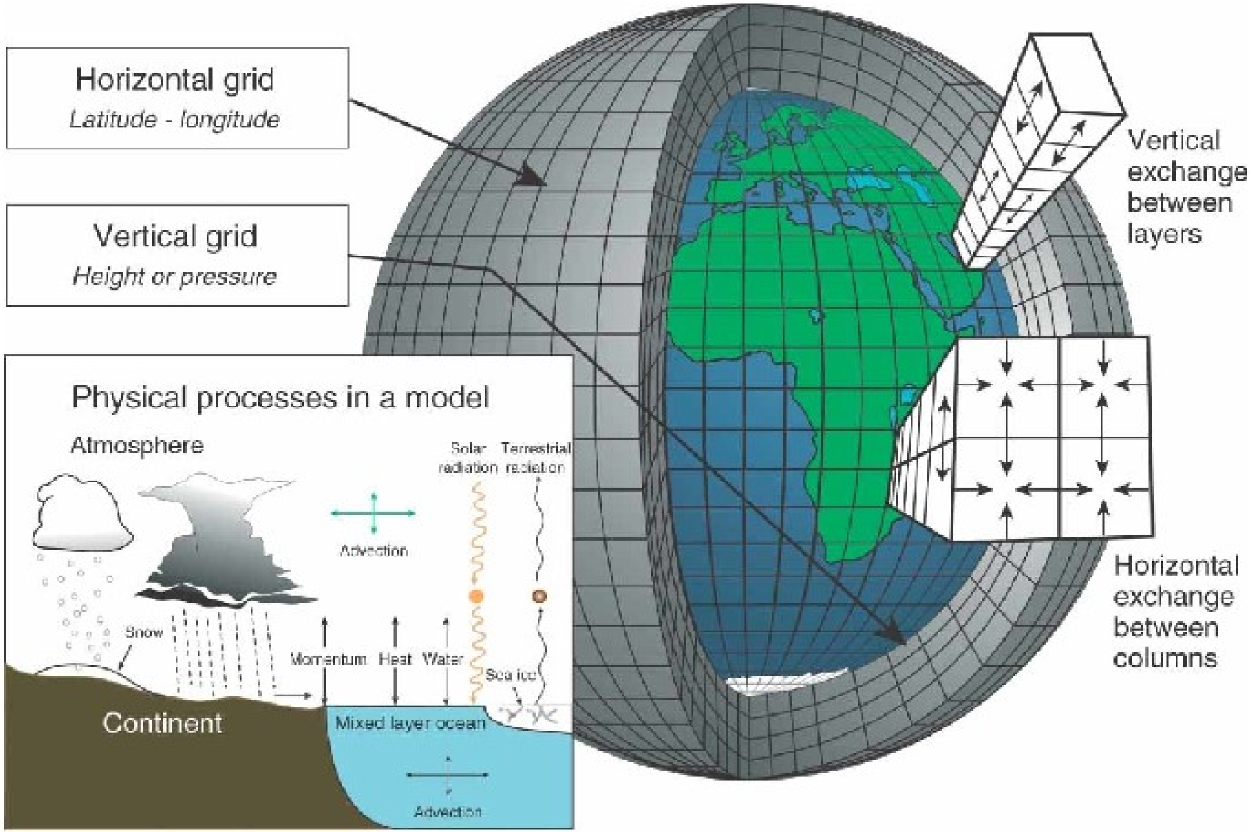
Combining **Neural Networks** and **Physics** for Weather and Climate Predictions

STK-IN9300
Anna Lina Sjur

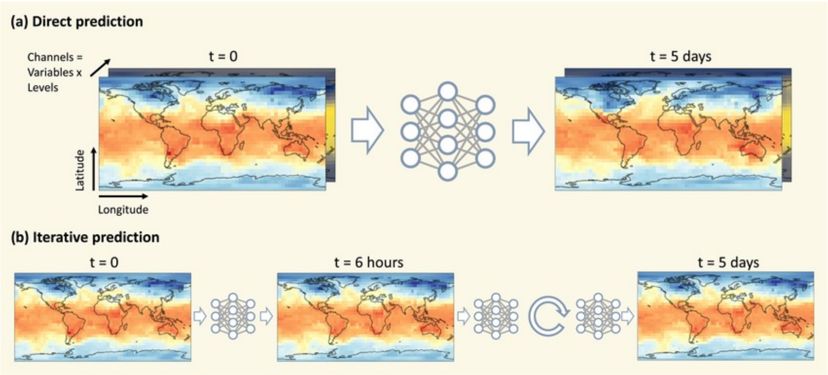
General Circulation Models (GCM) basics

- Built on well-understood **physical principles**
- Defined on a **grid**
- Sub-grid scale processes are **parameterized**
- Run in **ensembles**
- Computational heavy
  - One run would take centuries on a laptop

