ENSTA

IP PARIS

Article Summary

ClimODE : Climate and weather forecasting with physics-informed neural ODEs

Physics-Informed Neural Networks

Author : Billy Weynans in 2025

Contents

I Introduction

ClimODE [2], a novel spatiotemporal model inspired by the principle of advection from statistical mechanics, where weather evolves through the spatial movement of quantities over time. By learning global weather transport as a neural flow, ClimODE provides accurate, value-conserving forecasts with uncertainty estimation, achieving state-of-the-art performance in global and regional predictions with significantly fewer parameters than existing models.

The results of this method will not be presented in this document as it is the method itself that is of interest. More informations in the ClimODE page

II Advection equation

In the field of physics, engineering, and earth sciences, advection is the transport of a substance or quantity by bulk motion of a fluid. The properties of that substance are carried with it.

$$\underbrace{\frac{\partial u}{\partial t}}_{\text{time evolution } \dot{u}} + \underbrace{\overbrace{\vec{v} \cdot \nabla u}^{\text{transport}} + \overbrace{u \nabla \cdot \vec{v}}^{\text{compression}}}_{\text{advection}} = \underbrace{s}_{\text{sources}}$$

with u being the described quantity and $\vec{v}$ the associated flow velocity. Hence u is what the whole pipeline is meant to predict and $\vec{v}$ is an intermediate vector that respects physical properties and makes the whole pipeline physics-informed.

In our case, the weather is considered as a closed system, therefore the sources s are 0. Hence,

$$\dot{u} = -\vec{v} \cdot \nabla u - u \nabla \cdot \vec{v}$$

In our case, a subsample of the low-resolution ERA5 dataset is used :

- 2m_temperature : Temperature at 2 m above the ground
- 10m_u_component_of_wind : Latitude-wise wind
- 10m_v_component_of_wind : Longitude-wise wind
- geopotential_500 : geopotential height at 500 hPa
- temperature_850 : Temperature at 850 hPa

Plus two constants fields :

- orography
- land-sea mask

III Neural ODE

[1] defines Neural Ordinary Differential Equation (Neural ODE). Frequently in neural networks, there are successive transformations of hidden states h such as :

$$h_{t+1} = h_t + \mathrm{f}(h_t, \theta_t)$$

The idea of Neural ODE is to take smaller steps and make the transformations continuous instead of discrete. Hence we would have :

$$\frac{d\,\mathrm{h}(t)}{dt} = \mathrm{f}(\mathrm{h}(t), t, \theta)$$

With h(0) as the initial value and an ODE solver, a solution h can be computed for arbitrary evaluation points.

This will be used to compute the velocity field $\vec{v}$ with a neural network f_θ :

$$\dot{\vec{v}} = f_\theta(u, \nabla u, \vec{v}, \psi)$$

Hence, giving this ODE system to solve :

$$\begin{bmatrix} u(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} u(t_0) \\ v(t_0) \end{bmatrix} + \int_{t_0}^{t} \begin{bmatrix} \dot{u}(\tau) \\ \dot{\vec{v}}(\tau) \end{bmatrix} d\tau = \begin{bmatrix} u_k(t_0) \\ v_k(t_0) \end{bmatrix} + \int_{t_0}^{t} \begin{bmatrix} -\nabla \cdot (u_k(\tau) v_k(\tau)) \\ f_\theta\big(u(\tau), \nabla u(\tau), v(\tau), \psi_k\big) \end{bmatrix} d\tau,$$

With $u(t_0)$ being given and $v(\vec{t_0})$ being computed through the following optimisation problem :

$$\hat{\vec{v}} = \arg\min_{\vec{v}} \left\{ \|\dot{u} + \vec{v} \cdot \nabla u + u \nabla \cdot \vec{v}\|_2^2 + \alpha \|\vec{v}\| \right\}$$

This optimization problem comes from the advection equation to which a smoothing term is added. For a given time t_0, the spatial and time derivatives are approximated using previous states ($t < t_0$).

