ClimODE: Climate and Weather Forecasting with Physics-informed Neural ODEs

Yogesh Verma

Aalto University

Markus Heinonen

Aalto University

Vikas Garg

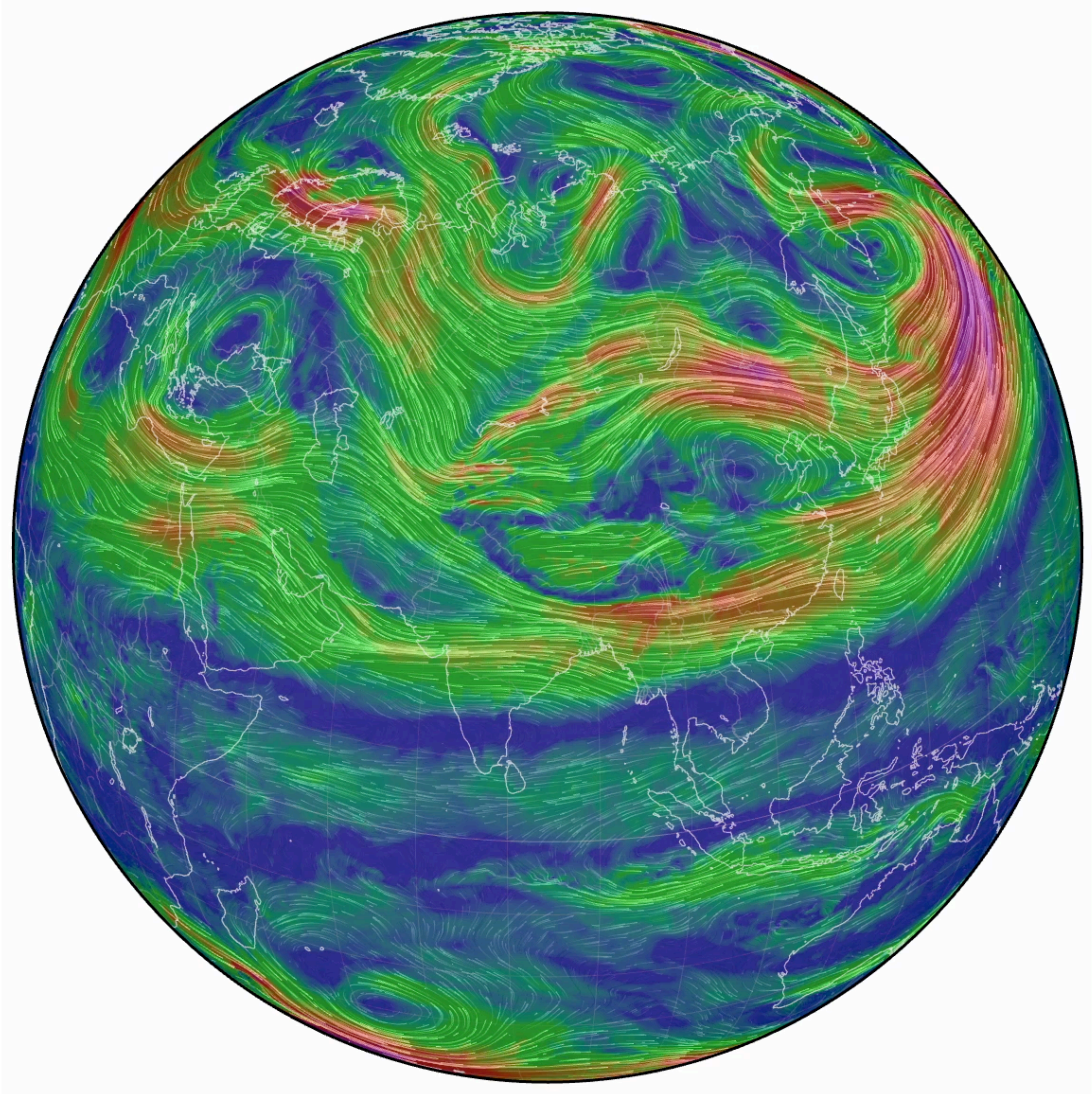
Aalto University
YaiYai Ltd

Twelfth International Conference on Learning Representations (ICLR 2024)

Problem: Forecast Weather

$$\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} u_1(\mathbf{x}, t) \\ \vdots \\ u_K(\mathbf{x}, t) \end{pmatrix} \begin{matrix} \text{Temperature} \\ \vdots \\ \text{Precipitation} \end{matrix}$$

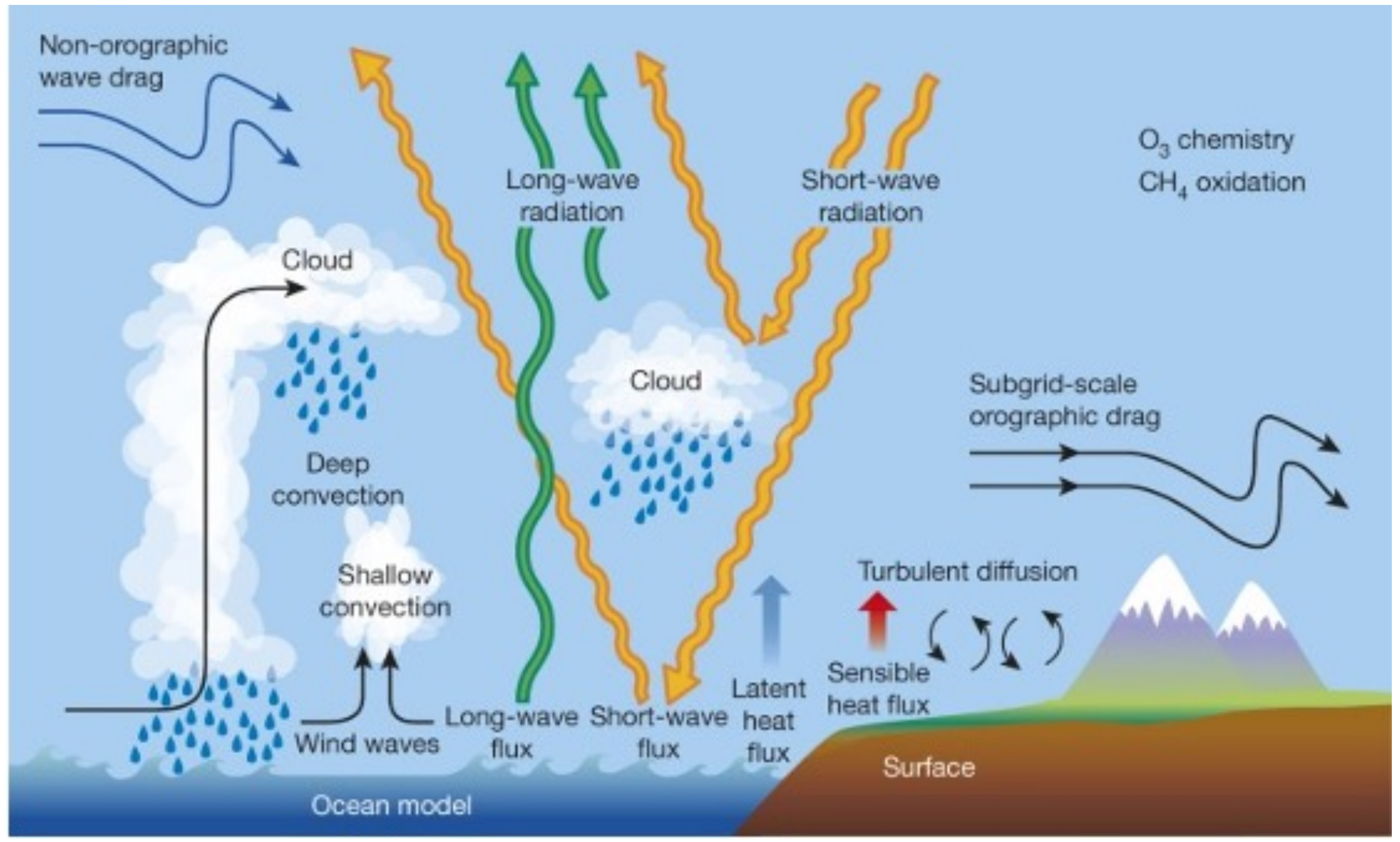
$$\mathbf{x} \in \mathbb{R}^3, t \in \mathbb{R}$$



How to model weather?

- ❌ Complex
- ❌ Expensive
- ✅ Accurate
- ✅ Reliable

Physics



Governing Differential Equations

$$\frac{d}{dt}\mathbf{v} = -2\boldsymbol{\Omega} \times \mathbf{v} - \frac{1}{\rho}\nabla_3 p + \mathbf{g} + \mathbf{F}$$

$$C_v \frac{d}{dt}(\rho q) + p\frac{d}{dt}\left(\frac{1}{\rho}\right) = J$$

$$\frac{\partial}{\partial t}(\rho) = -\nabla_3 \cdot (\rho \mathbf{v})$$

$$\frac{\partial}{\partial t} = -\nabla_3 \cdot (\rho \mathbf{v} q) + \rho\,(E - C)$$

$$p = \rho R T$$

Discrete Numerics

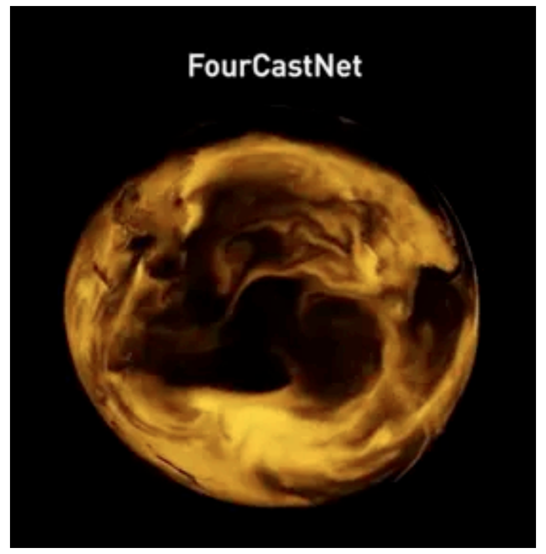
Source: Bauer et al. The quiet revolution of numerical weather prediction. Nature 2015