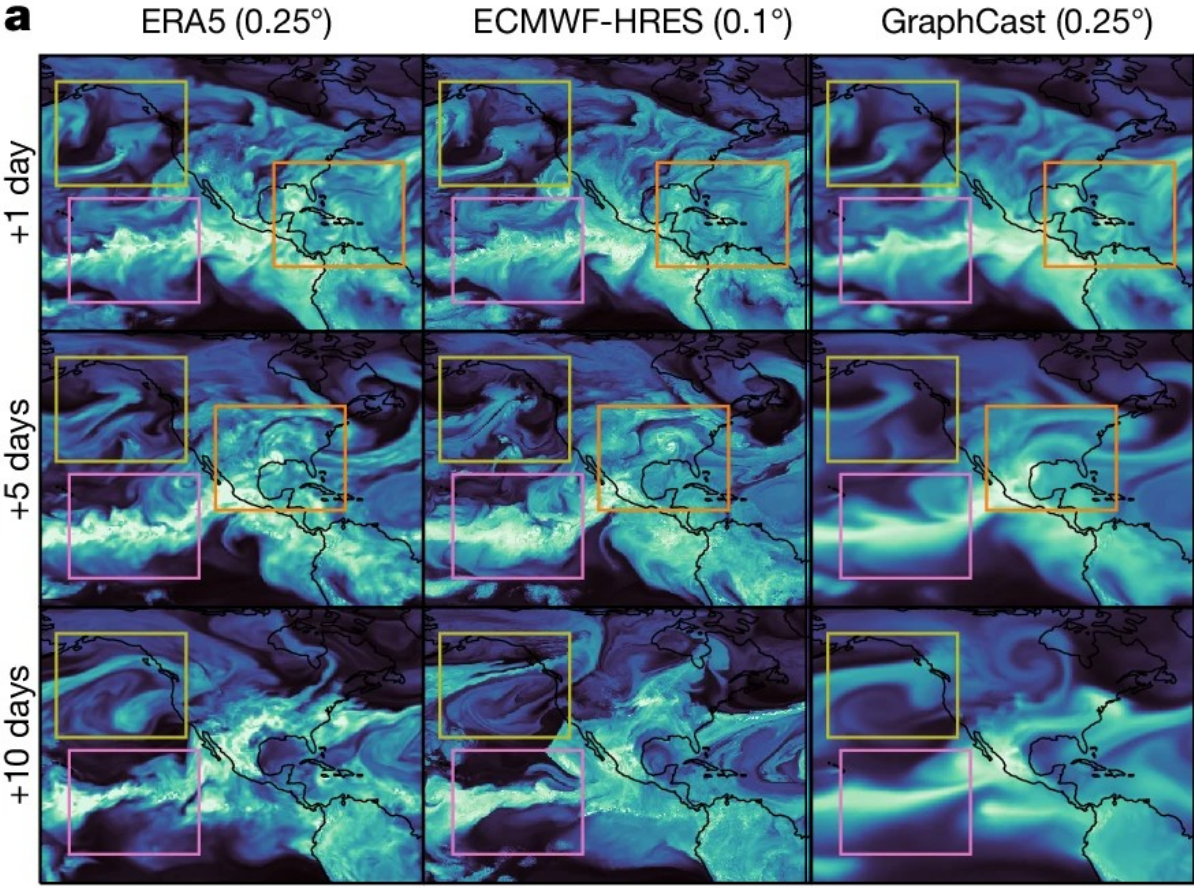


Balaji et al (2022)

Neural Networks for weather prediction



Rasp et al. (2020)

