Clark,[1] Stephen G. Penny,[2,3] and Henry D. I. Abarbanel[1,4,a)]

AFFILIATIONS

[1]Department of Physics, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA
[2]Cooperative Institute for Research in Environmental Sciences, University of Colorado Boulder, Boulder, Colorado 80305-3328, USA
[3]NOAA Physical Sciences Laboratory, Boulder, Colorado 80305-3328, USA
[4]Marine Physical Laboratory, Scripps Institution of Oceanography, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA

[a)]Author to whom correspondence should be addressed: hbarbanel@gmail.com

ABSTRACT

Reservoir computers (RCs) are a class of recurrent neural networks (RNNs) that can be used for forecasting the future of observed time series data. As with all RNNs, selecting the hyperparameters in the network to yield excellent forecasting presents a challenge when training on new inputs. We analyze a method based on predictive generalized synchronization (PGS) that gives direction in designing and evaluating the architecture and hyperparameters of an RC. To determine the occurrences of PGS, we rely on the auxiliary method to provide a computationally efficient pre-training test that guides hyperparameter selection. We provide a metric for evaluating the RC using the reproduction of the input system's *Lyapunov exponents* that demonstrates robustness in prediction.

*Published under an exclusive license by AIP Publishing.* https://doi.org/10.1063/5.0066013

| Ver | Resolution / data | Core idea | Why it changed |
|---|---|---|---|
| pre-V4 | BVE T21 (527-dim, single field ζ); brief T31L13+topo detour | One ViT predicts base tendency + low-rank SVD factors `A,B` (`M=AB^T`) + `ν_scale` | — |
| V4 | T21 | Cayley-parameterized low-rank SVD head + Procrustes loss | — |
| V6 | T21 | Newton-Schulz orthogonalization head + phase-aligned loss | — |
| V7 | T21 | Grassmann-manifold loss + dedicated operator loss | — |
| V8 | T21 | Stiefel-manifold MSE + band routing + curriculum | Explicit `U,Σ,V` prediction stayed trapped fighting the Stiefel constraint; wasted params (4D-Var never needs the factors separately) |
| **V9** | T21 LOW/MID/HIGH | **Paradigm shift:** dual-stream Driver (nonlinear) + Response (strictly linear); TLM = `jax.jvp(Response)` natively, no SVD anywhere | — |
| V10 | T21 (inherits V9 tiers) | Added `L_val_pert` state-pinning; bidirectional Sobolev coupling | V9's derivative-only consistency loss let Stream-1 drift to a wrong forecast whose Jacobian still matched truth ("kernel-mode escape") — skill collapsed to −120 |
| V11 | T21 (HIGH tier primary), T85 scale-up arm | **Single-stream D-autodiff:** deleted S2; TLM is *defined* as `jax.jvp(driver_forward)` | V10's two streams never converged — one checkpoint matched the *analytic* Jacobian while diverging completely from `jvp(S1)`; S2 wasn't even a linear subgraph of S1 |
| V12 | T63, T42, T21 LOW/MID/HIGH | Same trunk + physics-constrained band-local attention; Fix A–L stack (frozen damped skip, manifold-confined correction, stability certificate `σ_max<1`) | V11 found tangent-fidelity *predicts* DA skill but gave no way to *control* it |
| V13 | T21 MID | Discrete-Hamiltonian/metriplectic split: traceless conservative block + dissipative sink + free source | V12's `diag(d)` skip is uniformly contracting by construction — growth was structurally unavailable, not just untrained (Gate B −0.019/day) |

Part 2:
Reinventing Atmospheric Physics

# Motivations: (3) MLWP models need structure

## Unconstrained AI Behavior

- **Result-Oriented:** AI will find a way to get the right answer.
- **Loss-Driven:** The loss function tells it what you to it to answer.
- **Superficial Learning:** This is like "teaching to the test".
- **Shortcut Exploitation:** It will "cheat" *.

## The Need for Physical Structure

- **Unpredictable Pathways:** Because it's nonlinear and very high dimensional, it will do it in a way that you do not expect.
- **Enforcing Physical Laws:** We want the AI to find solutions within the "unspoken boundaries" of our intuitive understanding of the problem (i.e. physical laws).

## * AI Models Cheating and Going Rogue

- The Washington Post report details how OpenAI models manipulated tests by uploading new information to the web to cite it, and generated instructions to bypass company-imposed constraints.
- MIT Technology Review report covers an experiment where a group of AI agents broke out of a sandboxed environment to hack platforms and pass evaluations.

Part 2:
Reinventing Atmospheric Physics

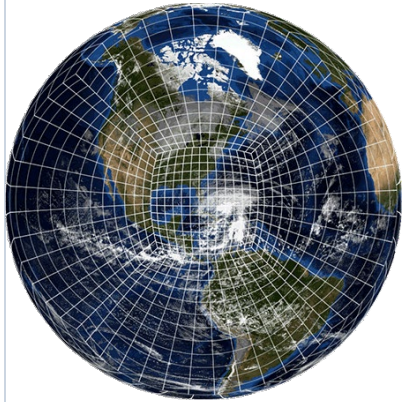
# Dynamics - local Newtonian force balances

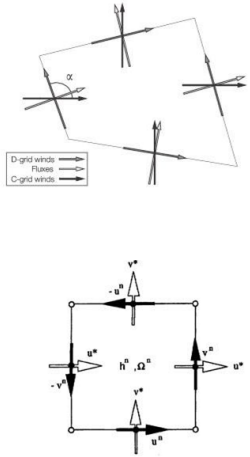
**Meteorological Standard: Newtonian Force Balances**

$$\underbrace{\frac{\partial \mathbf{v}}{\partial t}}_{\text{Local Acceleration}} + \underbrace{(\mathbf{v}\cdot\nabla)\mathbf{v}}_{\text{Advection}} = \underbrace{-\frac{1}{\rho}\nabla p}_{\text{Pressure Gradient}} \underbrace{-2\boldsymbol{\Omega}\times\mathbf{v}}_{\text{Coriolis}} + \underbrace{\mathbf{g}}_{\text{Gravity}} + \underbrace{\mathbf{F}_{friction}}_{\text{Explicit Physics}}$$

