# ClimODE: Climate Forecasting With Physics-informed Neural ODEs

Aymeric Delefosse[1] Mathis Koroglu[1] Charles Vin[1] | Yogesh Verma[2] Markus Heinonen[2] Vikas Garg[2]

[1]Reviewers [2]Original authors

## Problem Definition and Contribution

**Goal:** Enhancing climate forecasting by integrating physics-informed neural ordinary differential equations (ODEs) with uncertainty quantification.
**Motivations:** Existing models neglect the underlying physics, lack of uncertainty quantification and are computationally intensive.

## Problem Formulation

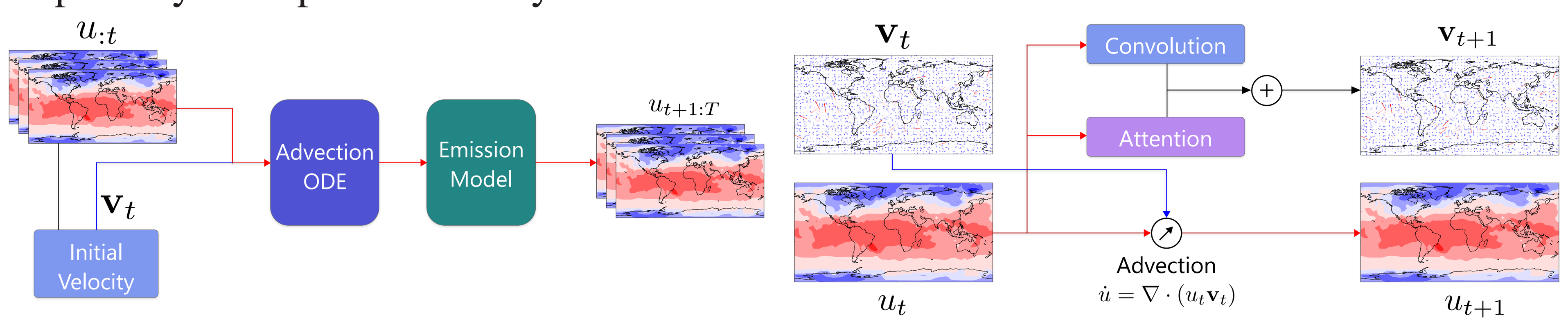
**Statistical Mechanics:** Weather can be described as a spatial movement of quantities over time, governed by the partial differential continuity equation:

$$\underbrace{\frac{du}{dt}}_{\text{time evolution } \dot{u}} + \underbrace{\overbrace{\mathbf{v}\cdot\nabla u}^{\text{transport}} + \overbrace{u\nabla\cdot\mathbf{v}}^{\text{compression}}}_{\text{advection}} = \underbrace{s}_{\text{sources}},$$

where $u(x,t)$ is a quantity evolving over space $\mathbf{x}$ and time $t$ driven by a flow's velocity $\mathbf{v}(\mathbf{x},t)$.
**Main Idea:** We solve the continuity equation over entire Earth as a system of neural ODEs.

## Method

**Network Architecture:** Spatiotemporal embedding $\psi$ encodes time and location information to capture cyclical patterns of day and season.




(a) ClimODE (b) Advection ODE

**Initial Velocity Inference:**

$$\hat{\mathbf{v}}_k(t) = \arg\min_{\hat{\mathbf{v}}_k(t)} \left\{ \left\|\tilde{\dot{u}}_k(t) + \mathbf{v}_k(t)\cdot\tilde{\nabla}u_k(t) + u_k(t)\tilde{\nabla}\cdot\mathbf{v}_k(\mathbf{x},t)\right\|_2^2 + \alpha\left\|\mathbf{v}_k(t)\right\| \right\}$$

**Advection ODE:** Modeling the dynamic evolution of quantities at specific locations by transforming second-order PDEs into first-order ODEs:

$$\begin{bmatrix}\mathbf{u}(t)\\ \mathbf{v}(t)\end{bmatrix} = \begin{bmatrix}\{u_k(t_0)\}_k\\ \{\mathbf{v}_k(t_0)\}_k\end{bmatrix} + \int_{t_0}^{t} \begin{bmatrix}\{-\nabla\cdot(u_k(\tau)\mathbf{v}_k(\tau))\}_k\\ \{f_\theta(\mathbf{u}(\tau),\nabla\mathbf{u}(\tau),\tau(t),\psi)_k\}_k\end{bmatrix} d\tau$$

