

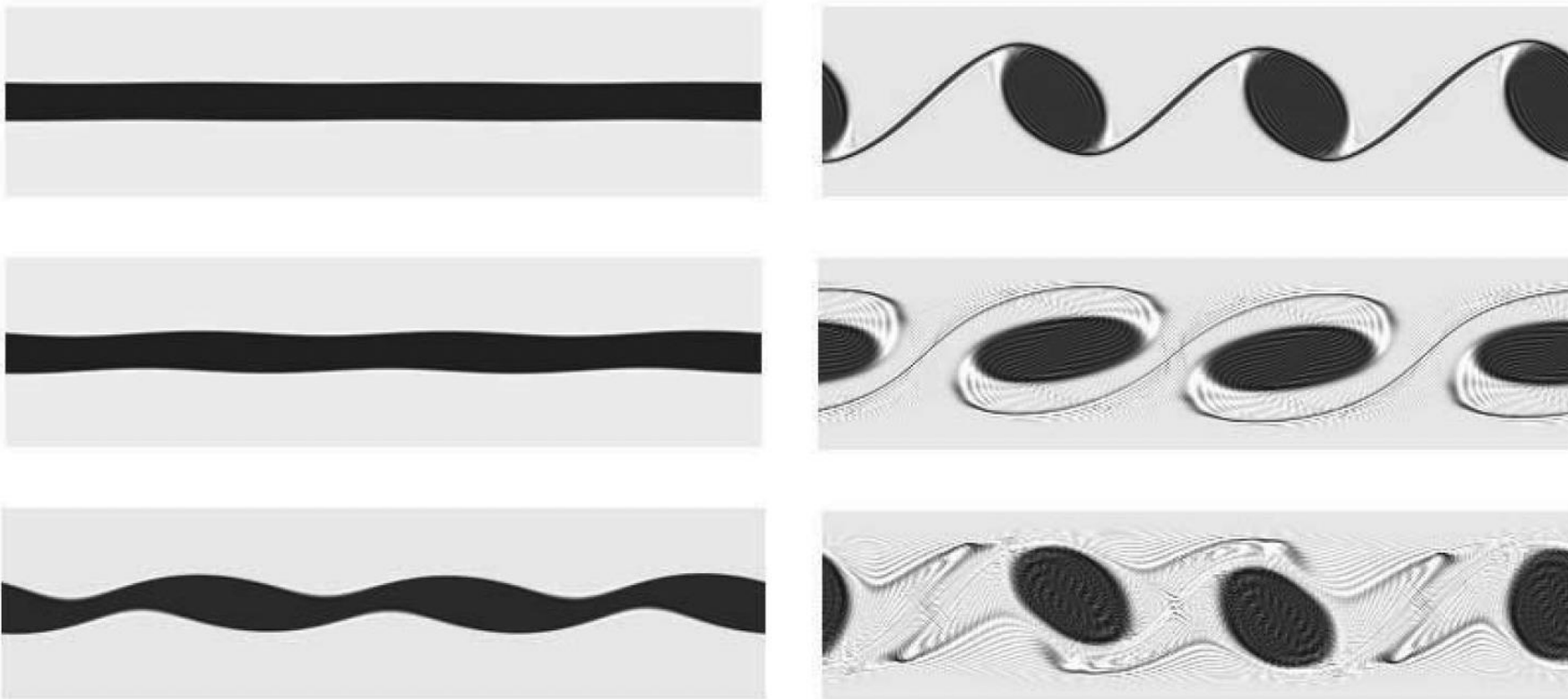
$$\frac{D}{Dt} \left(\nabla^2 \psi + \beta y - \frac{1}{L_d^2} \psi \right) = 0 ,$$