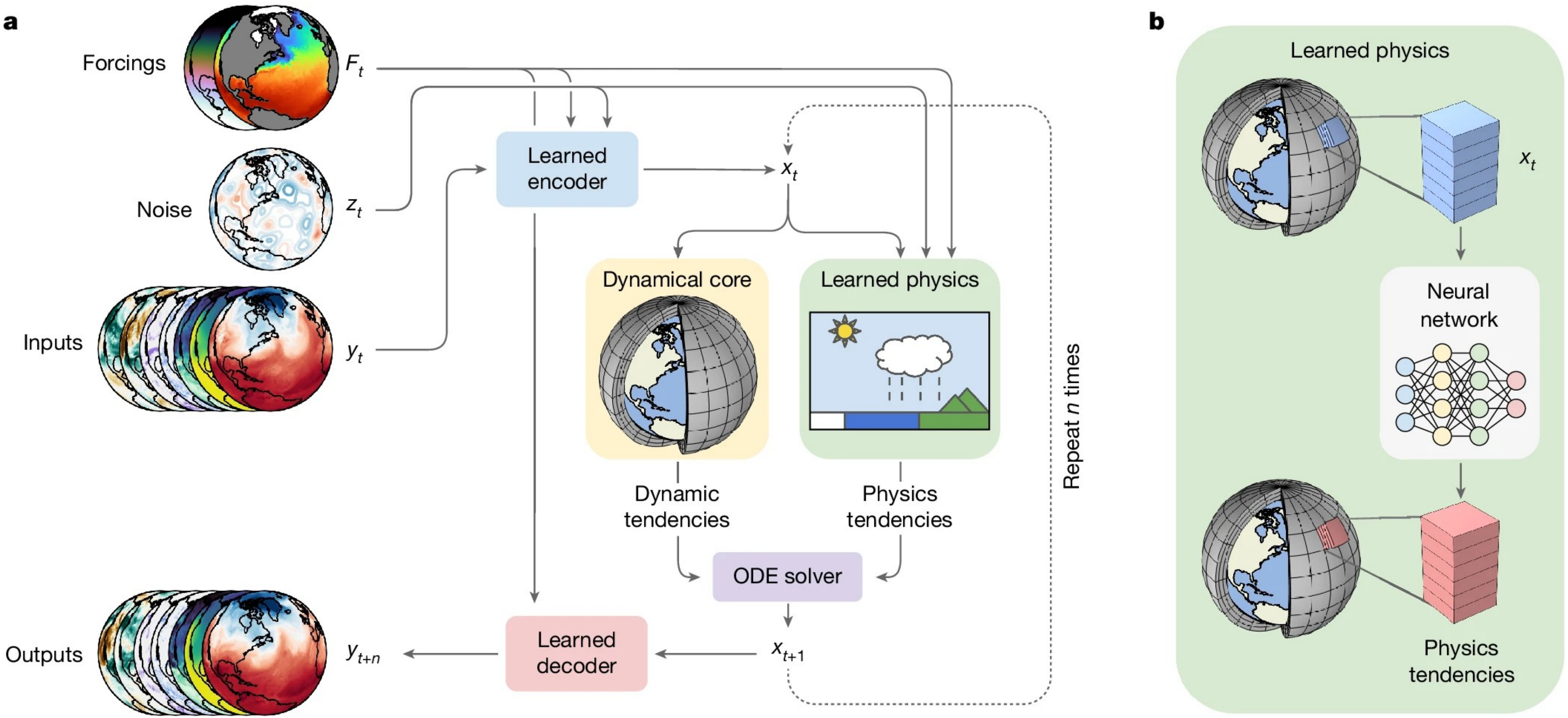


Watt-Meyer et al. (2024)

- Train on reanalysis data
- Computationally cheaper than GCMs
- Good short-time deterministic forecast
- Blurry on longer time scales

Combining the two → NeuralGCM

Watt-Meyer et al. (2024)



Dynamical core

Watt-Meyer et al. (2024)