**Flow Velocity:** Modeling local and global effects using a hybrid network:

$$\dot{\mathbf{v}}_k(\mathbf{x},t) = f_{\text{conv}}\left(\mathbf{u}(t),\nabla\mathbf{u}(t),\mathbf{v}(t),\psi\right) + \gamma f_{\text{att}}\left(\mathbf{u}(t),\nabla\mathbf{u}(t),\mathbf{v}(t),\psi\right)$$

**Emission Model:** Accounting for uncertainty and value changes in climate estimates:

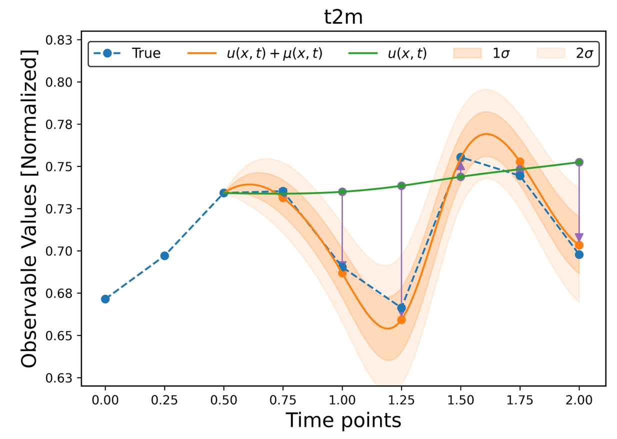
$$u_k^{\text{obs}}(\mathbf{x},t) \sim \mathcal{N}\left(u_k(\mathbf{x},t)+\mu_k(\mathbf{x},t),\sigma_k^2(\mathbf{x},t)\right), \quad \mu_k(\mathbf{x},t),\sigma_k(\mathbf{x},t) = g_k\left(\mathbf{u}(\mathbf{x},t),\psi\right)$$

**Loss Function:** Gaussian negative log-likelihood:

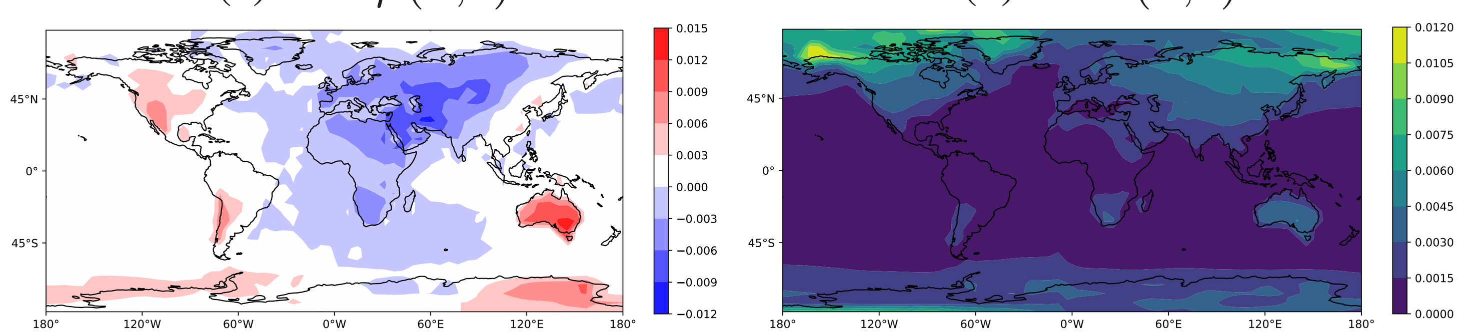
$$\mathcal{L}_\theta = -\frac{1}{NKHW}\sum_{i=1}^{N}\left(\log\mathcal{N}\left(\mathbf{y}_i|\mathbf{u}(t_i)+\boldsymbol{\mu}(t_i),\operatorname{diag}\boldsymbol{\sigma}^2(t_i)\right) + \log\mathcal{N}_+\left(\boldsymbol{\sigma}(t_i)|\mathbf{0},\lambda_\sigma^2 I\right)\right)$$