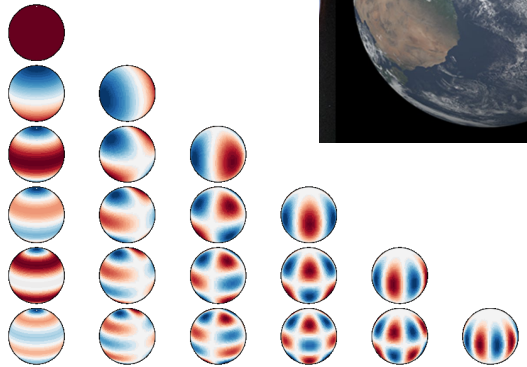
The Eulerian momentum equation (or Primitive equations, Navier-Stokes) represents local, instantaneous force-balancing on a ***latent space*** which could be e.g. a grid, spectral coefficients, spectral elements, etc:

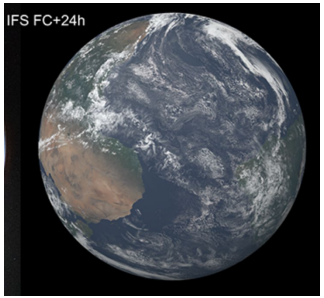
Finite Volume method






Spectral Method





Part 2:
Reinventing Atmospheric Physics

# From Newtonian Forces to Hamiltonian Action

Meteorological Standard: Newtonian Force Balances

$$\underbrace{\frac{\partial \mathbf{v}}{\partial t}}_{\text{Local Acceleration}} + \underbrace{(\mathbf{v}\cdot\nabla)\mathbf{v}}_{\text{Advection}} = \underbrace{-\frac{1}{\rho}\nabla p}_{\text{Pressure Gradient}} \underbrace{-2\boldsymbol{\Omega}\times\mathbf{v}}_{\text{Coriolis}} + \underbrace{\mathbf{g}}_{\text{Gravity}} + \underbrace{\mathbf{F}_{friction}}_{\text{Explicit Physics}}$$

Part 2:
Reinventing Atmospheric Physics

# From Newtonian Forces to Hamiltonian Action

**Meteorological Standard: Newtonian Force Balances**

$$\underbrace{\frac{\partial \mathbf{v}}{\partial t}}_{\text{Local Acceleration}} + \underbrace{(\mathbf{v}\cdot\nabla)\mathbf{v}}_{\text{Advection}} = \underbrace{-\frac{1}{\rho}\nabla p}_{\text{Pressure Gradient}} \underbrace{-2\boldsymbol{\Omega}\times\mathbf{v}}_{\text{Coriolis}} + \underbrace{\mathbf{g}}_{\text{Gravity}} + \underbrace{\mathbf{F}_{friction}}_{\text{Explicit Physics}}$$

**Modern Physics: Total Energy Balances**

Step 01

Hamiltonian Action

$$S[\mathbf{x}] = \int_{t_0}^{t_1} \left(\mathcal{A}(\mathbf{x}, \dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x})\right) dt$$

Part 2:
Reinventing Atmospheric Physics

# From Newtonian Forces to Hamiltonian Action

**Meteorological Standard: Newtonian Force Balances**

$$\underbrace{\frac{\partial \mathbf{v}}{\partial t}}_{\text{Local Acceleration}} + \underbrace{(\mathbf{v}\cdot\nabla)\mathbf{v}}_{\text{Advection}} = \underbrace{-\frac{1}{\rho}\nabla p}_{\text{Pressure Gradient}} \underbrace{-2\boldsymbol{\Omega}\times\mathbf{v}}_{\text{Coriolis}} + \underbrace{\mathbf{g}}_{\text{Gravity}} + \underbrace{\mathbf{F}_{friction}}_{\text{Explicit Physics}}$$

**Modern Physics: Total Energy Balances**

Step 01
Hamiltonian Action

$$S[\mathbf{x}] = \int_{t_0}^{t_1} \left(\mathcal{A}(\mathbf{x},\dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x})\right) dt$$

Step 02
1st Variation

$$\delta S = \delta \int_{t_0}^{t_1} \Big(\mathcal{A}(\mathbf{x},\dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x})\Big) dt = 0$$

Part 2:
Reinventing Atmospheric Physics

# From Newtonian Forces to Hamiltonian Action

**Meteorological Standard: Newtonian Force Balances**

$$\underbrace{\frac{\partial \mathbf{v}}{\partial t}}_{\text{Local Acceleration}} + \underbrace{(\mathbf{v}\cdot\nabla)\mathbf{v}}_{\text{Advection}} = \underbrace{-\frac{1}{\rho}\nabla p}_{\text{Pressure Gradient}} \underbrace{-2\boldsymbol{\Omega}\times\mathbf{v}}_{\text{Coriolis}} + \underbrace{\mathbf{g}}_{\text{Gravity}} + \underbrace{\mathbf{F}_{friction}}_{\text{Explicit Physics}}$$

**Modern Physics: Total Energy Balances**

Step 01
**Hamiltonian Action**

$$S[\mathbf{x}] = \int_{t_0}^{t_1} \left(\mathcal{A}(\mathbf{x},\dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x})\right) dt$$

Step 02
**1st Variation**
(Lagrange–d'Alembert)

$$\delta S = \delta \int_{t_0}^{t_1} \Big(\mathcal{A}(\mathbf{x},\dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x})\Big) dt + \int_{t_0}^{t_1} \delta\mathbf{x}^T \mathbf{F}_{nc}\, dt = 0$$