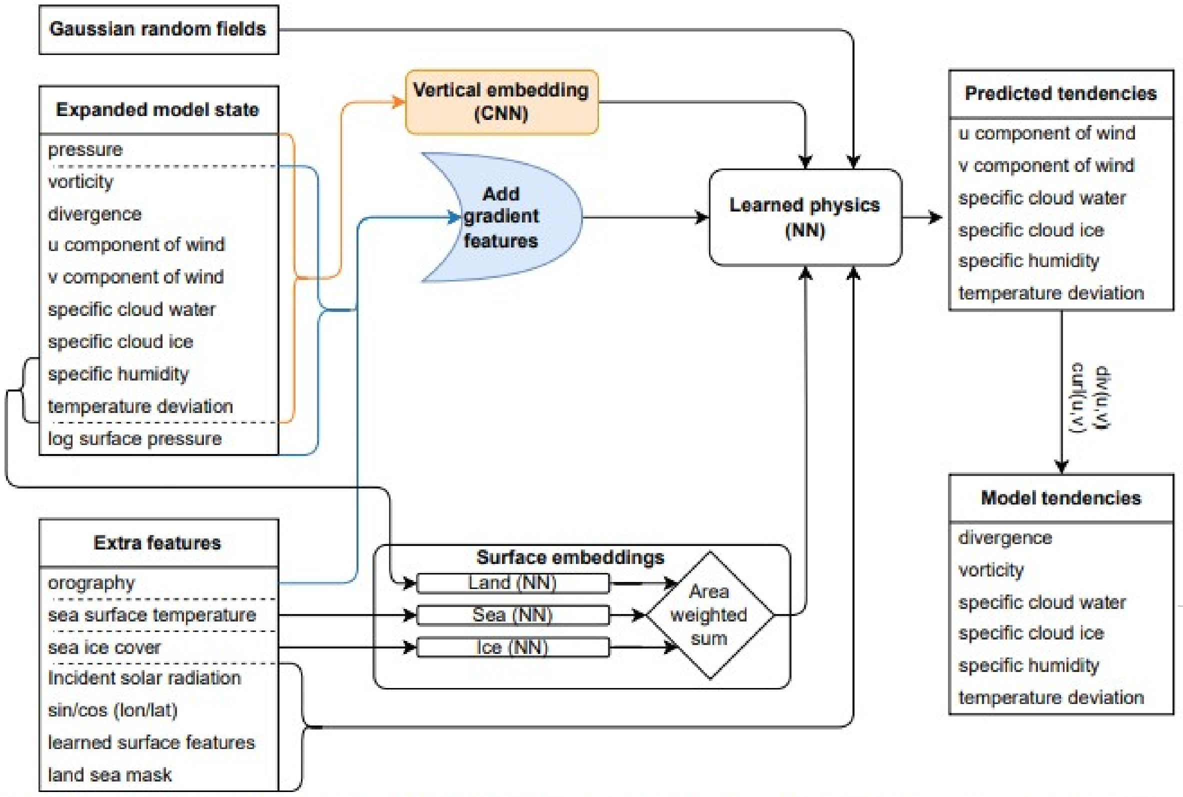
Solve physical equations, which are a combination of
- momentum equations
- the second law of thermodynamics,
- thermodynamic equation of state (ideal gas)
- continuity equation
- hydrostatic approximation

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= -\nabla \times \left((\zeta + f)\mathbf{k} \times \mathbf{u} + \dot{\sigma}\frac{\partial \mathbf{u}}{\partial \sigma} + RT'\nabla \log p_s \right) \\
\frac{\partial \delta}{\partial t} &= -\nabla \cdot \left((\zeta + f)\mathbf{k} \times \mathbf{u} + \dot{\sigma}\frac{\partial \mathbf{u}}{\partial \sigma} + RT'\nabla \log p_s \right) - \nabla^2 \left(\frac{||\mathbf{u}||^2}{2} + \Phi + R\bar{T} \log p_s \right) \\
\frac{\partial T}{\partial t} &= -\mathbf{u} \cdot \nabla T - \dot{\sigma}\frac{\partial T}{\partial \sigma} + \frac{\kappa T \omega}{p} = -\nabla \cdot \mathbf{u}T' + T'\delta - \dot{\sigma}\frac{\partial T}{\partial \sigma} + \frac{\kappa T \omega}{p} \\
\frac{\partial q_i}{\partial t} &= -\nabla \cdot \mathbf{u}q_i + q_i\delta - \dot{\sigma}\frac{\partial q_i}{\partial \sigma} \\
\frac{\partial \log p_s}{\partial t} &= -\frac{1}{p_s}\int_0^1 \nabla \cdot (\mathbf{u}p_s)\, d\sigma = -\int_0^1 (\delta + \mathbf{u} \cdot \nabla \log p_s)\, d\sigma
\end{aligned}
\tag{1}
$$

$$\rightarrow \Phi(\tilde{X})$$

Learned physics

Watt-Meyer et al. (2024)



Supplementary Figure 1: Visualization of the data flow in the learned physics module of NeuralGCM.

$$\rightarrow \quad \Psi(\tilde{X})$$

Combining tendencies and training

Watt-Meyer et al. (2024)

NeuralGCM implements

$$\frac{\partial \tilde{X}}{\partial t} = \Phi(\tilde{X}) + \Psi(\tilde{X}), \quad t_0 < t < t_0 + \tau,$$
$$\tilde{X}(t_0) = \text{Encode}(Y(t_0)).$$

Then finally, $X = \text{Decode}(\tilde{X})$ is evaluated against Y.