## Experiments & Results

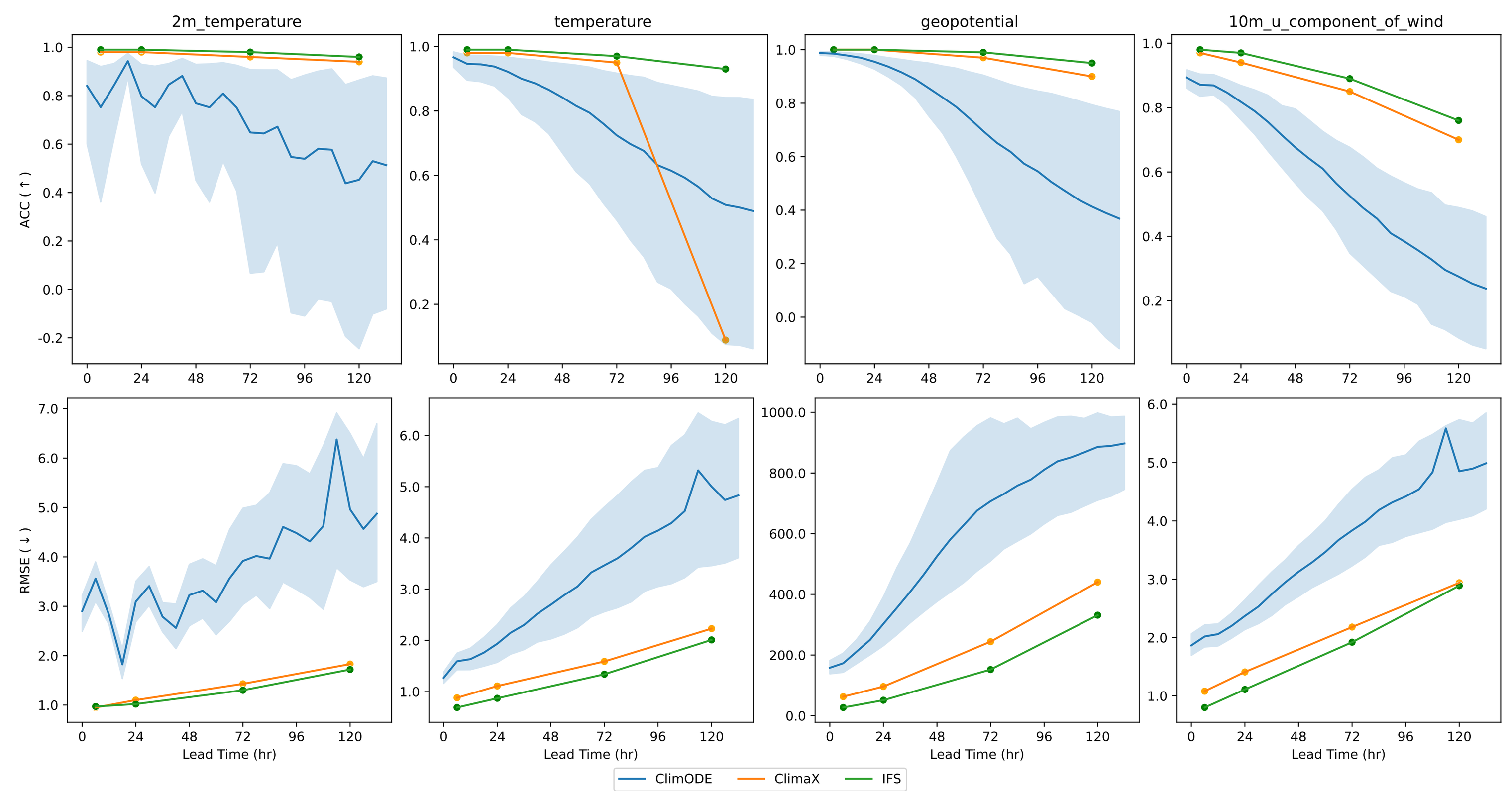
### Effect of Emission Model:




(a) Bias $\mu(\mathbf{x},t)$ (b) Std $\sigma(\mathbf{x},t)$



### Quantitative Results and Comparison on ERA5:

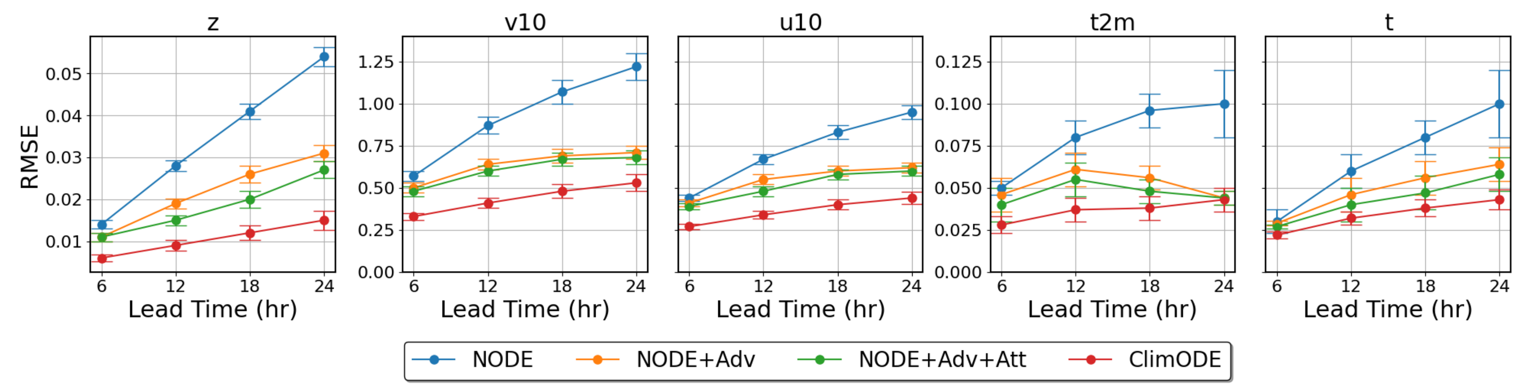



### Effect of ODE Solver (per batch, on a single A100):

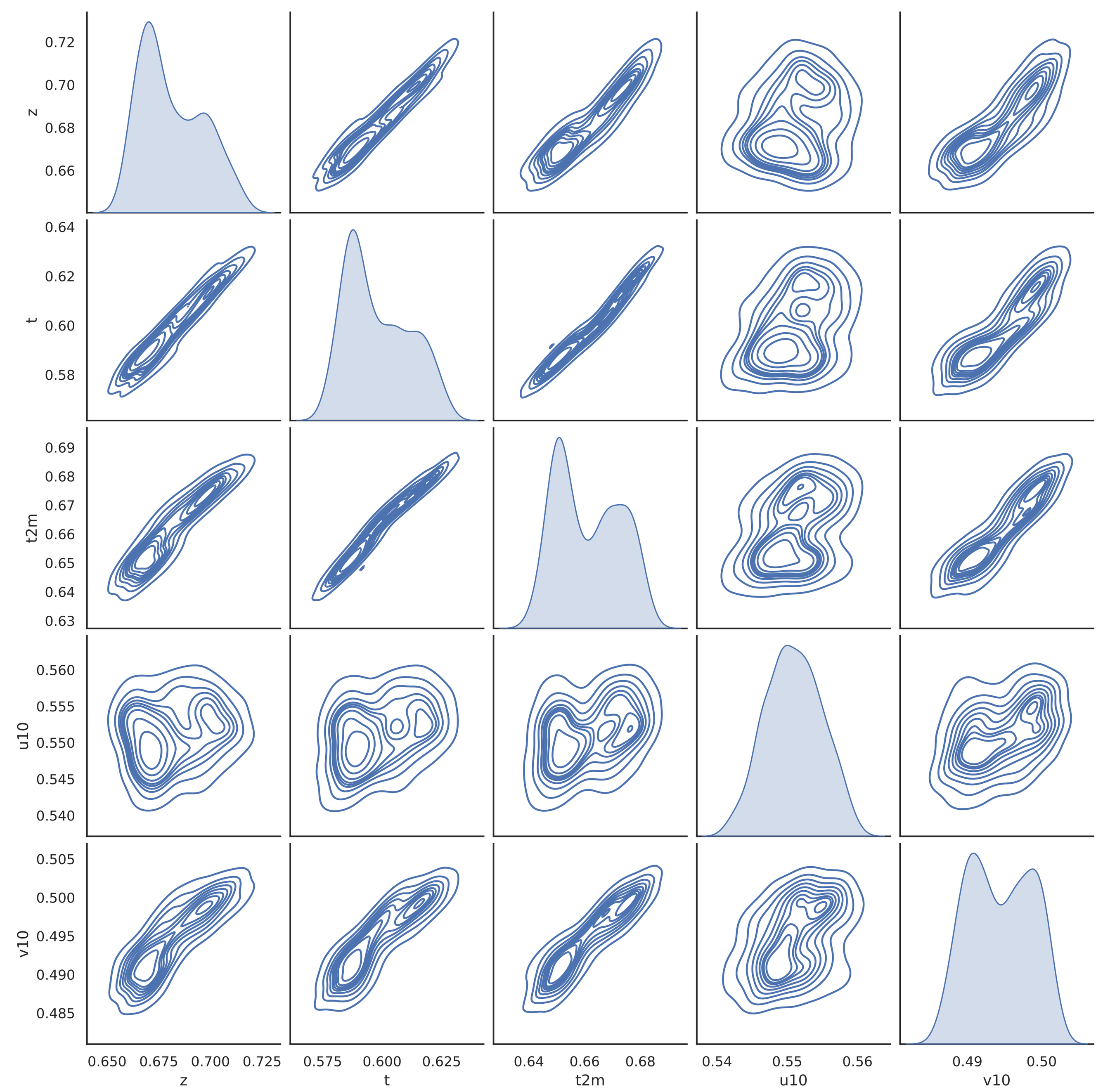
| Solver | Euler | | RK4 | | Implicit Adams | |
|---|---|---|---|---|---|---|
| Adjoints | w | w/o | w | w/o | w | w/o |
| Time (s) | 0.27 | 0.18 | 0.34 | 0.31 | 0.69 | 0.45 |
| VRAM (Gb) | 10.2 | 15.7 | 11.8 | 17.8 | 13.7 | 30.1 |