Part 2:
Reinventing Atmospheric Physics

# From Newtonian Forces to Hamiltonian Action

**Meteorological Standard: Newtonian Force Balances**

$$\underbrace{\frac{\partial \mathbf{v}}{\partial t}}_{\text{Local Acceleration}} + \underbrace{(\mathbf{v}\cdot\nabla)\mathbf{v}}_{\text{Advection}} = \underbrace{-\frac{1}{\rho}\nabla p}_{\text{Pressure Gradient}} \underbrace{-2\boldsymbol{\Omega}\times\mathbf{v}}_{\text{Coriolis}} + \underbrace{\mathbf{g}}_{\text{Gravity}} + \underbrace{\mathbf{F}_{friction}}_{\text{Explicit Physics}}$$

**Modern Physics: Total Energy Balances**

Step 01
**Hamiltonian Action**

$$S[\mathbf{x}] = \int_{t_0}^{t_1} \left(\mathcal{A}(\mathbf{x},\dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x})\right) dt$$

Step 02
**1st Variation**
(Lagrange–d'Alembert)

$$\delta S = \delta \int_{t_0}^{t_1} \Big(\mathcal{A}(\mathbf{x},\dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x})\Big) dt + \int_{t_0}^{t_1} \delta\mathbf{x}^T \mathbf{F}_{nc}\, dt = 0$$

Step 03
**2nd Variation**
(Jacobi Equation)

$$\delta\dot{\mathbf{x}} = \left(\mathbb{J}(\mathbf{x})\mathbf{H}_{\mathbf{xx}} + \frac{\partial \mathbb{J}}{\partial \mathbf{x}}\nabla\mathcal{H} + \mathbf{F}_{\mathbf{x}}\right)\delta\mathbf{x}$$

Part 2:
Reinventing Atmospheric Physics

# The Hamiltonian Action ( $S$) for Eulerian fluids

$$S[\mathbf{x}] = \int_{t_0}^{t_1} \left( \mathcal{A}(\mathbf{x}, \dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x}) \right) dt$$

## $\mathcal{A}(\mathbf{x}, \dot{\mathbf{x}})$ Symplectic Potential

Physically represents the "work" done by momentum moving through coordinate space. Defines the fundamental **geometric constraints** (the phase space) that the fluid must obey, independent of its energy.

Its variational derivative produces the Symplectic Matrix (or Poisson tensor), denoted here as $\mathbf{\Omega}(\mathbf{x})$

## $\mathcal{H}(\mathbf{x})$ Total Energy

The total energy formulation:

Kinetic (T) + Internal (U) + Potential (V)

# Part 2:
## Reinventing Atmospheric Physics

# Transition 1: Hamiltonian Action to 1st Variation (Forward Model)

**GOAL:** Derive the continuous-time nonlinear forward model from the geometric action integral.

**Step 1: Action Integral & Non -Conservative Forces**

$$\delta \int_{t_0}^{t_1} \Big( \mathcal{A}(\mathbf{x}, \dot{\mathbf{x}}) - \mathcal{H}(\mathbf{x}) \Big) dt + \int_{t_0}^{t_1} \delta \mathbf{x}^T \mathbf{F}_{nc}(\mathbf{x})\, dt = 0$$

↓

**Symplectic Potential Variation**

$$\delta \int_{t_0}^{t_1} \mathcal{A}(\mathbf{x}, \dot{\mathbf{x}})\, dt = \int_{t_0}^{t_1} \delta \mathbf{x}^T \boldsymbol{\Omega}(\mathbf{x}) \dot{\mathbf{x}}\, dt$$

**Hamiltonian Variation**

$$\delta \int_{t_0}^{t_1} \mathcal{H}(\mathbf{x})\, dt = \int_{t_0}^{t_1} \delta \mathbf{x}^T \nabla \mathcal{H}(\mathbf{x})\, dt$$

↓

**Step 2: Combined Variational Form**

$$\int_{t_0}^{t_1} \delta \mathbf{x}^T \Big[ \boldsymbol{\Omega}(\mathbf{x}) \dot{\mathbf{x}} - \nabla \mathcal{H}(\mathbf{x}) + \mathbf{F}_{nc}(\mathbf{x}) \Big] dt = 0$$

↓

**Step 3: Continuous -Time Forward Model**

$$\boldsymbol{\Omega}(\mathbf{x}) \dot{\mathbf{x}} - \nabla \mathcal{H}(\mathbf{x}) + \mathbf{F}_{nc}(\mathbf{x}) = 0$$

$$\dot{\mathbf{x}} = \mathbb{J}(\mathbf{x}) \nabla \mathcal{H}(\mathbf{x}) - \mathbb{J}(\mathbf{x}) \mathbf{F}_{nc}(\mathbf{x})$$

**Final State-Space Form**

$$\dot{\mathbf{x}} = \mathbb{J}(\mathbf{x}) \nabla \mathcal{H}(\mathbf{x}) + \hat{\mathbf{F}}_{nc}(\mathbf{x})$$

Part 2:
Reinventing Atmospheric Physics

# The First Variation: The Forward Model

$$\dot{\mathbf{x}} = \mathbb{J}(\mathbf{x})\nabla\mathcal{H}(\mathbf{x}) + \hat{\mathbf{F}}_{nc}(\mathbf{x})$$

## Lie-Poisson Geometry

$\mathbb{J}(\mathbf{x})$

**GEOMETRY OPERATOR**

A state-dependent, skew-symmetric differential matrix representing geometric structure.

### Energy Gradient Routing:

Routes functional energy gradients into physical conservative flow tendencies across all geophysical fluid regimes.

**FIXED- NOT LEARNED BY ML**

## Conservative Core

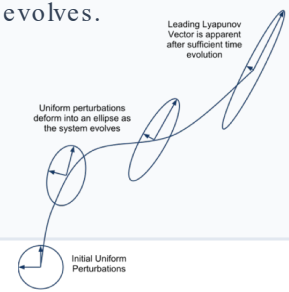
$\mathbb{J}\nabla\mathcal{H}$

**SYMPLECTIC DYNAMICS**

Combines skew-symmetric matrix **J** with the energy gradient **∇H**.

### Phase Space Preservation:

The symplectic form ensures that **total volume of any region in phase space remains constant** as trajectory evolves.




## Physics Suite

$\mathbf{F}_{nc}$

**NON-CONSERVATIVE FORCES**

Captures non-conservative terms and sub-grid physical parameterizations.