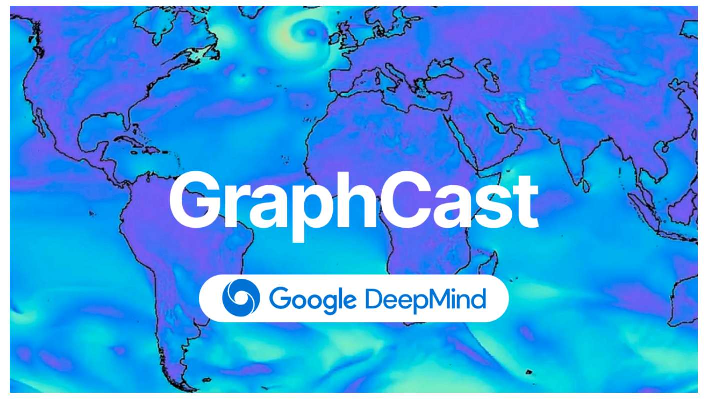
Rise of machine learning

Numerous ML methods like FourCastNet (NVIDIA), GraphCast (DeepMind), ClimaX (Microsoft), etc, forecast weather and have surpassed IFS (conventional physics method). But:

Previous Methods

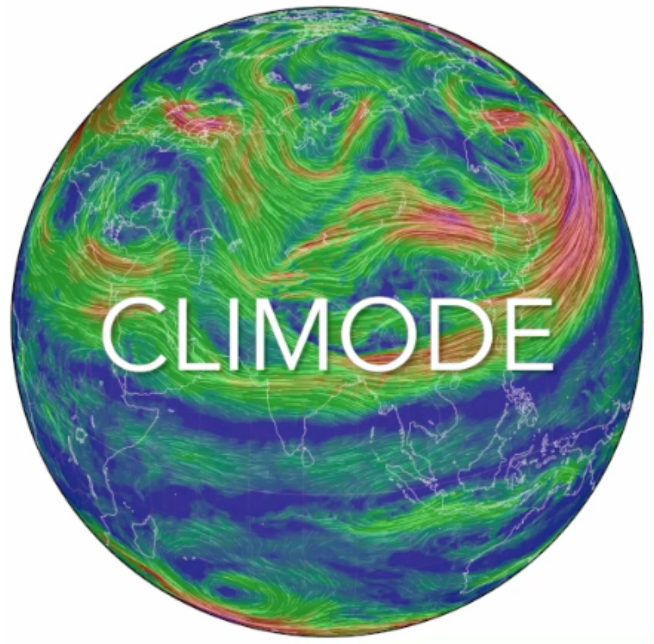
NVIDIA





Microsoft

This work



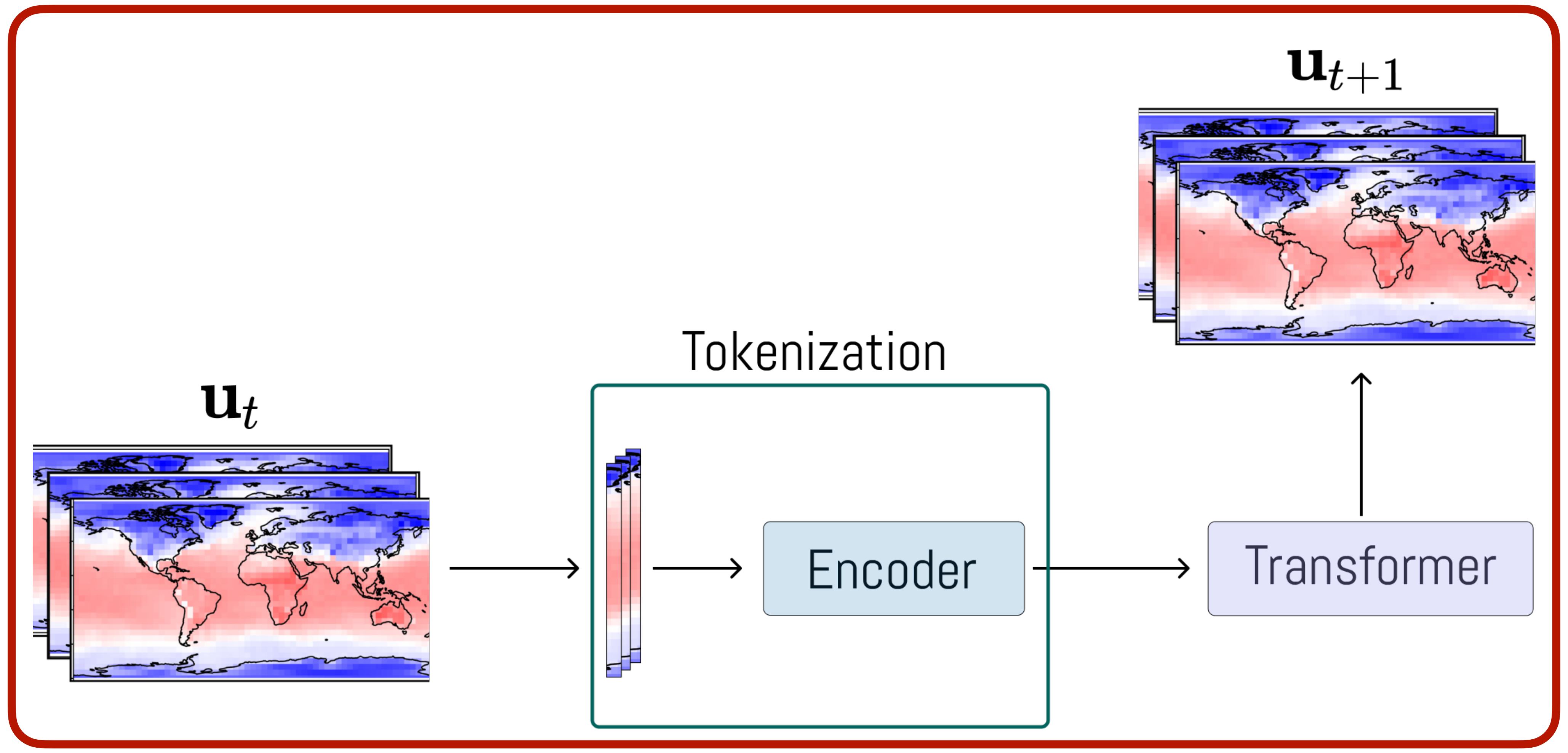
- **Do they produce physically consistent forecasts?**
- **Are they compact?**
- **Do they follow the underlying physics?**

Method	**Value preserving**	**Explicit Periodicity/ Seasonality**	**Uncertainty**	**Continuous time**	**Parameters (M)**
FourCastNet	✗	✗	✗	✗	N/A
GraphCast	✗	✗	✗	✗	37
Pangu-Weather	✗	✗	✗	✗	256
ClimaX	✗	✗	✗	✗	107
NowcastNet	✓	✗	✗	✗	N/A
ClimODE (ours)	✓	✓	✓	✓	**2.8**

Issues with black-box modeling approaches

Black box methods based on Transformers, UNets, GNNs, etc. overlook the fundamental **physical dynamics** and **continuous time nature** of weather.

Vision Transformer



+ AR nature

$$\mathbf{u}_{t+1} = f(\mathbf{u}_t)$$

→

Physics ✗

Cont. time ✗

Compact ✗

Neural PDEs as modeling choice

Neural PDEs do not include any physical dynamics, but gives the solution for continuous time.

$$\dot{u}(x,t) := \frac{\partial u(x,t)}{\partial t} = F\left(x, u, \nabla_x u, \nabla_x^2 u, \ldots\right) \longrightarrow$$

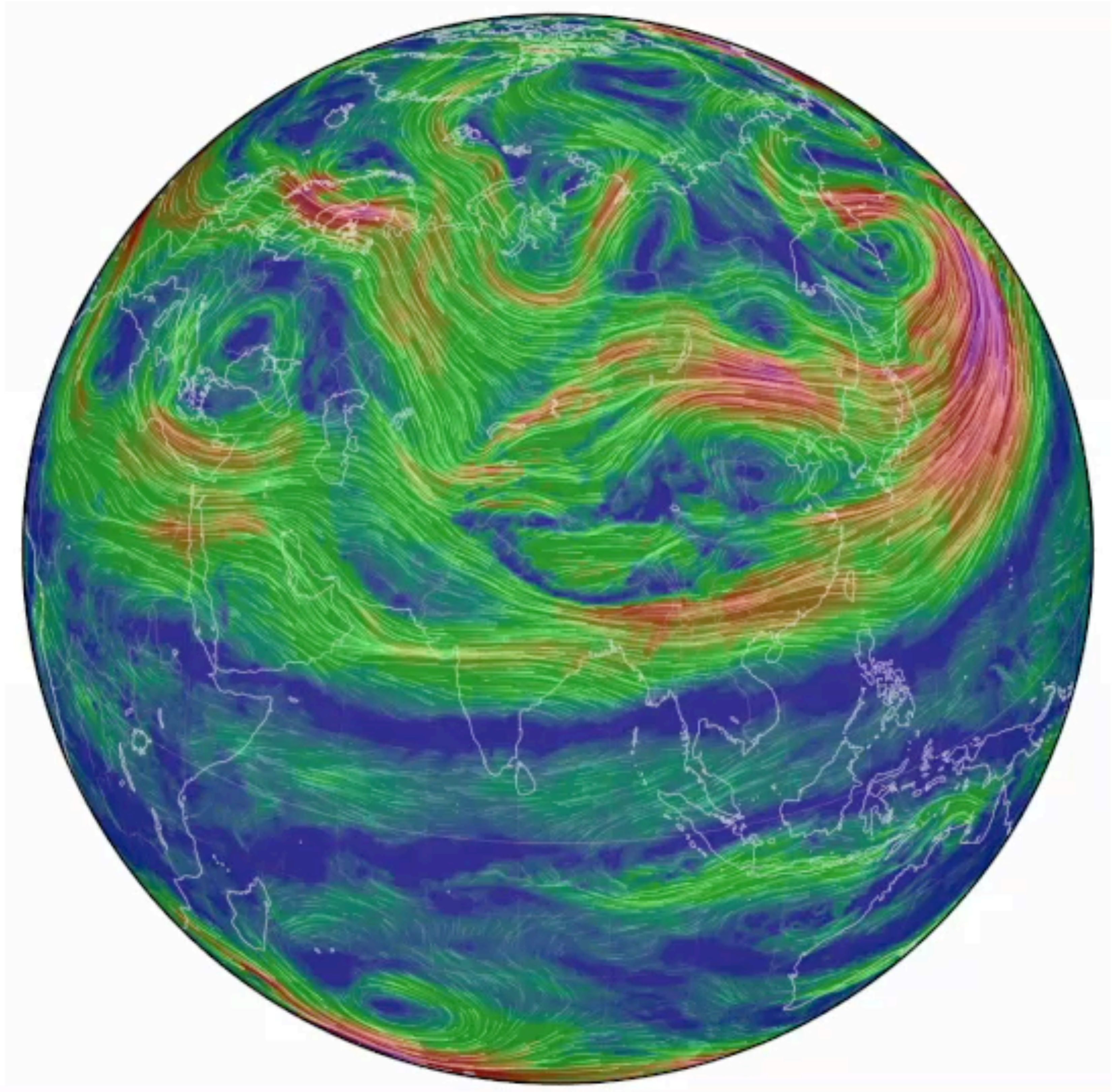
Physics ✗

Cont. time ✓

Compact ✗

Contributions

Develop a **continuous-time** model (neural ODEs/PDEs) that **(i) follows the underlying physics** and **(ii) is compact.**