where $\psi = (g/f_0)\eta$, $L_d^2 = gH/f_0^2$, and the advective derivative is

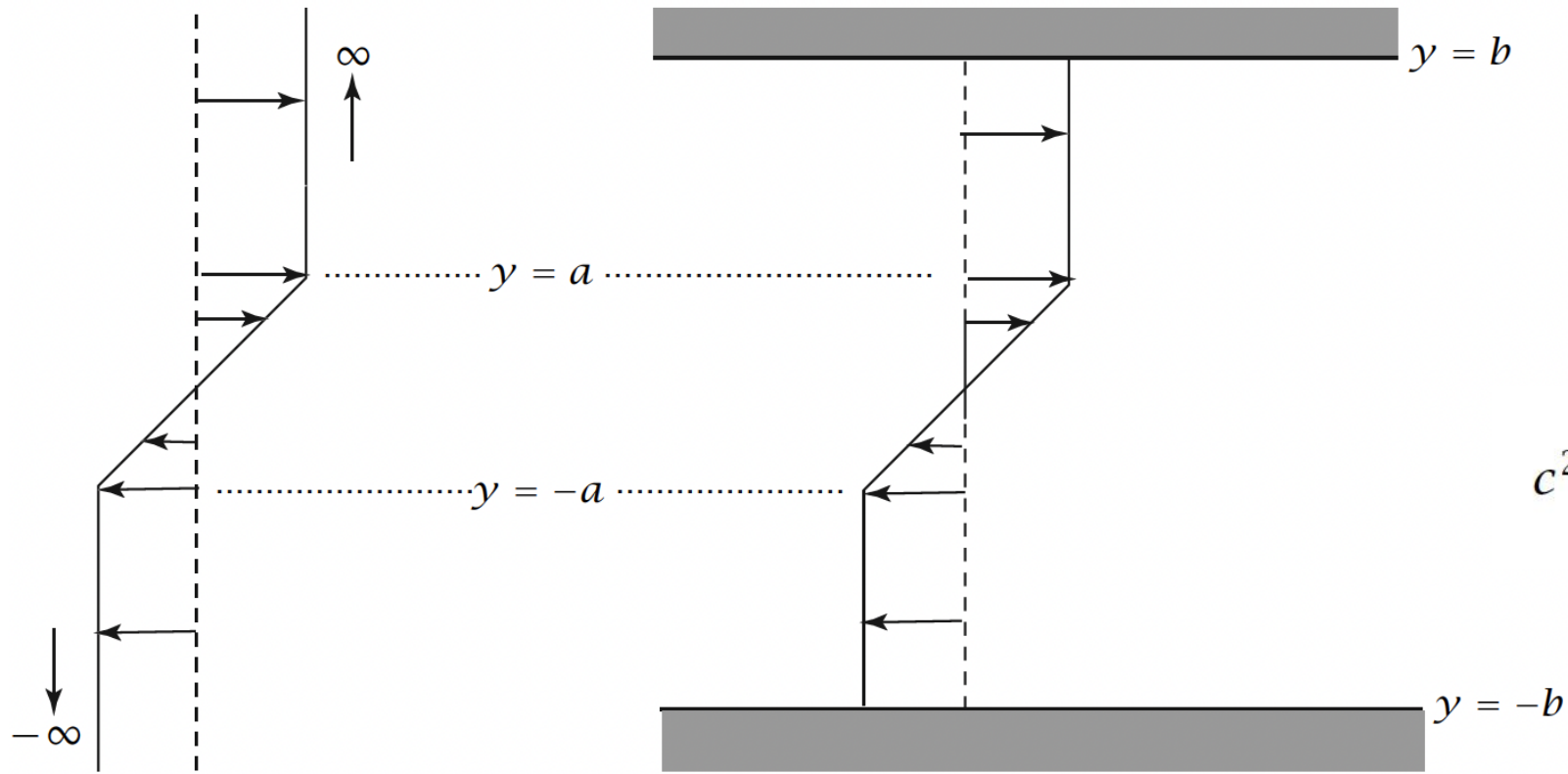
$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u_g \frac{\partial}{\partial x} + v_g \frac{\partial}{\partial y} = \frac{\partial}{\partial t} - \frac{\partial \psi}{\partial y} \frac{\partial}{\partial x} + \frac{\partial \psi}{\partial x} \frac{\partial}{\partial y} = \frac{\partial}{\partial t} + J(\psi, \cdot). \quad (5.67)$$

$$q \equiv \zeta + \beta y - \frac{f_0}{H} \eta = \nabla^2 \psi + \beta y - \frac{1}{L_d^2} \psi \quad (5.69)$$

is the *shallow water quasi-geostrophic potential vorticity*.