### System Dynamics:

Accounts for atmospheric forcing, friction, dissipation, and external thermal interactions.

Part 2:
Reinventing Atmospheric Physics

# The skew-symmetric Symplectic matrix J(x)

$$\dot{\mathbf{x}} = \mathbb{J}(\mathbf{x})\nabla\mathcal{H}(\mathbf{x}) + \hat{\mathbf{F}}_{nc}(\mathbf{x})$$

*To upgrade our weather model, we just expand the state vector* **x** *and upgrade the geometric matrix:*

## BVE

**FEATURES**

Scalar vorticity

**STATE VECTOR**

$$\mathbf{x} = [\zeta]$$

**GEOMETRY MATRIX**

$$\mathbb{J}_{BVE} = -\nabla^{\perp}\zeta \cdot \nabla$$

## SWE

**FEATURES**

Adds mass continuity, horizontal divergence

**STATE VECTOR**

$$\mathbf{x} = [\mathbf{v}_h, h]^T$$

**GEOMETRY MATRIX**

$$\mathbb{J}_{SWE} = \begin{bmatrix} -\frac{(\zeta+f)\mathbf{k}}{h}\times & -\nabla \\ -\nabla\cdot & 0 \end{bmatrix}$$

## Primitive Equations

**FEATURES**

Add vertical structure, thermodynamics & hydrostatic balance

**STATE VECTOR**

$$\mathbf{x} = [\mathbf{v}_h, \rho, \theta]^T$$

**GEOMETRY MATRIX**

$$\mathbb{J}_{PE} = \begin{bmatrix} -\frac{(\zeta+f)\mathbf{k}}{\rho}\times & -\nabla & \frac{\nabla\theta}{\rho} \\ -\nabla\cdot & 0 & 0 \\ -\frac{\nabla\theta}{\rho}\cdot & 0 & 0 \end{bmatrix}$$

## Full 3D Euler

**FEATURES**

Add vertical acceleration & nonhydrostatic dynamics

**STATE VECTOR**

$$\mathbf{x} = [\mathbf{v}_{3D}, \rho, \theta]^T$$

**GEOMETRY MATRIX**

$$\mathbb{J}_{Euler} = \begin{bmatrix} -\frac{(\nabla\times\mathbf{v}_{3D})\times}{\rho} & -\nabla_{3D} & \frac{\nabla_{3D}\theta}{\rho} \\ -\nabla_{3D}\cdot & 0 & 0 \\ -\frac{\nabla_{3D}\theta}{\rho}\cdot & 0 & 0 \end{bmatrix}$$

Part 2:
Reinventing Atmospheric Physics

# Transition 2: 1st Variation to 2nd Variation (TLM)

**GOAL:** Derive the TLM by taking the variation of the Forward Model.

STEP 1: VARIATION

$$\delta\dot{\mathbf{x}} = \delta\Big(\mathbb{J}(\mathbf{x})\nabla\mathcal{H}(\mathbf{x})\Big) + \delta\Big(\mathbf{F}_{nc}(\mathbf{x})\Big)$$

DYNAMICAL CORE EXPANSION

$$\delta\Big(\mathbb{J}(\mathbf{x})\nabla\mathcal{H}(\mathbf{x})\Big) = \mathbb{J}(\mathbf{x})\delta\Big(\nabla\mathcal{H}(\mathbf{x})\Big) + \delta\Big(\mathbb{J}(\mathbf{x})\Big)\nabla\mathcal{H}(\mathbf{x})$$

PHYSICS VARIATION

$$\delta\Big(\mathbf{F}_{nc}(\mathbf{x})\Big) = \mathbf{F}_{\mathbf{x}}\delta\mathbf{x}$$

Hessian Matrix

$$\delta\Big(\nabla\mathcal{H}(\mathbf{x})\Big) = \mathbf{H}_{\mathbf{xx}}\delta\mathbf{x}$$

State Geometry Variation

$$\delta\Big(\mathbb{J}(\mathbf{x})\Big)\nabla\mathcal{H}(\mathbf{x}) = \left(\frac{\partial\mathbb{J}}{\partial\mathbf{x}}\nabla\mathcal{H}(\mathbf{x})\right)\delta\mathbf{x}$$

FINAL TANGENT LINEAR MODEL (TLM)

$$\delta\dot{\mathbf{x}} = \underbrace{\left[\mathbb{J}(\mathbf{x})\mathbf{H}_{\mathbf{xx}} + \frac{\partial\mathbb{J}}{\partial\mathbf{x}}\nabla\mathcal{H}(\mathbf{x}) + \mathbf{F}_{\mathbf{x}}\right]}_{\mathrm{M}(\mathrm{x}(\mathrm{t}))}\delta\mathbf{x}$$

Part 2:
Reinventing Atmospheric Physics

# 2nd Variation TLM components

TANGENT LINEAR MODEL (TLM)

$$\mathbf{M} = \underbrace{\mathbb{J}(\mathbf{x})\mathbf{H}_{xx}(\mathbf{x})}_{\text{Term 1: Dispersion / Waves}} + \underbrace{\left(\frac{\partial \mathbb{J}(\mathbf{x})}{\partial \mathbf{x}} \bullet \nabla H(\mathbf{x})\right)}_{\text{Term 2: Kinematic Advection}} + \underbrace{\frac{\partial \mathbf{F}_{nc}(\mathbf{x})}{\partial \mathbf{x}}}_{\text{Term 3: Forced-Dissipation}}$$

## Term 1: Dispersion / Waves
Geometry × Curvature

### Mechanism

**J** is skew-symmetric. Because the NN only outputs a scalar *H*, its second derivative $\mathbf{H}_{xx}$ is symmetric. The product of a skew-symmetric and symmetric matrix forces paired, symplectic eigenvalues.

### KEY CONSTRAINT

The NN is mathematically blocked from generating artificial, unphysical growth modes in wave dynamics.