Physics ✓

Cont. time ✓

Compact ✓

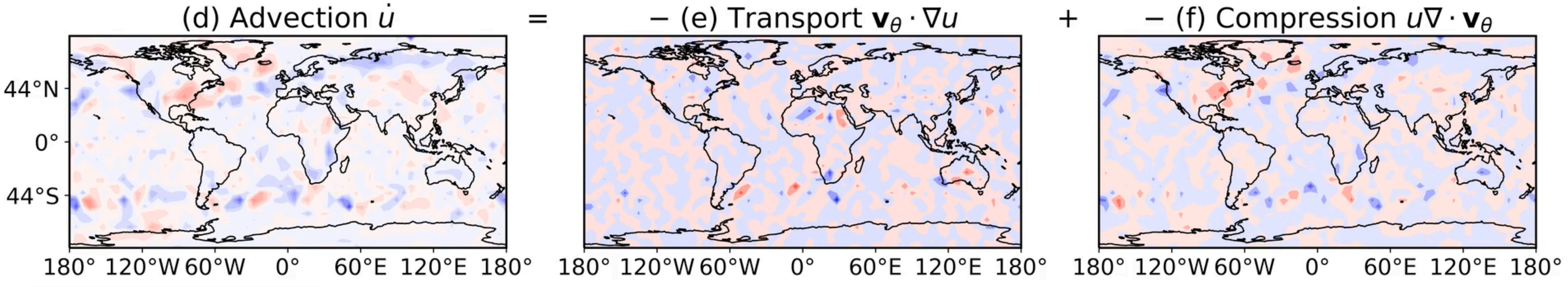
A look at physics: Advection PDE

Weather can be described as a spatial movement of quantities over time. An approach to model the **movement** of a quantity is **advection**.

Transport | Compression

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u + u \nabla \cdot \mathbf{v} = 0$$

Time evolution $\dot{u}$



How to model weather using advection equation? 🤔

Weather advection

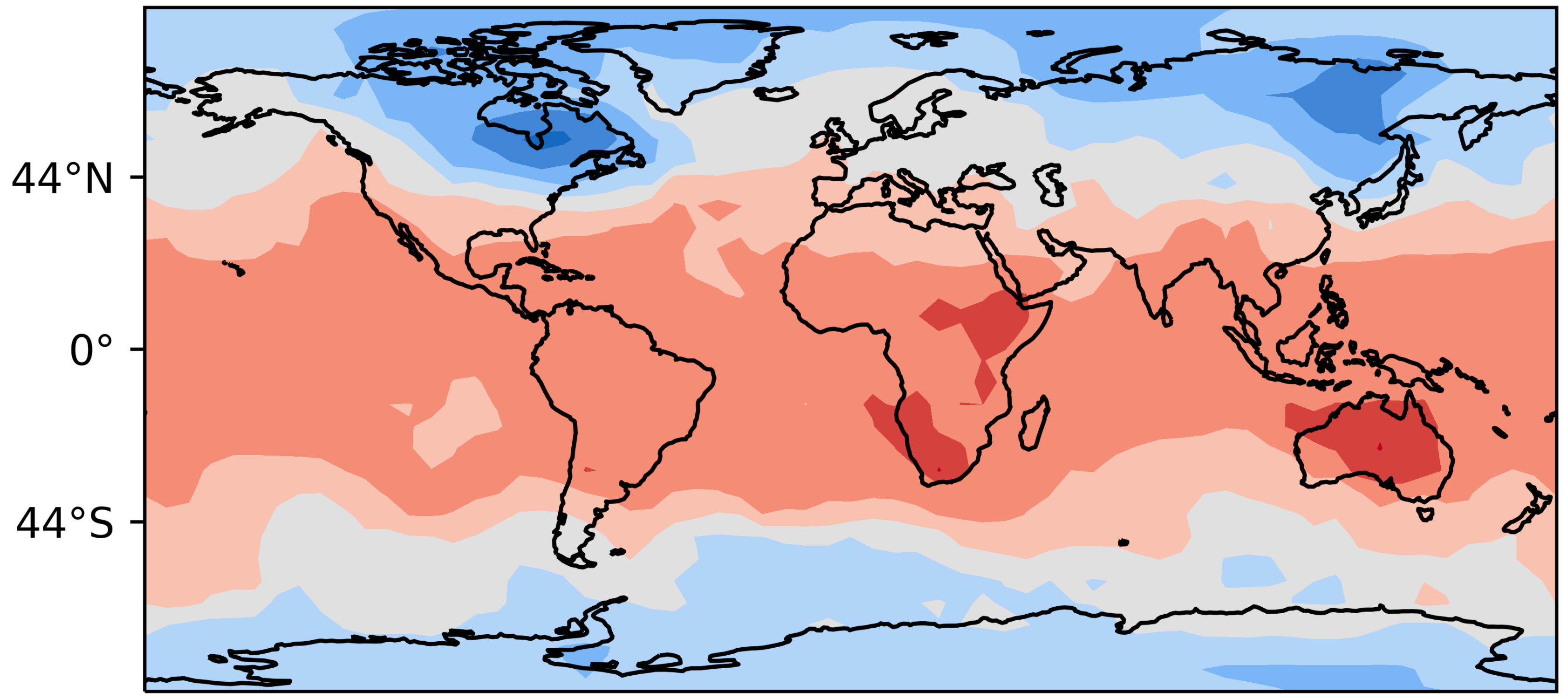
We model weather as a spatiotemporal process $\mathbf{u}(x,t) = (u_1(x,t), \dots, u_K(x,t))$ of K quantities, $u_k(x,t) \in \mathbb{R}$, and assume process follows an advection PDE as