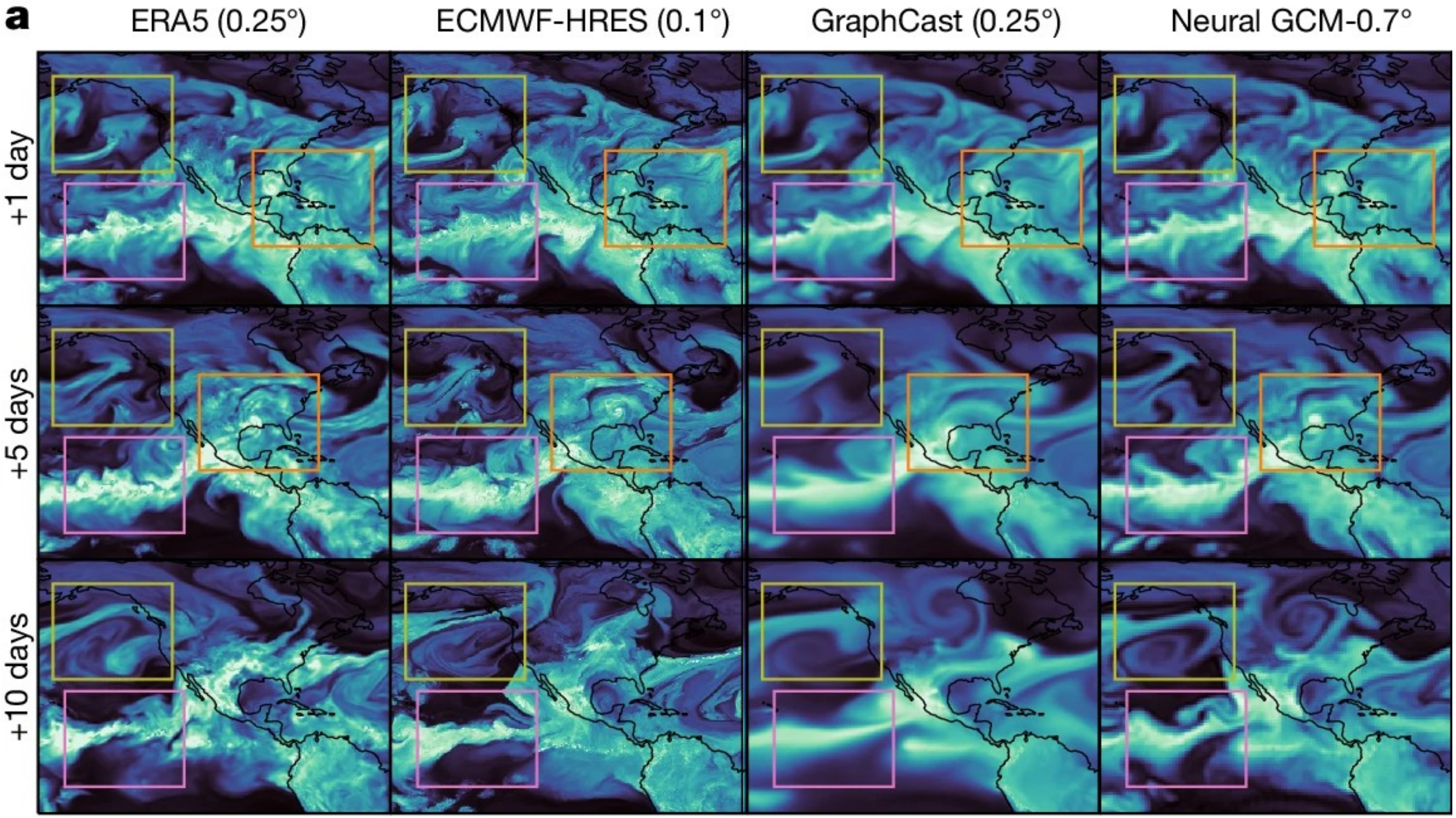
Determenistic forecast → Mean Squared Error (MSE)

Ensamble forecast → Continuous Ranked Probability Score (CRPS)

Finally, use Adam to minimize the loss.