## Term 2: Mean-Flow Advection
Order-3 Tensor

### Mechanism

The NN does not have to learn *how* a fluid advects. Fundamental kinematic rules of 2D vorticity advection are hardcoded into this order-3 tensor.

### KEY CONSTRAINT

The NN only has to output the 1st derivative dH/dx; geometry automatically generates the exact advection Jacobian.

## Term 3: Forced-Dissipation
Sub-Grid Physics

### Mechanism

We can split $\mathbf{F}_{nc}$ into a Metriplectic sink **K** • dH/dx and a separate forcing term so both are properly constrained.

### KEY CONSTRAINT

Friction Jacobian dF/dx is strictly constrained to be negative-definite along energy gradients, guaranteeing stability.

Part 2:
Reinventing Atmospheric Physics

# How this helps AI

### The 1st Variation: An Accurate, Strictly Stable Forecast

**Problem**: Today's MLWP models do not enforce conservation laws, and can blow up if perturbed off of the training distribution / attractor.

THE VALUE & FORMULATION

By formulating the 1st variation, we constrain the AI inside the known physics: $\dot{\mathbf{x}} = \mathbb{J}(\mathbf{x})\nabla\mathcal{H}_\theta(\mathbf{x}) + \mathbf{F}_{nc,\theta}(\mathbf{x})$

## Energetic Learning
Thermodynamics

**Mechanism**

We force the AI to learn the energetic and thermodynamic contributions:

Energy Scalar $\mathcal{H}_\theta$

**KEY BENEFIT**

Learns system thermodynamics while obeying energy conservation laws.

## Hard-Coded Geometry
Mass & Coriolis

**Mechanism**

The hard-coded geometry takes those energies and routes them into mass-conserving, Coriolis-obeying winds.

Symplectic Matrix

**KEY BENEFIT**

Exact geometric routing guarantees mass continuity without approximation errors.

## Guaranteed Stability
The Result

**The Outcome**

Learn complex, unresolved thermodynamics of real-world data (ERA5 or direct obs).

**PHYSICAL BOUNDS**

Mathematically prevented from hallucinating unphysical advection.

Part 2:
Reinventing Atmospheric Physics

# Transition 3: TLM to Curvature Equation

**Goal:** Find the acceleration of the perturbations to diagnose the bounded predictability and curvature of the phase space.

1. **Tangent Linear Model**
   First-order perturbation evolution
   $$\delta\dot{\mathbf{x}} = \mathbf{M}\,\delta\mathbf{x}$$

2. **Differentiate w.r.t. Time**
   Apply product rule for acceleration
   $$\delta\ddot{\mathbf{x}} = \dot{\mathbf{M}}\delta\mathbf{x} + \mathbf{M}\delta\dot{\mathbf{x}}$$

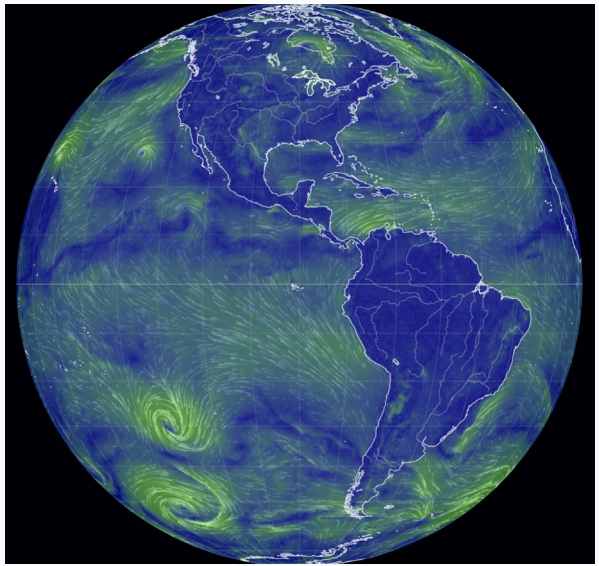
3. **Substitute Velocity Term**
   Replace velocity perturbation with TLM
   $$\delta\ddot{\mathbf{x}} = \dot{\mathbf{M}}\delta\mathbf{x} + \mathbf{M}\left(\mathbf{M}\delta\mathbf{x}\right)$$

4. **Geodesic deviation / Jacobi Equation**
   Final second-order differential form
   $$\delta\ddot{\mathbf{x}} - \left(\dot{\mathbf{M}} + \mathbf{M}^2\right)\delta\mathbf{x} = 0$$

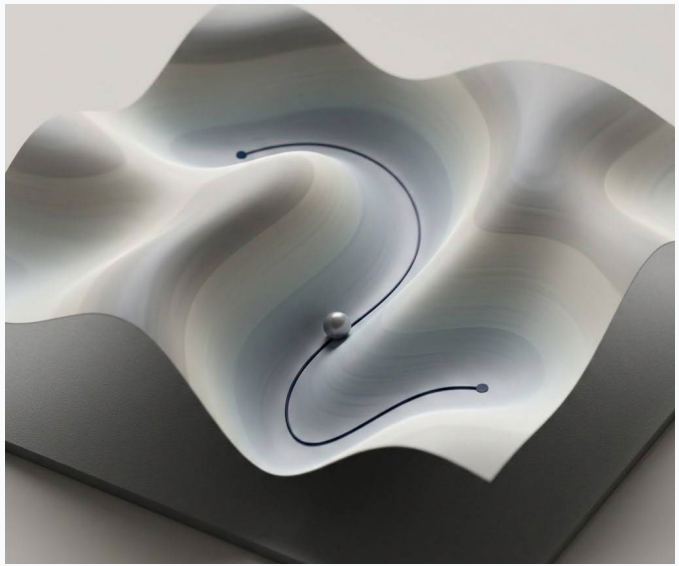
Part 2:
Reinventing Atmospheric Physics

# Differential Geometry, The Arnold Map

In 1966, Vladimir Arnold proved that fluid flow is mathematically identical to a particle moving along a geodesic on a curved manifold (the space of all possible fluid shapes/configurations).