$\dot{u}_k(x,t)$

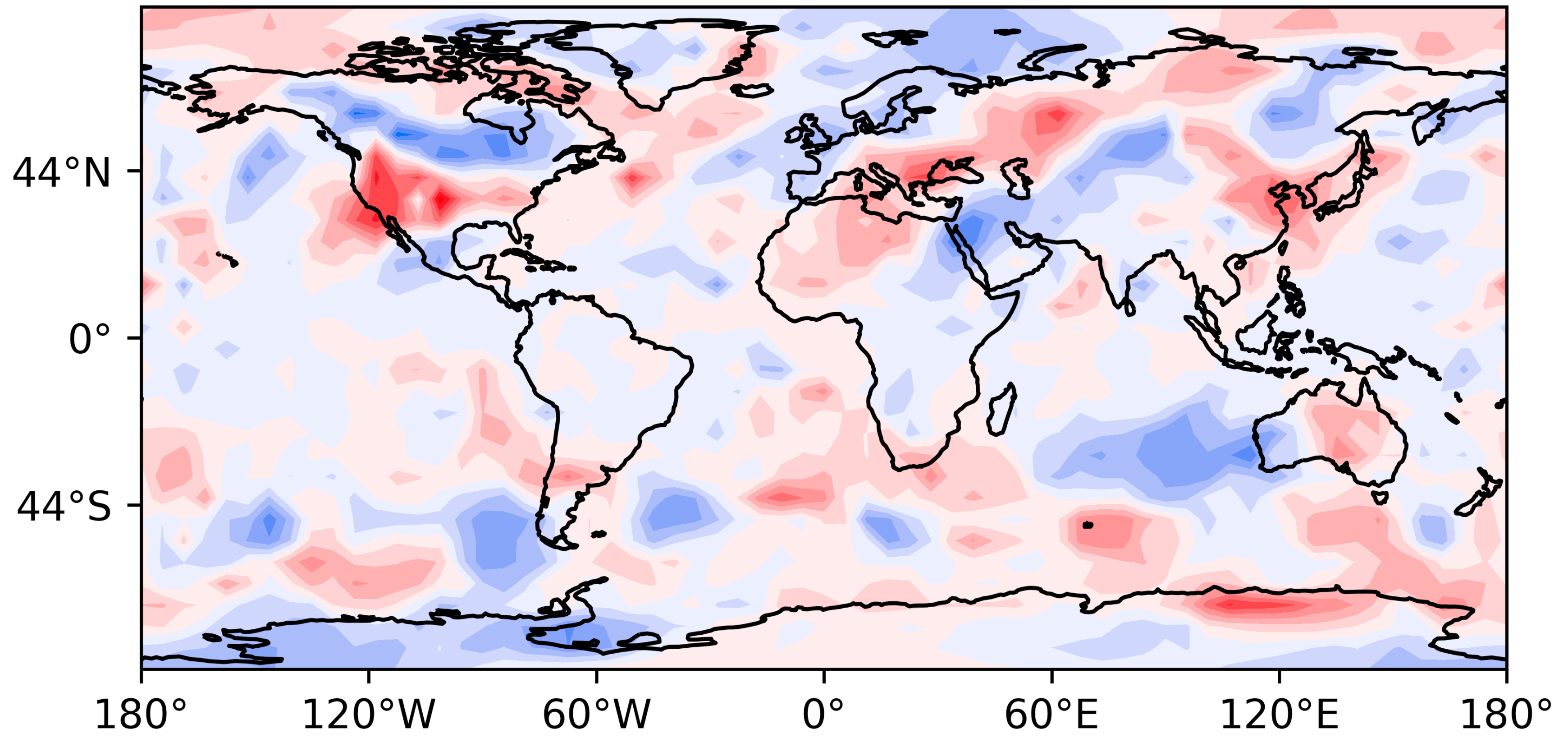
State

Change

(a) State u



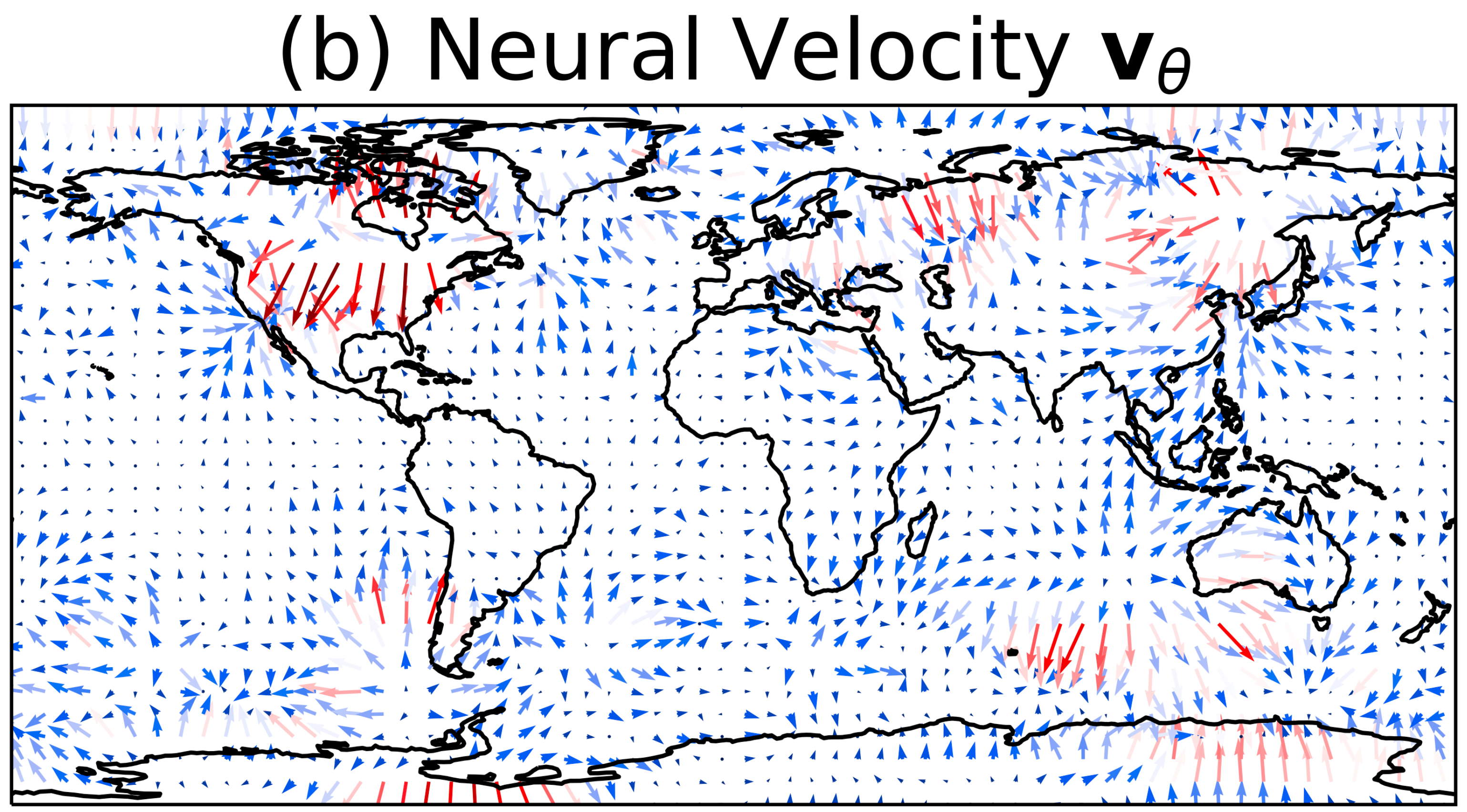
(d) Advection $\dot{u}$

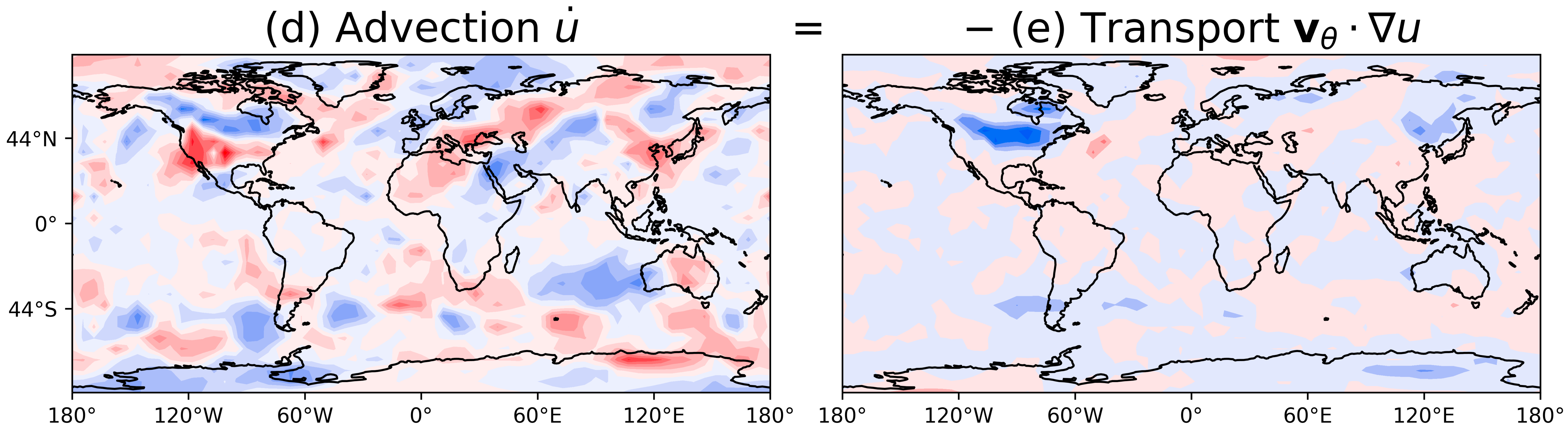


Weather advection

We model weather as a spatiotemporal process $\mathbf{u}(x, t) = (u_1(x, t), \ldots, u_K(x, t))$ of K quantities, $u_k(x, t) \in \mathbb{R}$, and assume process follows an advection PDE as,

$$\dot{u}_k(x, t) = -\mathbf{v}_k(x, t) \cdot \nabla u_k(x, t)$$





Weather advection

We model weather as a spatiotemporal process $\mathbf{u}(x, t) = (u_1(x, t), \ldots, u_K(x, t))$ of K quantities, $u_k(x, t) \in \mathrm{R}$, and assume process follows an advection PDE as,

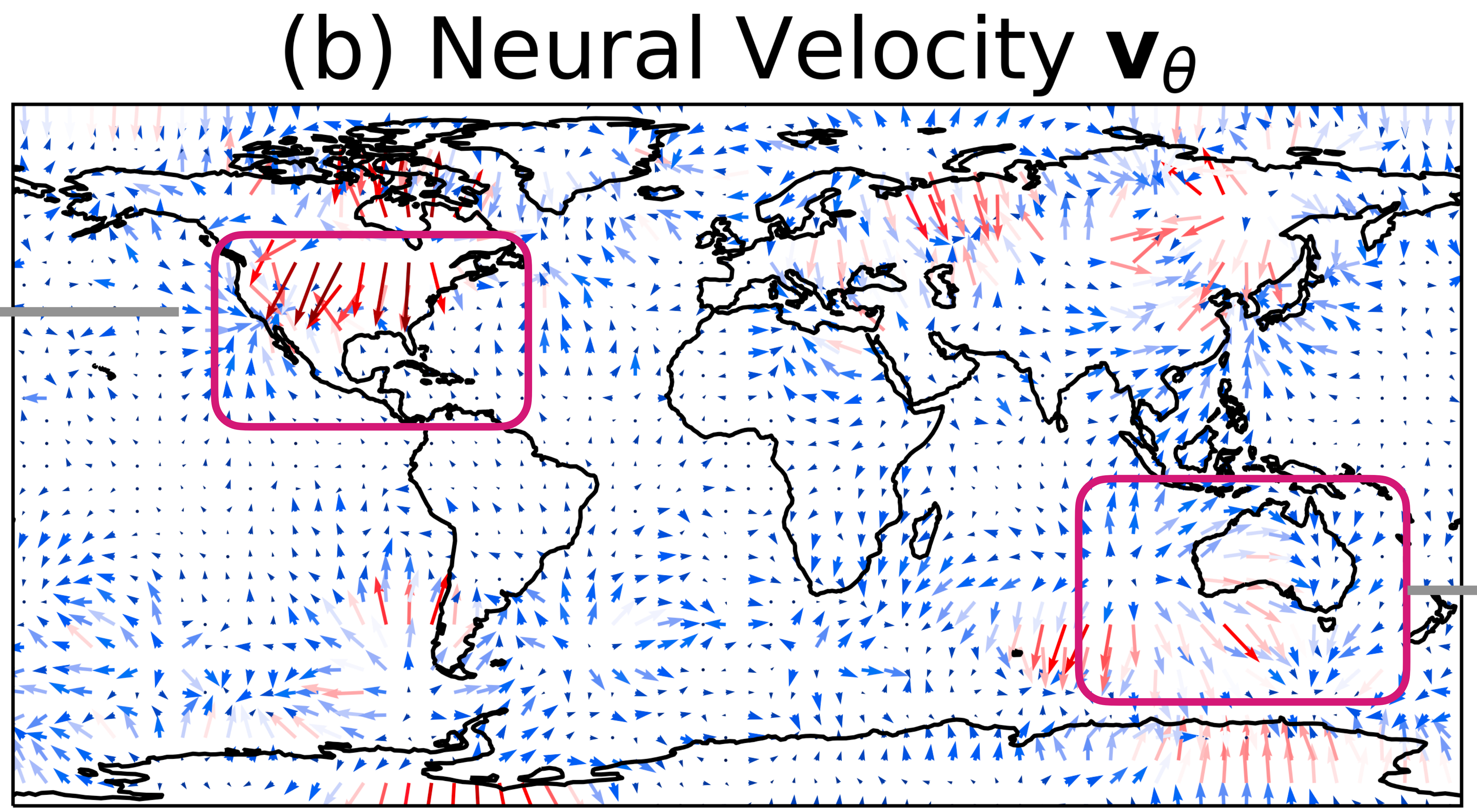
$$\dot{u}_k(x, t) = -\mathbf{v}_k(x, t) \cdot \nabla u_k(x, t) - u_k(x, t) \nabla \cdot \mathbf{v}_k(x, t)$$

Towards Neural Advection

We model change in the velocity by parametrising it as a function of $\mathbf{u}(t) = \{\mathbf{u}(x,t) : \mathbf{x} \in \Omega\}$, spatial gradients $\nabla \mathbf{u}$, current velocity $\mathbf{v}(t) = \{\mathbf{v}(x,t) : \mathbf{x} \in \Omega\}$ and spatiotemporal embeddings ψ

$$\dot{\mathbf{v}}_k(x,t) = f_\theta\Big(\mathbf{u}(t), \nabla \mathbf{u}(t), \mathbf{v}(t), \psi\Big)$$