IV Emission model

The assumption of a closed system for the advection equation forces the predicted states $u(x,t)$ to be in a value-preserving manifold. Hence, this emission model is added to allow states that better fit the reality of the weather not being a true closed system.

$$u_{\text{obs}}(x,t) \sim \mathcal{N}\left(u(x,t) + \mu(x,t), \sigma^2(x,t)\right), \quad \mu(x,t), \sigma(x,t) = g\big(u(x,t), \psi\big)$$

Hence σ is the uncertainty of the output and μ is a bias that the authors interpret as the day-night cycle.

V Spatio-temporal embeddings

We encode daily and yearly periodicity of time t with trigonometric time embeddings:

$$\psi_{temporal}(t) = \left[\sin(\tfrac{2\pi t}{24}), \quad \cos(\tfrac{2\pi t}{24}), \quad \sin\left(\tfrac{2\pi t}{365}\right), \quad \cos\left(\tfrac{2\pi t}{365}\right)\right] \tag{7}$$

We encode latitude h and longitude w with trigonometric and spherical-position encodings:

$$\psi_{spatial}(x) = \left[\sin(h), \quad \cos(h), \quad \sin(w), \quad \cos(w), \quad \sin(h)\cos(w), \quad \sin(h)\sin(w)\right] \tag{8}$$

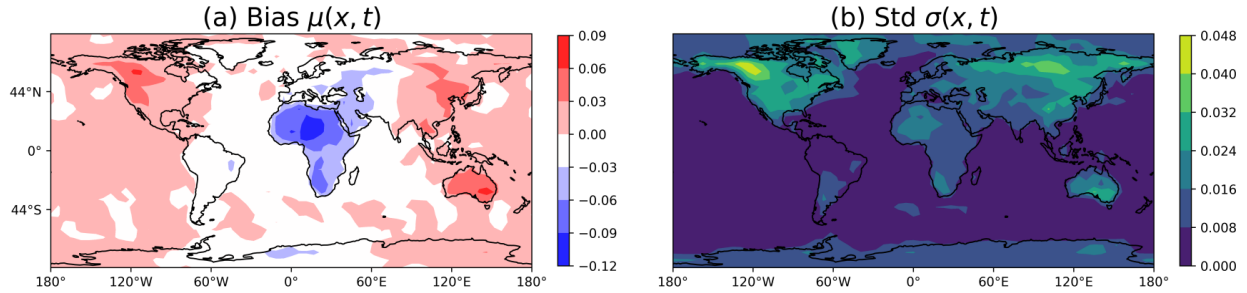

Figure 8: **Effect of emission model**: Global bias and standard deviation maps at 12:00 AM UTC. The bias explains day-night cycle **(a)**, while uncertainty is highest on land, and in north **(b)**.

Figure 1: Enter Caption

$$\psi_{constants}(x) = \big[\text{longitude}, \quad \text{latitude}, \quad \text{land-sea mask}, \quad \text{orography}\big] \tag{9}$$

$$\psi(x,t) = \big[\psi_{temporal}(t), \quad \psi_{spatial}(x), \quad \psi_{temporal}(t) \times \psi_{spatial}(x), \quad \psi_{constants}(x)\big]$$

The spatio-temporal features ψ are additional input channels to the neural networks.

VI Training objective

Let $D = (y_1, \ldots, y_N), y_i \in R^{K \times H \times W}$ be the N weather states at timepoints t_i. We minimize the negative log-likelihood of the observations y_i:

$$\mathcal{L}(\theta; D) = -\frac{1}{NKHW} \sum_{i=1}^{N} \left[\log \mathcal{N}\left(y_i \mid u(t_i) + \mu(t_i), \operatorname{diag} \sigma^2(t_i)\right) + \log \mathcal{N}^{+}\left(\sigma(t_i) \mid 0, \lambda_\sigma^2 I\right)\right] \tag{10}$$

to which a Gaussian prior if added for the uncertainties $\sigma(t_i)$ with a hypervariance λ_σ to prevent variance explosion during training.λ_σ^{-1} is decayed using cosine annealing during training to remove its effects and arrive at a maximum likelihood estimate.

VII Paper code utilisation

The initial GitHub is here. As I plan to use this method, I worked a lot on the code but could not create visualisations in time. Here's my current version.

References

[1] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

[2] Yogesh Verma, Markus Heinonen, and Vikas Garg. ClimODE: Climate forecasting with physics-informed neural ODEs. In *The Twelfth International Conference on Learning Representations*, 2024.