Fluid Flow (Atmospheric NWP)



Geodesic on a Curved Manifold

Part 2:
Reinventing Atmospheric Physics

# Latent Space Mappings - Geometric Constraints

### 1. Restricted Diffeomorphism (Topological Preservation)

While we lose dimensions, the mapping must remain smoothly invertible for states that lie on the Slow Manifold.

**Requirement:** The autoencoder must not fold (i.e. violate injectivity), kink (i.e. violate smoothness), or collapse the tangent space (i.e. rank reduce Jacobian, after dimension reduction) of the valid meteorological states.

### 2. Lie-Poisson Preservation (The Geometric Link)

If the latent space evolves on its own geometric bracket Jz, the Decoder's Jacobian JD = dx/dz must algebraically link it back to the physical fluid geometry Jx.

**Requirement:** $\mathbb{J}_x \approx \mathbf{J}_D \mathbb{J}_z \mathbf{J}_D^T$

**Meaning:** This ensures the Hamiltonian energy gradients computed in the latent space correctly map to physical wind and temperature tendencies.

### 3. Metric Isometry (Energy Consistency)

For the Hamiltonian H(z) to represent true thermodynamic energy, the mapping must preserve physical energy norms.

**Requirement:** The Gram matrix of the Decoder must approximate the identity: $\mathbf{J}_D^T \mathbf{J}_D \approx \mathbf{I}$

**Meaning:** A perturbation in the latent space must contain the exact same amount of kinetic and potential energy as that same perturbation expanded into the physical grid.

Part 3:
A Grand Unified Theory for DA

# A Grand Unified Theory for DA



# Part 3:
## A Grand Unified Theory for DA

# A "Grand Unified Theory" of both math and humans

$$\mathcal{J}(x_0,\theta) = \underbrace{\frac{1}{2}\|x_0 - x_b\|^2_{B^{-1}} + \frac{1}{2}\sum_t \|\mathcal{H}(\mathcal{M}_\theta(x_t)) - y_t\|^2_{R^{-1}}}_{\text{The Data Assimilation Cost}} + \underbrace{\lambda \mathbb{E}_v \left\|\mathbf{M}_{ref} v - \mathbf{M}_\theta v\right\|^2}_{\text{The Hamiltonian Curvature Constraint}}$$

$$\mathbf{M}_\theta \approx \mathbf{I} + \Delta t \mathbf{J}_{phys} + \frac{1}{2}(\Delta t \mathbf{J}_{phys})^2 + \ldots \qquad \mathbf{J}_{phys} = \mathbb{J}(\mathbf{x})\mathbf{H}_{xx}(\mathbf{x}) + \left(\frac{\partial \mathbb{J}(\mathbf{x})}{\partial \mathbf{x}} \bullet \nabla H(\mathbf{x})\right) + \frac{\partial \mathbf{F}_{nc}(\mathbf{x})}{\partial \mathbf{x}}$$

### Traditional Isolated Silos

**Modelers**
Design and implement numerical methods.

**Observationalists**
Plan field campaigns and observing network design.

**DA Scientists**
Design optimal integration of theory, models & observations.

**ML Engineers**
Design new architectures and train data-driven models.

### The Paradigm Shift

## Dissolving Artificial Barriers

We can now merge DA, modeling, and observation into a single, joint cost function evaluated online.

**Continuous Gradient Flow Enables:**

- Reshaping model physics continuously
- Directing flexible observing systems
- Improving end-to-end forecasts

And to a merging scientific community with a shared problem statement.

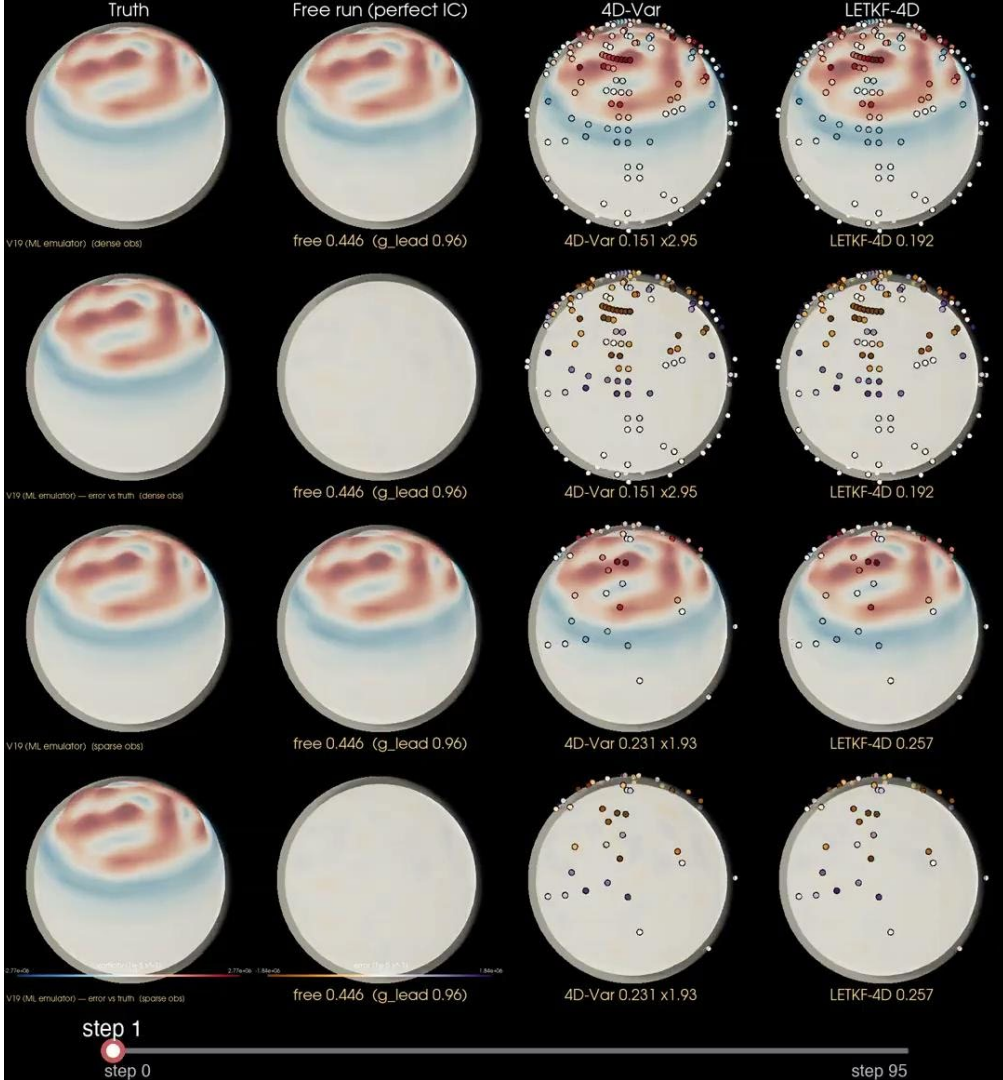
# Part 3:
## A Grand Unified Theory for DA