Stronger currents on land

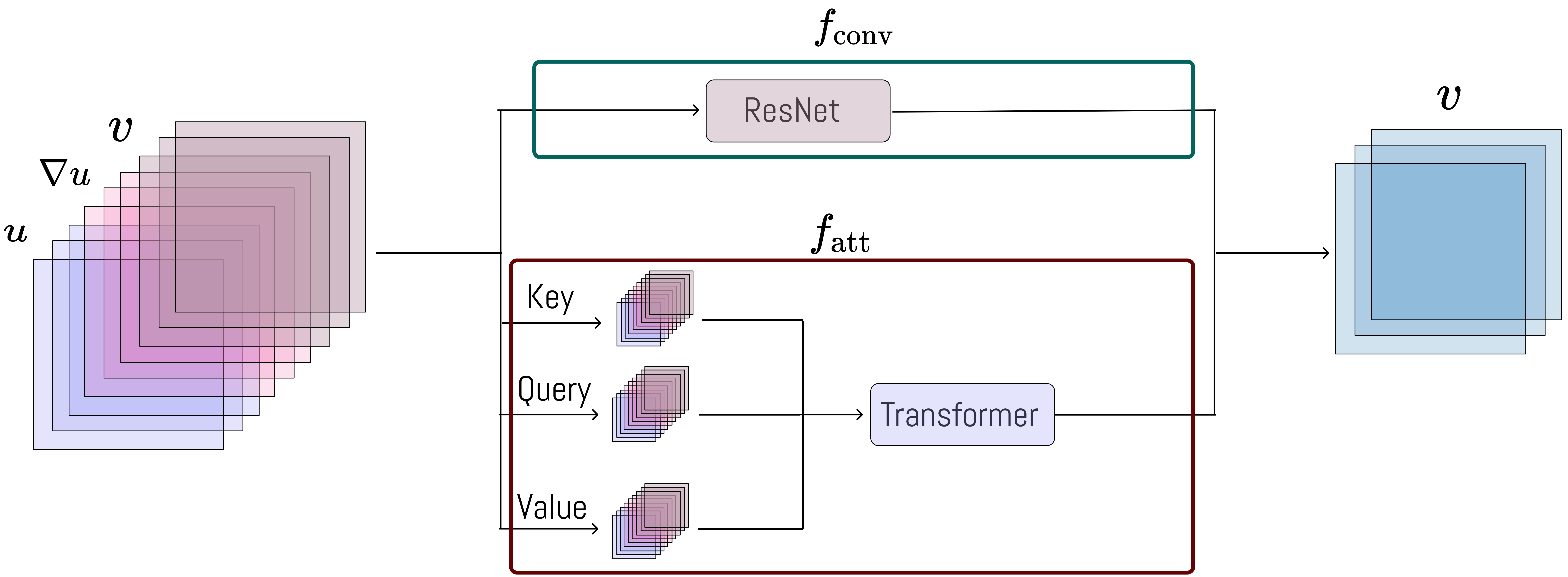


Cyclic currents

Local and global effects

To capture **local** and **global** effects pertaining to weather, we propose a hybrid network as,

$$f_\theta\Big(\mathbf{u}(t), \nabla\mathbf{u}(t), \mathbf{v}(t), \psi\Big) = f_{\text{conv}}\Big(\mathbf{u}(t), \nabla\mathbf{u}(t), \mathbf{v}(t), \psi\Big) + \gamma f_{\text{att}}\Big(\mathbf{u}(t), \nabla\mathbf{u}(t), \mathbf{v}(t), \psi\Big)$$



Spatio-temporal embeddings: Incorporate periodicity

Day and Season

$$\psi(t) = \left\{ \sin 2\pi t, \cos 2\pi t, \sin \frac{2\pi t}{365}, \cos \frac{2\pi t}{365} \right\}$$

Location

Longitude

$$\psi(\mathbf{x}) = \left[\{\sin, \cos\} \times \{h, w\}, \sin(h)\cos(w), \sin(h)\sin(w) \right]$$

Latitude

Joint time - location embeddings

Land sea mask

$$\psi(\mathbf{x}, t) = \left[\psi(t), \psi(\mathbf{x}), \psi(t) \times \psi(\mathbf{x}), \psi(c) \right], \qquad \psi(c) = \left[\psi(h), \psi(w), \mathrm{lsm}, \mathrm{oro} \right].$$

Orography (elevation)

Weather advection

Can we model sources and get predictive uncertainty ?

Modeling External Sources and Quantifying Uncertainty

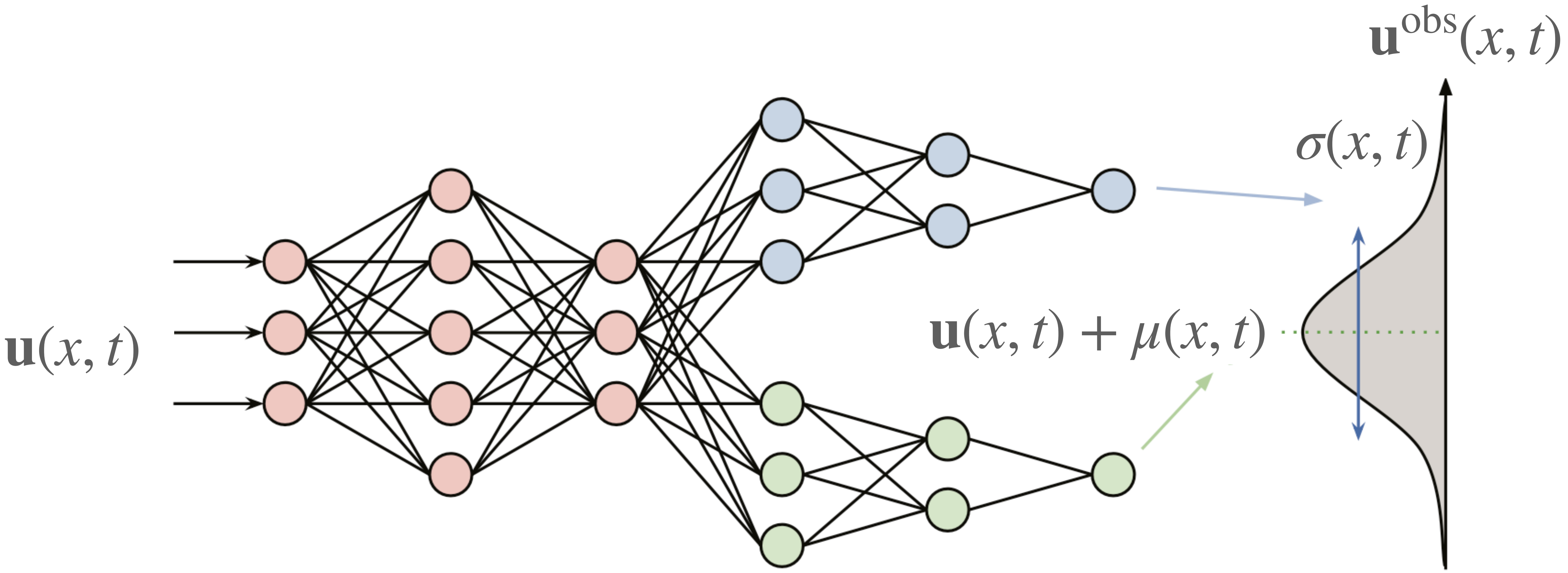
Limitation: Due to the closed system assumption, we cannot model value loss/gain and cannot quantify uncertainty.

Modeling External Sources and Quantifying Uncertainty

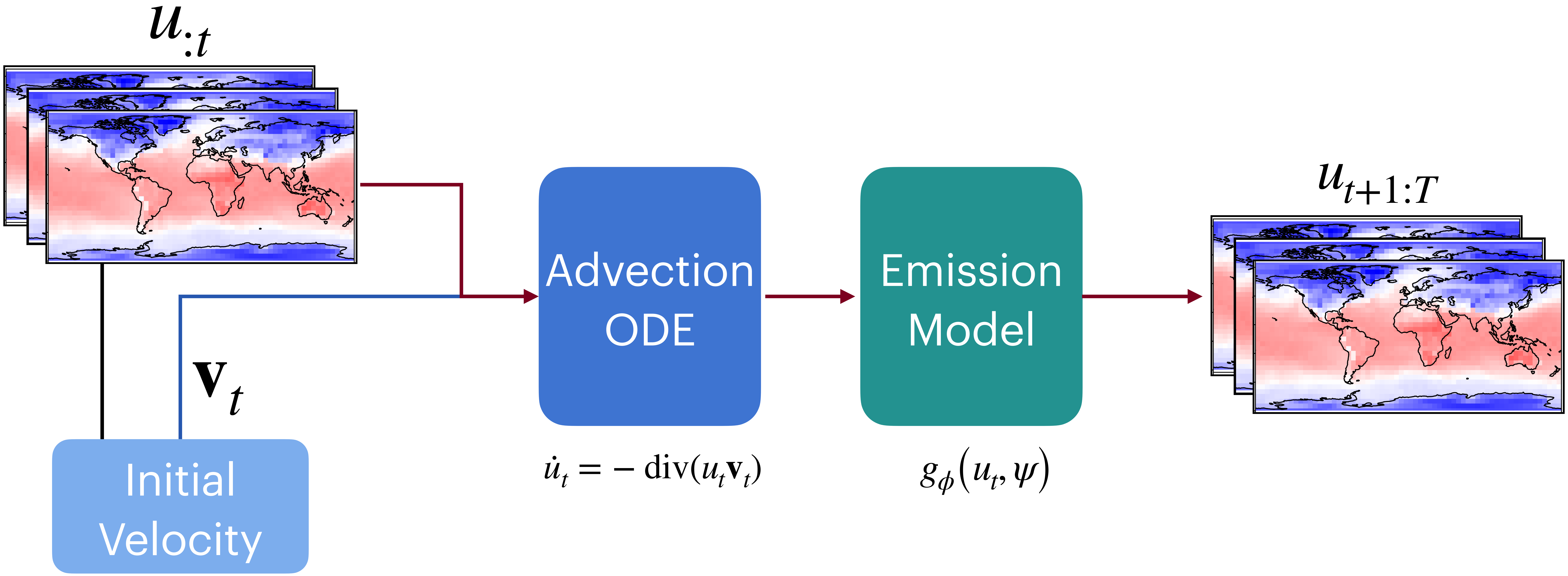
Limitation: Due to the closed system assumption, we cannot model value loss/gain and cannot quantify uncertainty.

Idea: Add a Gaussian emission model g to quantify value loss/gain and model uncertainty.