**Number of parameters:** 2,752,099

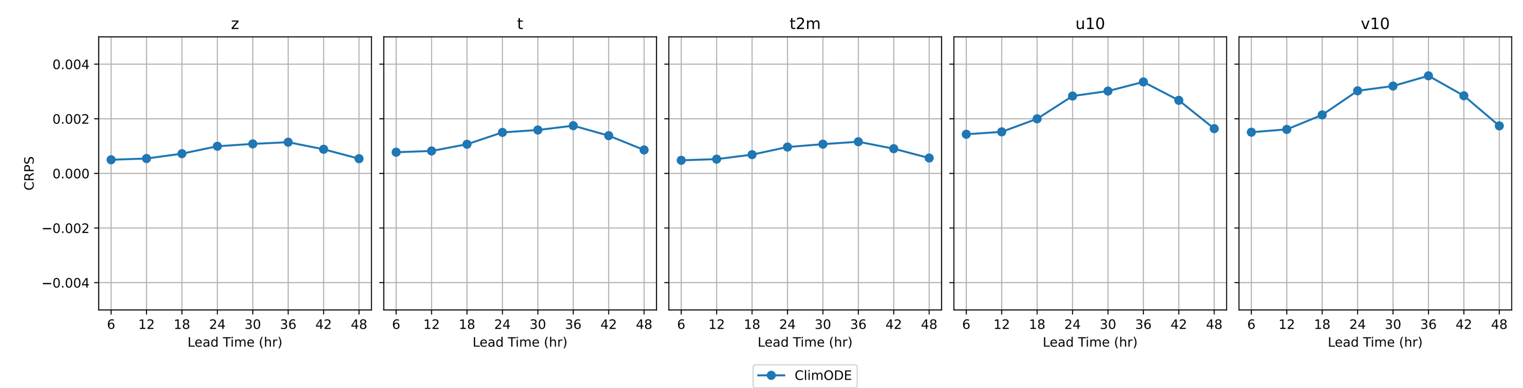
### Effect of Individual Components:




### Quantifying Climate Interdependencies:




### Validity of Mass Conservation:




## Critics

**Paper:**
- Lacks comparison to other models, such as FourCastNet, FuXi or Pangu-Weather (WeatherBench2)
- Is the comparison against ClimaX fair?
- Benchmark only up to 36 hours while state-of-the-art methods reported results for up to 10 days ahead
- Why choose ResNet over U-Net in the Emission Model?

**Implementation (original code):**
- Concerns about data leakage
- Absence of shuffling during training: impact?
- Unclear handling of residual architecture and dropout
- Code quality critique: Concerning coding standards
- Using Euler scheme to solve the ODE is known to be unstable but is a good compromise in terms of accuracy and efficiency