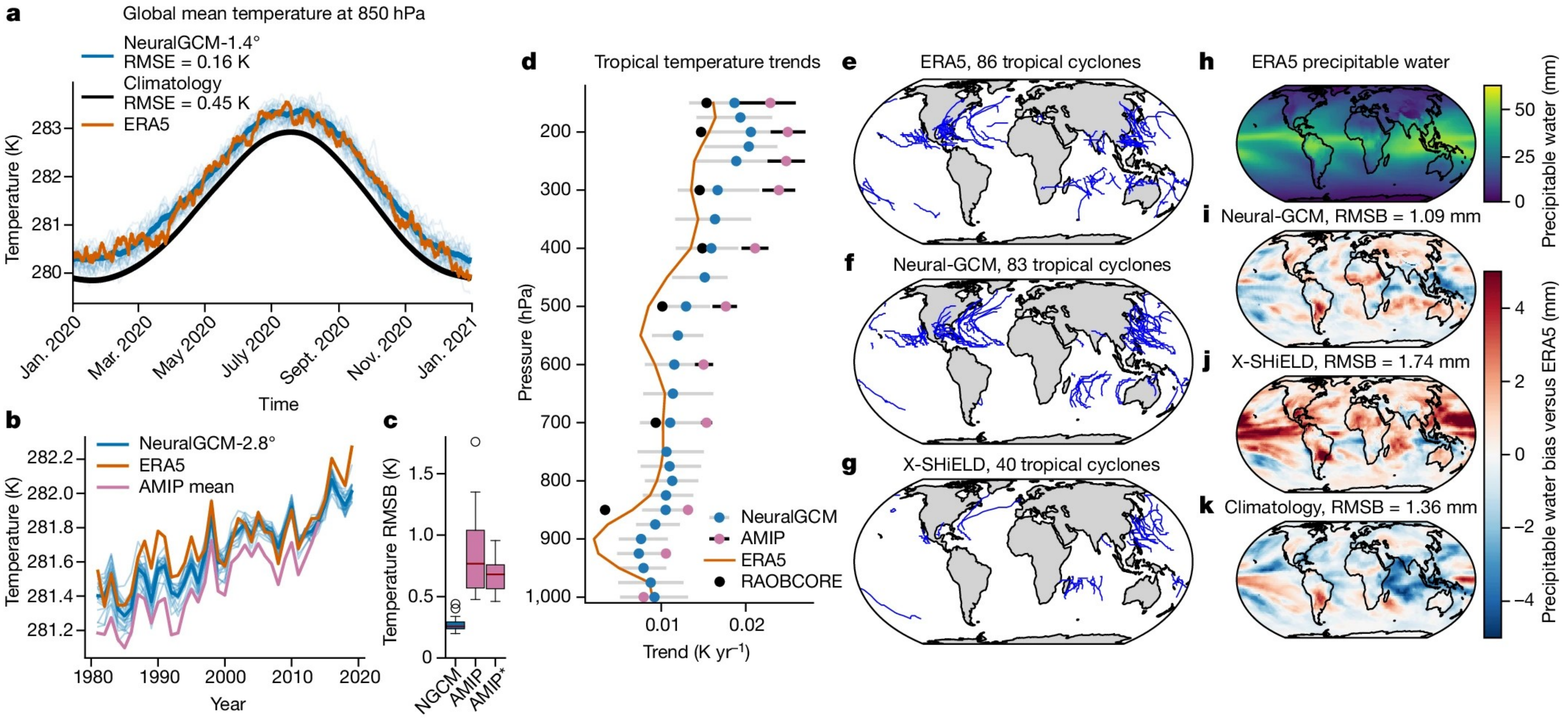
Medium-range weather forecast

Watt-Meyer et al. (2024)



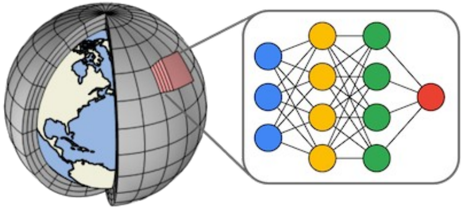
Simulations of climate

Watt-Meyer et al. (2024)