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



Whole Pipeline



$$\mathbf{v}_t = \arg\min_{\mathbf{v}_t} \{ || \dot{u}_t + \operatorname{div}(u_t \mathbf{v}_t) ||^2 + \alpha || \mathbf{v}_t ||_K^2 \}$$

Training Objective

Optimize the log-likelihood over the observations $\mathscr{D} = (\mathbf{y}_1, \ldots, \mathbf{y}_N), \mathbf{y}_i \in \mathbb{R}^{K \times H \times W}$ at $(t_1, \ldots, t_N)$

$$\log p(y \mid \theta, \phi) \propto \sum_{i=1}^{N} \log \mathscr{N}\left(y_i \mid \mathbf{u}_\theta(t_i) + \mu_\phi(t_i), \operatorname{diag}(\sigma_\phi(t_i))\right)$$

1. Solve $\mathbf{u}(t)$ forward with neural velocity $v_\theta(t)$
2. Evaluate likelihood
3. Backpropagate wrt (θ, ϕ)

$$\begin{pmatrix} \mathbf{u}_T \\ \mathbf{v}_T \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \end{pmatrix} + \int_0^T \begin{pmatrix} -\operatorname{div}(\mathbf{u}_t \mathbf{v}_t) \\ f_\theta\left(\mathbf{u}_t, \nabla \mathbf{u}_t, \mathbf{v}_t, \psi\right) \end{pmatrix} dt$$

Experiments

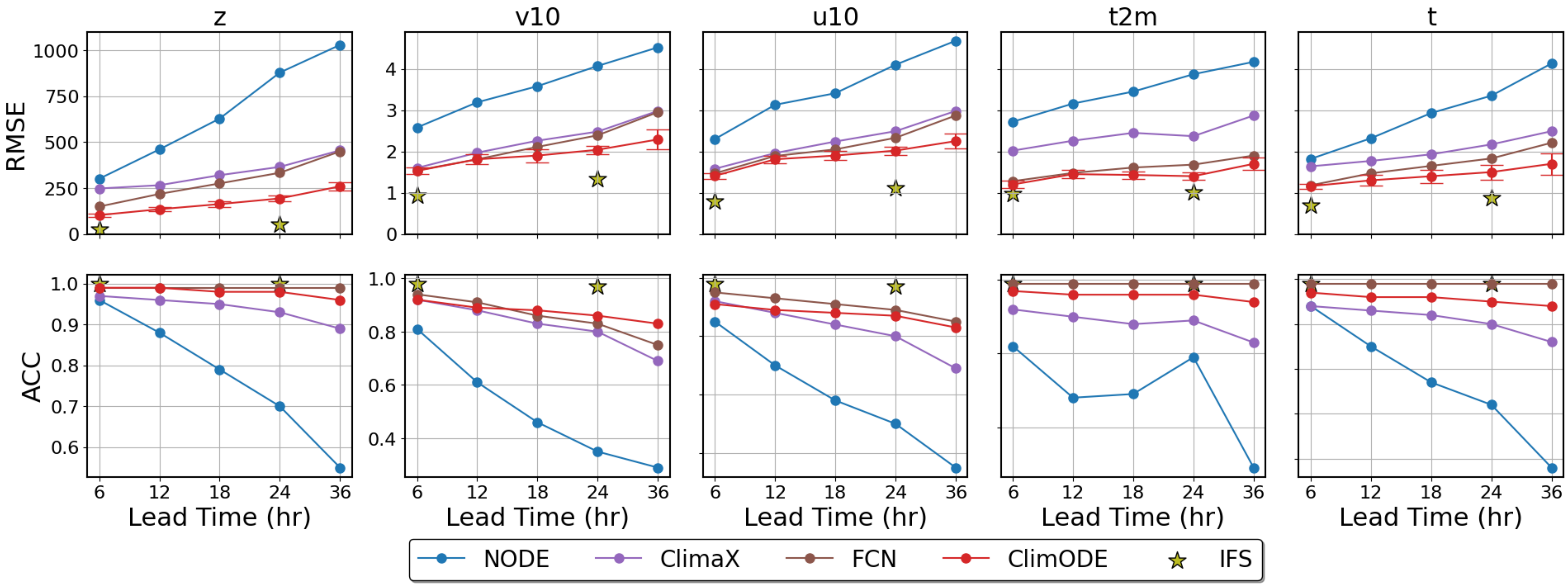
Results: Data

Data: We consider 5.625° resolution dataset from WeatherBench (regridded original ERA5) and chose following key meteorological variables.

Type	Variable name	Abbrev.	ECMWF ID	Levels
Static	Land-sea mask	lsm	172	
Static	Orography			
Single	2 metre temperature	t2m	167	
Single	10 metre U wind component	u10	165	
Single	10 metre V wind component	v10	166	
Atmospheric	Geopotential	z	129	500
Atmospheric	Temperature	t	130	850

Results: Global Forecasting

Global Forecasting: RMSE and ACC comparisons with baselines. We only consider these 5 quantities.



Results: Predictions

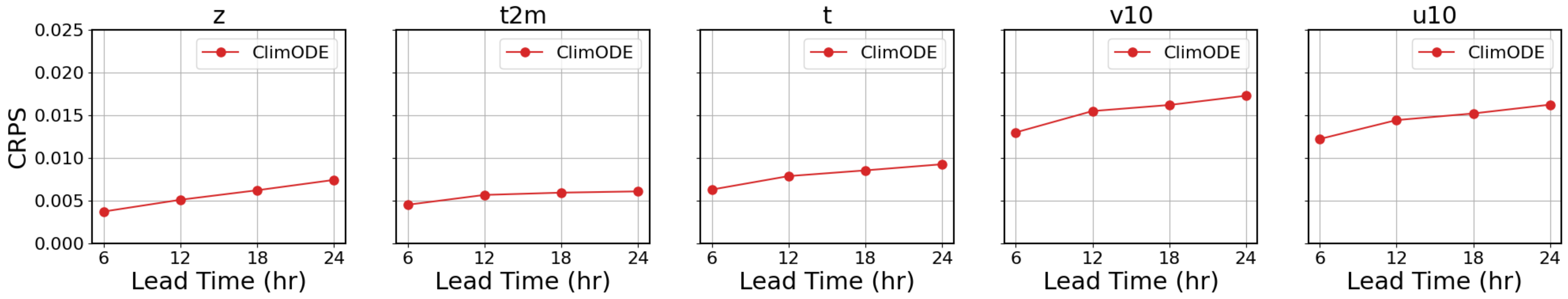
ODE predictions | ODE predictions + bias | Global Uncertainty Maps | Error = $u^{obs} - u^{pred}$

Results: CRP Scores

Continuous Ranked Probabilistic Scores (CRPS) is a measure of how good forecasts are in matching observed outcomes.

CRPS = 0 : Wholly Accurate
CRPS = 1 : Wholly Inaccurate

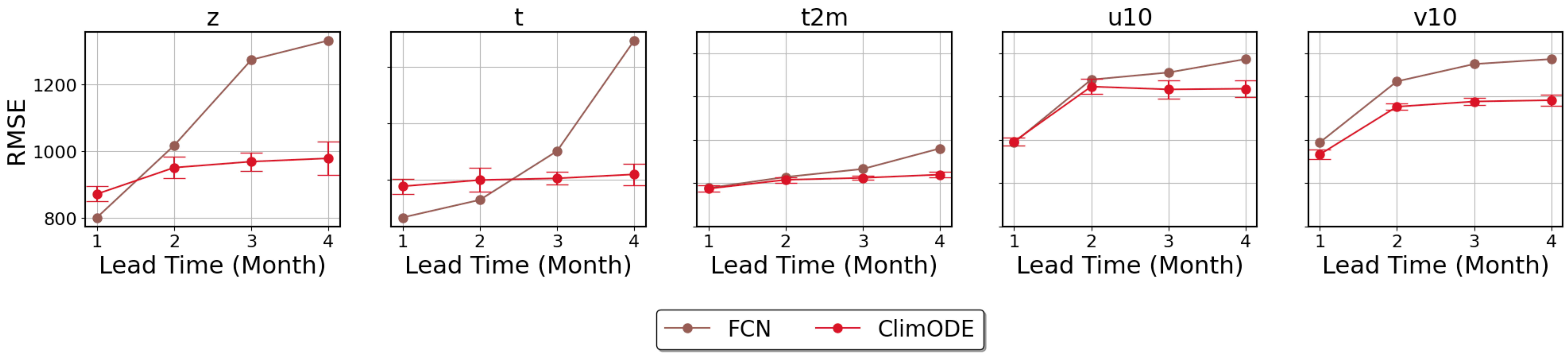
CRPS: ClimODE achieves lower scores demonstrating efficacy in prediction.



Climate Forecasting: Monthly Average Value Forecasting

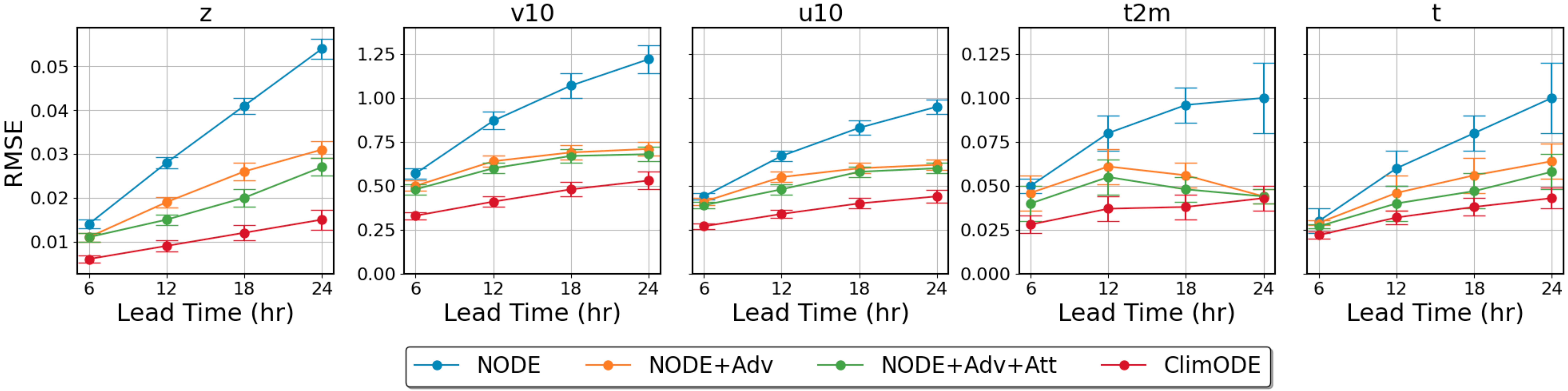
Forecast average weather conditions over one-month duration.

Climate Forecasting: Monthly average value forecasting.