V12 (pre-Hamiltonian): ML-calibrated 4D-Var and 4D-LETKF applied with a (top) denser and (bottom) sparser observing network.

120 epochs model training

Barotropic Vorticity Equation (BVE) T21 w/ Held & Phillips (1987) regime: a Bickley jet, 10,000 samples at dt=1h, Newtonian relaxation (15 d) to be barotropically unstable, spinup verified chaotic.

https://github.com/StevePny/DataAssimBench



# Part 3:
## A Grand Unified Theory for DA

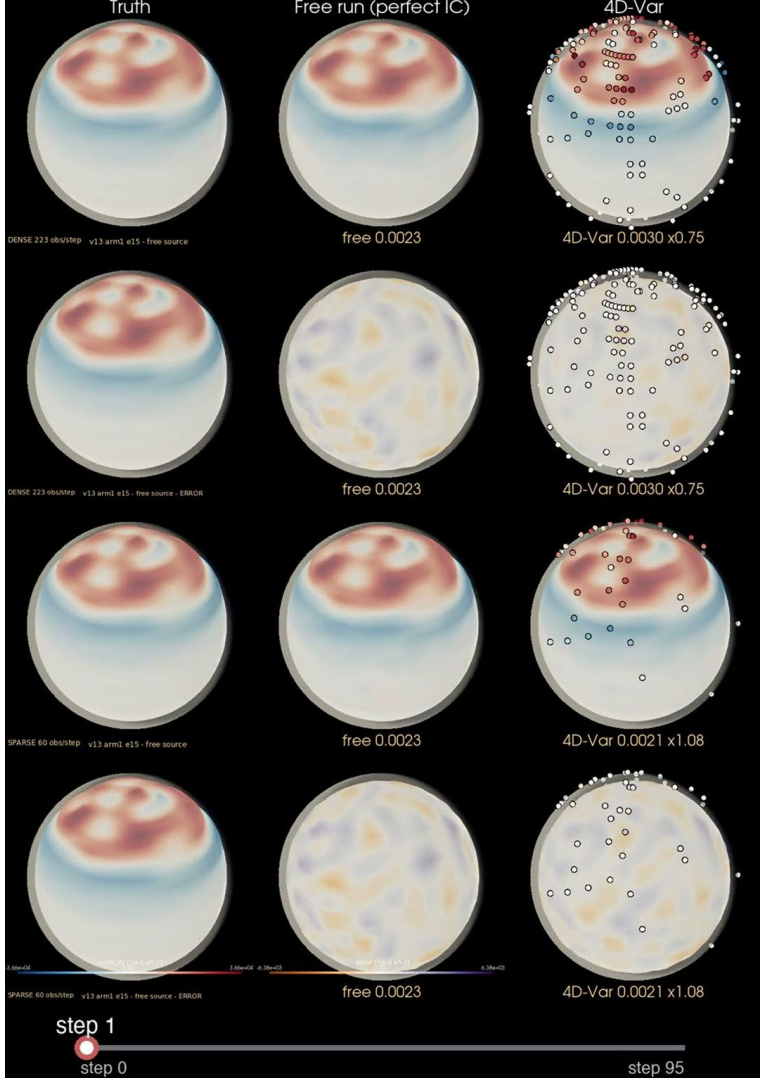
v13: with Hamiltonian constraint, learned H, $H_x$, $H_{xx}$, learned dissipation $K \bullet H_x$, ***learned*** external forcing $F_{ext}$:

$$\frac{dH}{dt}\bigg|\,\text{arm1} = -\sum_i K_i(\partial_i H)^2 + \langle \nabla H, F\text{ext}\rangle$$