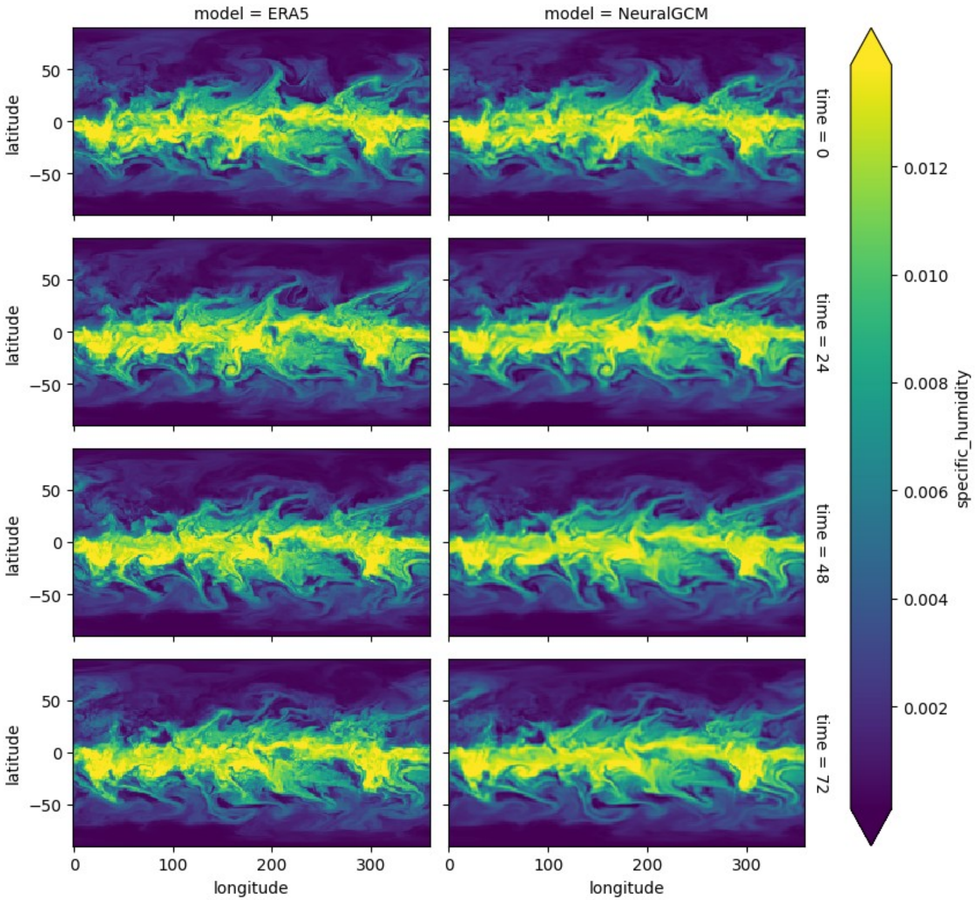


...and you can run it on your own computer*!

python™



NeuralGCM



Loss functions

Watt-Meyer et al. (2024)

$$\begin{aligned}\mathcal{L}_{Deterministic} =& \lambda_{data}\mathcal{M}_{Data} + \lambda_{spec}\mathcal{M}_{DataSpec} + \lambda_{model}\mathcal{M}_{Model} \\ &+ \lambda_{spec}\mathcal{M}_{ModelSpec} + \lambda_{bias}\mathcal{M}_{MSBias},\end{aligned}$$

$$\mathcal{M}_*(\tau) := \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\sum_{\sigma,v}\sum_{\substack{\tau'\in\mathcal{D}\\ \tau'\leq\tau}} \rho_*(X(t\to t+\tau', v, \sigma), Y(t+\tau', v, \sigma))^2,$$

$$\mathcal{L}_{CRPS}(\tau) := \sum_{t\in\mathcal{T}}\sum_{p,v}\sum_{\substack{\tau'\in\mathcal{D}\\ \tau'\leq\tau}} \left[\mathcal{C}_{spectral}(t,\tau',v) + \mathcal{C}_{nodal}(t,\tau',v)\right].$$

$$\begin{aligned}\mathcal{C}_{spectral}(t,p,v,\tau) = \frac{1}{2}\sum_{l,m}\Big[&|X(t\to t+\tau,\ldots,l,m) - Y(t+\tau,\ldots,l,m)| \\ &+ |X'(t\to t+\tau,\ldots,l,m) - Y(t+\tau,\ldots,l,m)| \\ &- |X(t\to t+\tau,\ldots,l,m) - X'(t\to t+\tau,\ldots,l,m)|\Big],\end{aligned}$$