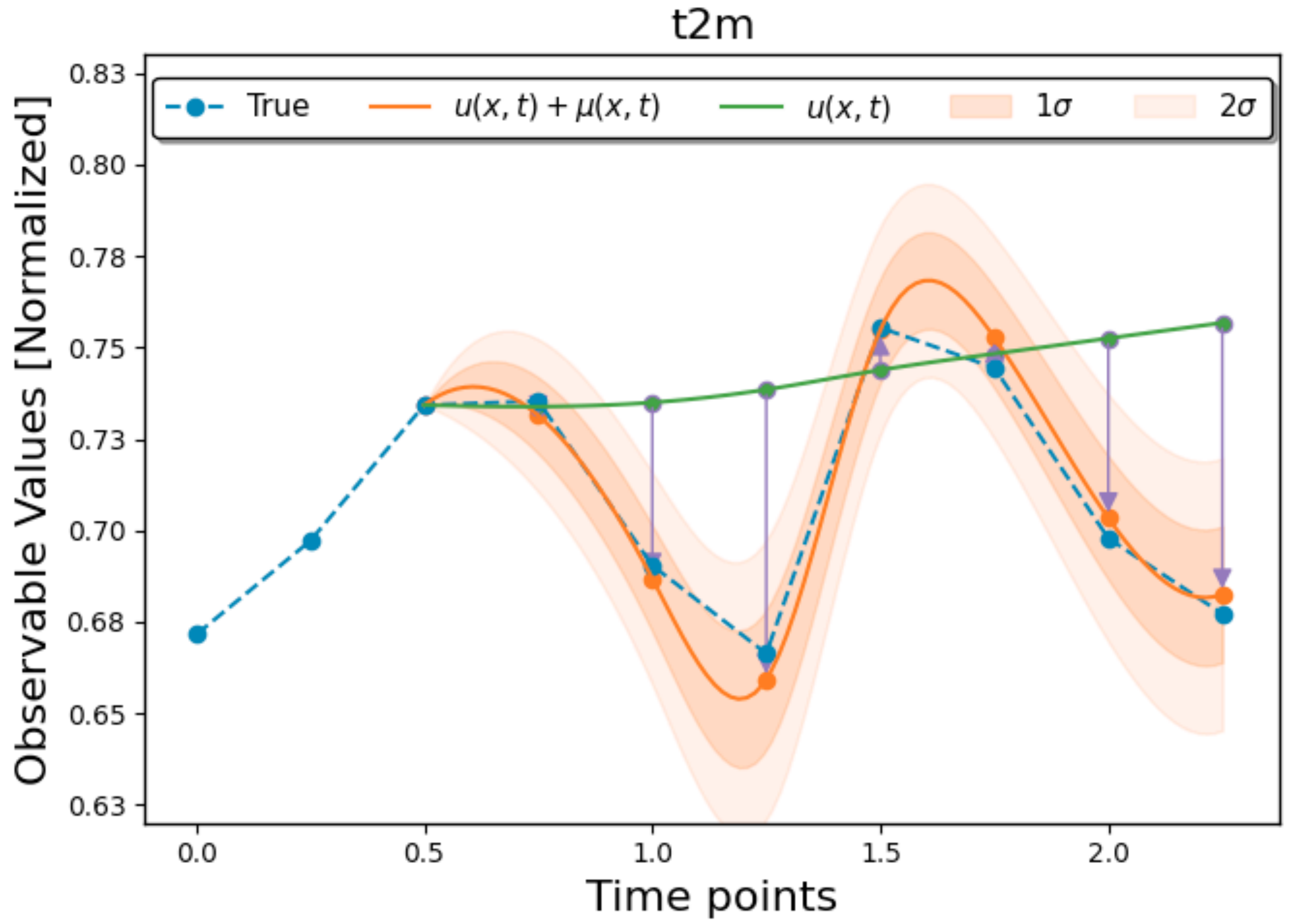
Ablation: Effect of individual components

Effect of each component: Advection and Emission components improves the best.

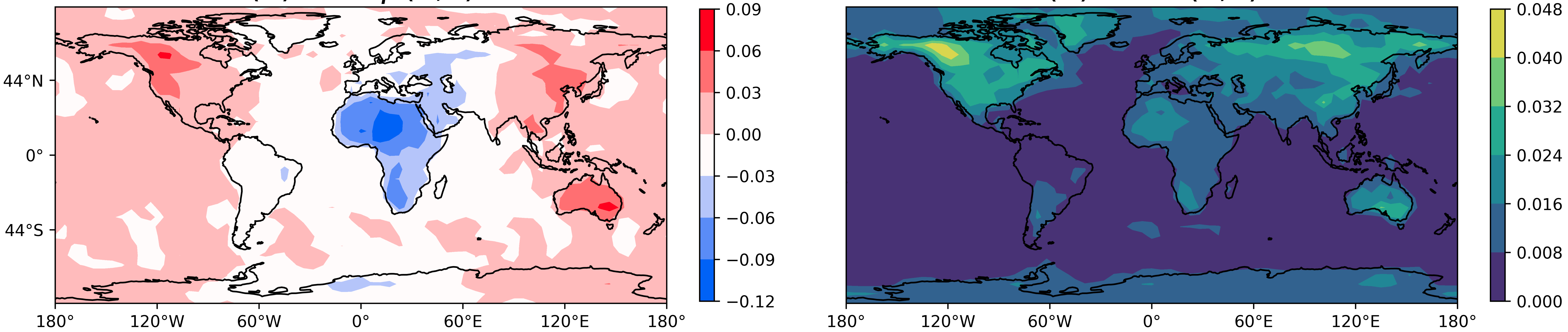


Ablation: Interpretability

Bias: Explains day-night cycle.
Uncertainty: Uncertainty highest on land and in north according to day-night cycle.

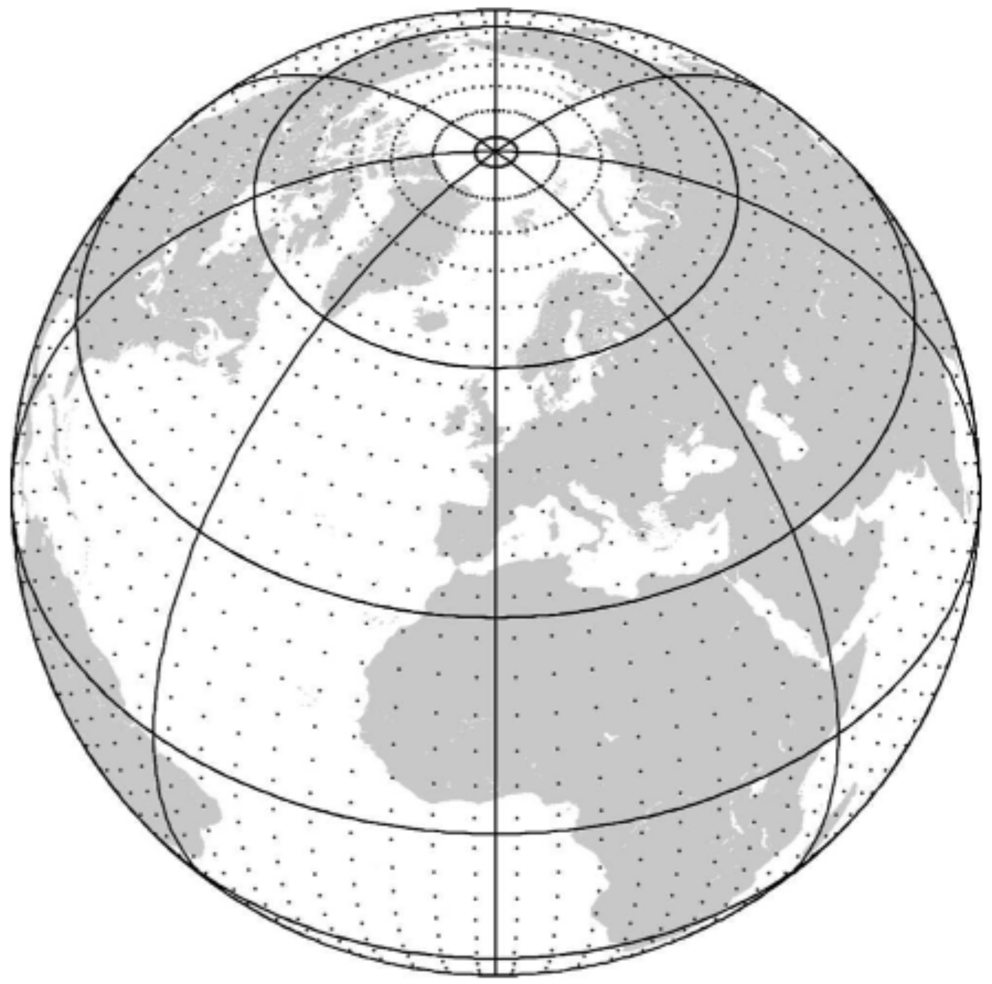


(a) Bias $\mu(x,t)$ (b) Std $\sigma(x,t)$



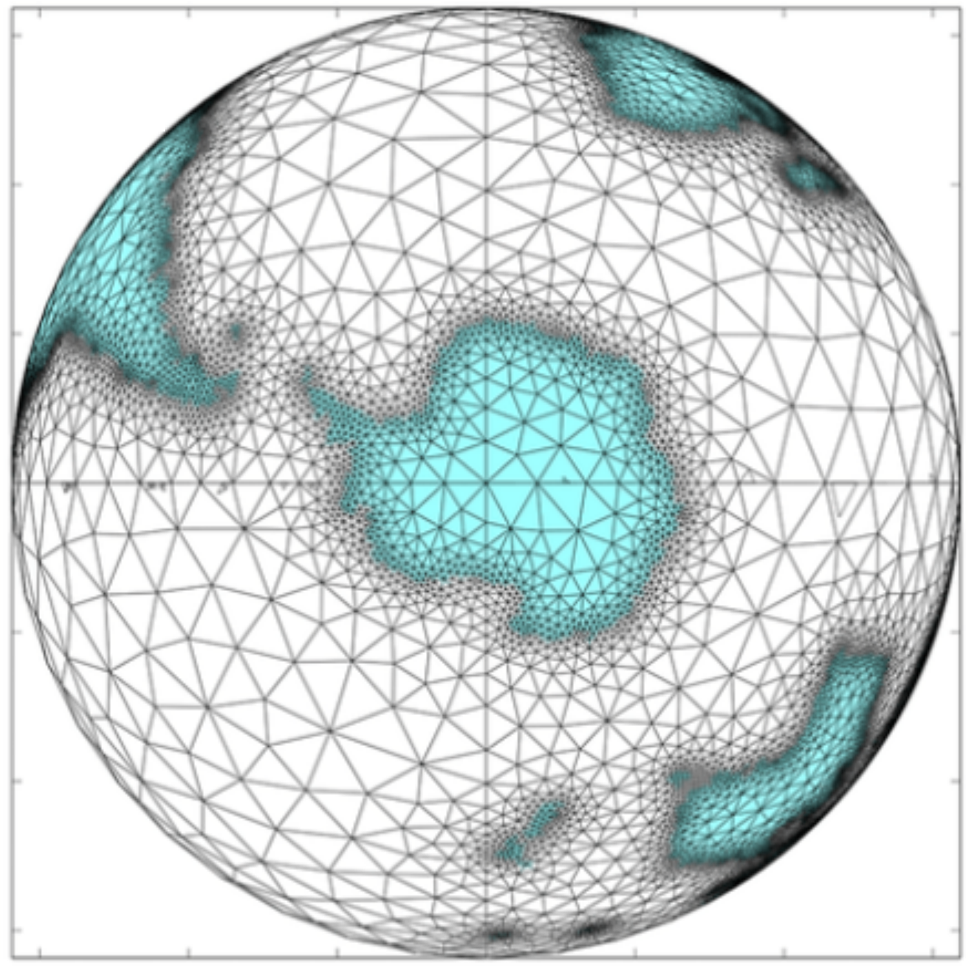
Conclusion

- We have shown an effective method to forecast macro-scale weather with advection.
- Establishes a new SOTA, provides interpretation and uncertainty quantification.