black color represents vorticity



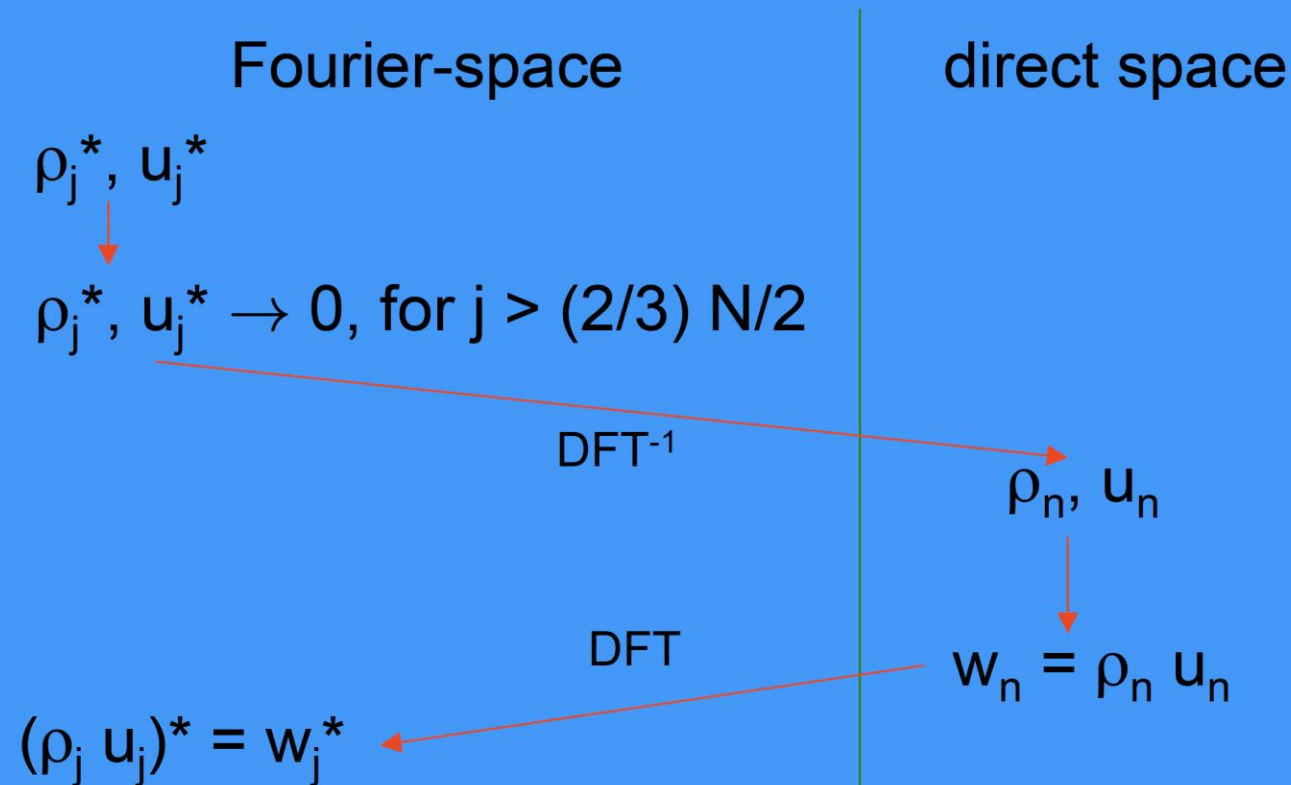
$$\psi' = \text{Re } \tilde{\psi}(y) e^{ik(x-ct)}$$

$$c^2 = \left(\frac{U_0}{2ka}\right)^2 \left[(1 - 2ka)^2 - e^{-4ka} \right],$$

Fig. 6.4 Barotropically unstable velocity profiles. In the simplest case, on the left, a region of shear is sandwiched between two infinite regions of constant velocity. The edge waves at $y = \pm a$ interact to produce an instability. If $a = 0$, then the situation corresponds to that of Fig. 6.1, giving Kelvin-Helmholtz instability. In the case on the right, the flow is bounded at $y = \pm b$. It may be shown that the flow is still unstable, provided that b is sufficiently larger than a . If $b = a$ (plane Couette flow) the flow is stable to infinitesimal disturbances.

non-linearities, de-aliased

- assume you need to evaluate $\text{DFT}(\rho_j u_j)$, having given the Fourier transforms ρ_j^* and u_j^* :