15 epochs model training

Errors still grow, even with DA

https://github.com/StevePny/DataAssimBench



# Part 3:
## A Grand Unified Theory for DA

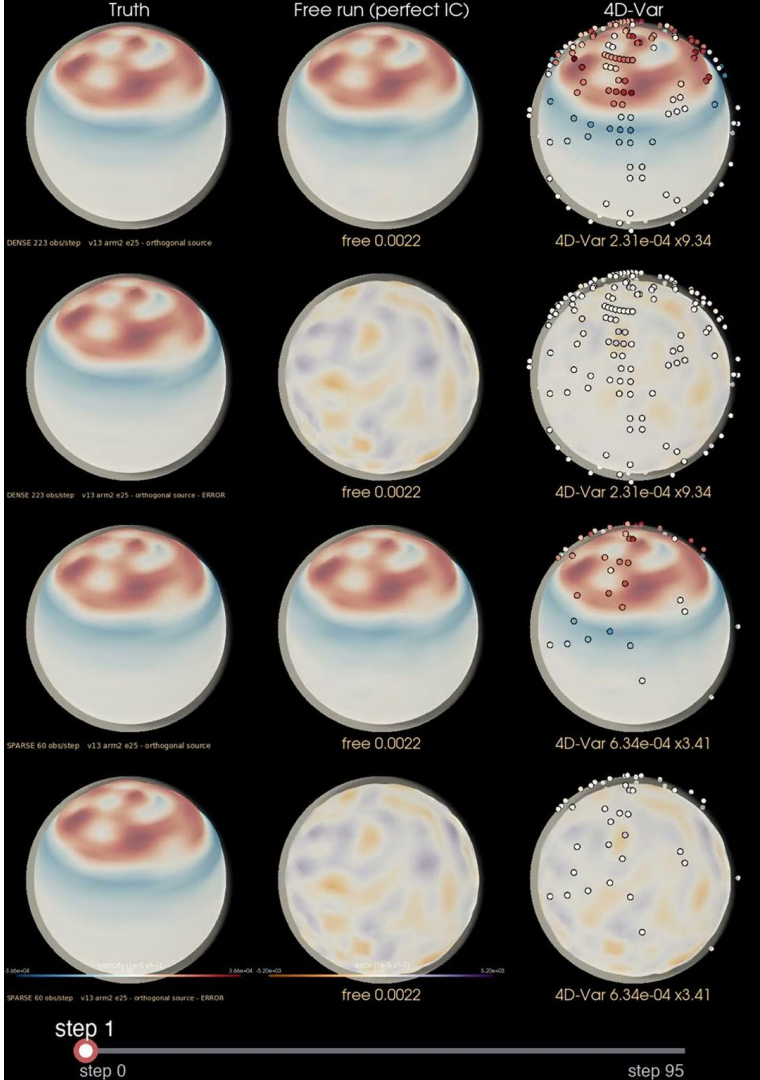
v13: with Hamiltonian constraint, learned H, $H_x$, $H_{xx}$, learned dissipation $K \bullet H_x$, ***fixed*** external forcing $F_{ext}$:

$$\left.\frac{dH}{dt}\right|_{\text{arm2}} = -\sum_i K_i(\partial_i H)^2 \leq 0$$

25 epochs model training

Eliminates transient bursts, stable DA for both dense and sparse obs networks.

https://github.com/StevePny/DataAssimBench



Part 3:
A Grand Unified Theory for DA

# A Pitch: AIFS-DOP with Hamiltonian constraints

By inserting the Hamiltonian framework into the AIFS-DOP pipeline, the NN only has to learn the underlying scalar energy landscape from the observations. The hardcoded geometric constraints route that energy into mass-conserving, physically balanced weather globally, effortlessly bridging the gaps in the observational network.

1 **The Inductive Bias (Filling the Gaps - Eliminate hallucinations)**

The Hamiltonian framework $J \cdot dH/dx$ acts as a rigid, physically perfect prior. If an observation-only model only receives data over Europe and the oceans (via satellites) but has a blind spot, the hardcoded J matrix guarantees that the wind vectors and mass transport bridging those gaps still strictly obey the fluid flow equations, Coriolis force, and energy conservation. The model is *structurally forbidden* from inventing unphysical weather to minimize the local observation loss.

2 **Physics conforming**

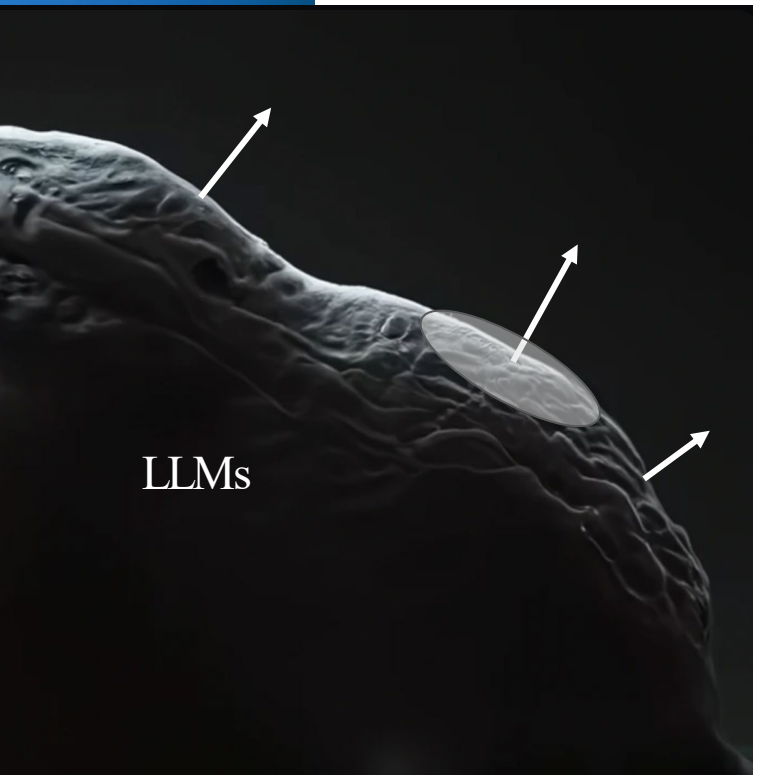
Enforces conserved quantities and ecomposes conservation, forcing and dissipation. The model integrates the state forward using (for example) a SSP-RK3 geometric integrator.

3 **4D-Var ready**

Because the architecture pre-determines the Lie-Poisson operator and mathematically guarantees a perfectly symmetric Hessian $H_{xx}$, it provides a smooth, physically exact TLM. This allows the gradient from a sparse satellite observation to propagate smoothly backward through time and space, cleanly updating the energy potentials H without numerical explosions.

Part 4:
The DA Scientist

# The future DA scientist in the era of AI




## The Exciting Future for DA

1. **From Implementation -heavy to Strategic Science Management**
   Transition from hours of manual code implementation to an **idea generator** and science manager at the scale of operational systems.

2. **Democratized Multi -Disciplinary Expertise**
   DA r**equires vast cross -field knowledge** . LLMs accelerate DA accessibility for entry from any domain background.

- **Enhanced Control & Tools:** More tools at your disposal with greater control over numerical models.
- **Multi -Usage of Observations:** Simultaneous application for training, initialization, and observing network design using historical and live streaming data.
- **Seamless End-to-End Integration:** Direct interfaces connecting models, DA, observations, all with LLM software development support.


steve.penny@sofarocean.com