Future Work

- Incorporating geometrical aspect of earth.
- Higher resolution and region specific modeling.

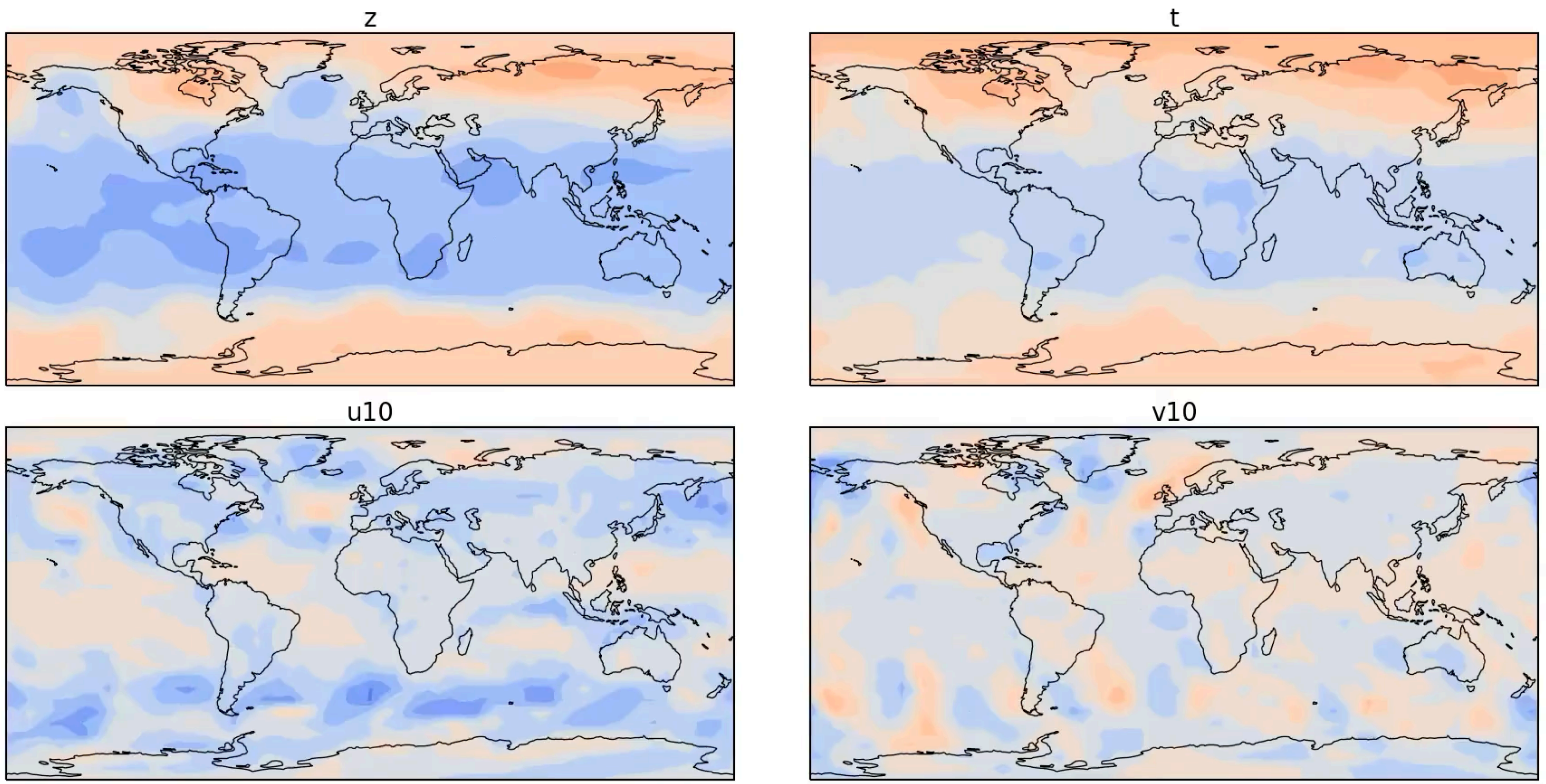


Wanna know more?

Visit our poster: **#62**

Today 7 May 10:45 a.m. CEST

Project Website



Yogesh Verma
yogesh.verma@aalto.fi

@Verma__Yogesh

Markus Heinonen
markus.o.heinonen@aalto.fi

@HeinonenMarkus

Vikas Garg
vgarg@csail.mit.edu

@montsgarg

How does it solve?

Given initial state $\mathbf{u_0}$ one can obtain forecast as, follows

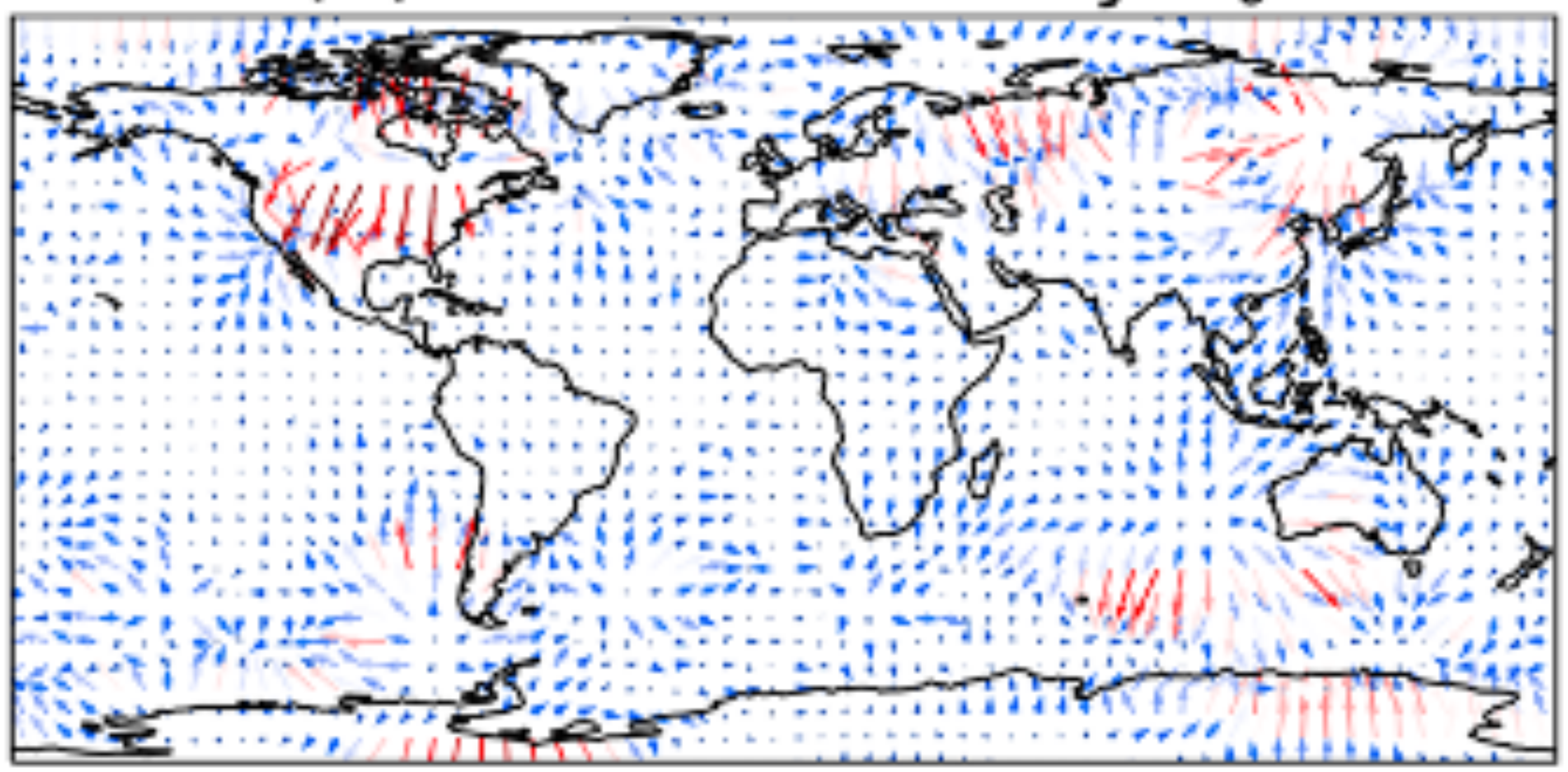
$$\begin{pmatrix} u_T \\ \mathbf{v}_T \end{pmatrix} = \begin{pmatrix} u_0 \\ \mathbf{v}_0 \end{pmatrix} + \int_0^T \begin{pmatrix} -\mathrm{div}(u_t \mathbf{v}_t) \\ f_\theta(u_t, \mathbf{v}_t, \nabla u_t) \end{pmatrix} dt$$

How does it solve?

Given initial state $\mathbf{u_0}$ one can obtain forecast as, follows

$$\begin{pmatrix} u_T \\ \mathbf{v}_T \end{pmatrix} = \begin{pmatrix} u_0 \\ \mathbf{v}_0 \end{pmatrix} + \int_0^T \begin{pmatrix} -\mathrm{div}(u_t \mathbf{v}_t) \\ f_\theta(u_t, \mathbf{v}_t, \nabla u_t) \end{pmatrix} dt$$

Only learn the change



How does it solve?

Given initial state $\mathbf{u_0}$ one can obtain forecast as, follows

$$\begin{pmatrix} u_T \\ \mathbf{v}_T \end{pmatrix} = \begin{pmatrix} u_0 \\ \mathbf{v}_0 \end{pmatrix} + \int_0^T \begin{pmatrix} -\mathrm{div}(u_t \mathbf{v}_t) \\ f_\theta(u_t, \mathbf{v}_t, \nabla u_t) \end{pmatrix} dt$$

Still unknown? How to start the dynamics?

Initial velocity Inference

We need an initial velocity estimate to start the system, obtain it why minimising the advection equation as,

$$\mathbf{v}_t = \arg\min_{\mathbf{v}_t} \{ ||\dot{u}_t + \nabla \cdot (u_t \mathbf{v}_t)||^2 + \alpha ||\mathbf{v}_t||_K^2 \}$$

where $\dot{u}$ is temporal derivative approximated via past states $u(t < t_0)$ and include a smoothness kernel K to obtain spatially smooth velocities.