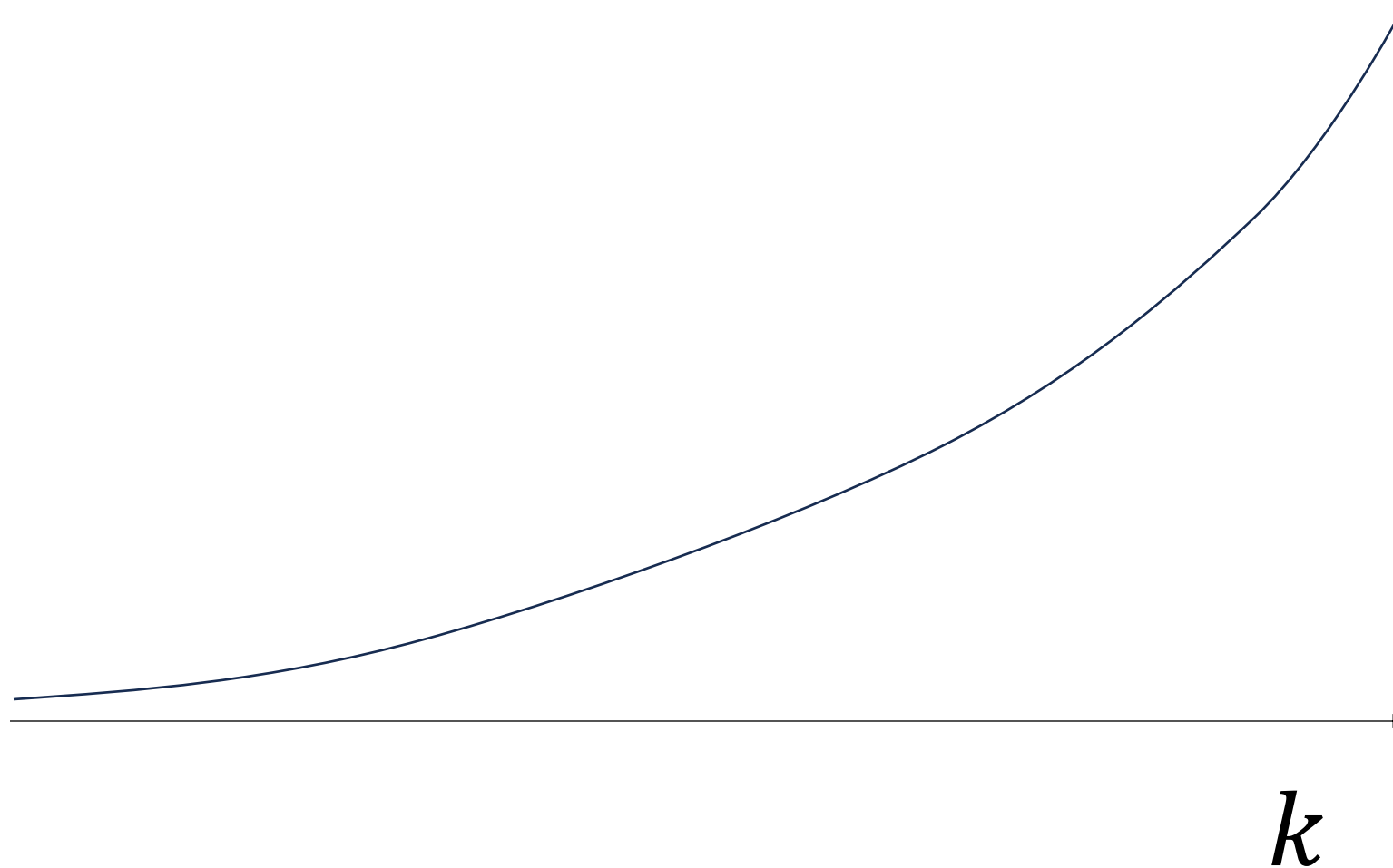
Runge-Kutta 2

$$\begin{aligned}q_1 &= hF(\phi^n), & \phi_1 &= \phi^n + q_1 \\q_2 &= hF(\phi_1) - q_1, & \phi^{n+1} &= \phi_1 + q_2/2\end{aligned}$$

Runge-Kutta 4

$$\begin{aligned}q_1 &= hF(\phi^n), & q_2 &= hF(\phi^n + q_1/2) \\q_3 &= hF(\phi^n + q_2/2), & q_4 &= hF(\phi^n + q_3) \\ \phi^{n+1} &= \phi^n + (q_1 + 2q_2 + 2q_3 + q_4)/6\end{aligned}$$

$$-\Delta^2 u \sim -(k^2 + l^2)^2 \tilde{u}$$



Homework 1

Homework 1 (deadline: Oct. 10)

Demonstrate barotropic instability numerically based on the shallow water quasi-geostrophic equation in the presence of a background zonal wind shear $U(y)$. You can build your own code based on the python code shown in the class (the python code can be downloaded from the course website https://qiuyang50.github.io/_pages/modeling_2024fall/). The shallow water quasi-geostrophic equation reads as follows,

$$\frac{D}{Dt} (\nabla^2 \psi - f^2 F \psi) + \beta \frac{\partial \psi}{\partial x} = 0 \quad (1)$$

where $\frac{D}{Dt} = \frac{\partial}{\partial t} - \frac{\partial \psi}{\partial y} \frac{\partial}{\partial x} + \frac{\partial \psi}{\partial x} \frac{\partial}{\partial y}$, f, F, β are all dimensionless constant, $U(y)$ has the following profile,

$$U(y) = \begin{cases} 1 & y > a \\ y/a & -a \leq y \leq a \\ -1 & y < -a \end{cases}$$

Step 1: initialize the variable ψ . Hint: you can set the initial value of ψ as a combination of background state and a perturbation, $\psi(x, y, 0) = \Psi(y) + \psi'$, where $\Psi(y)$ satisfies $U(y) = -\frac{\partial \Psi(y)}{\partial y}$.

Step 2: add a perturbation ψ' , which has to be small in amplitude and periodic at the domain boundaries. For example,

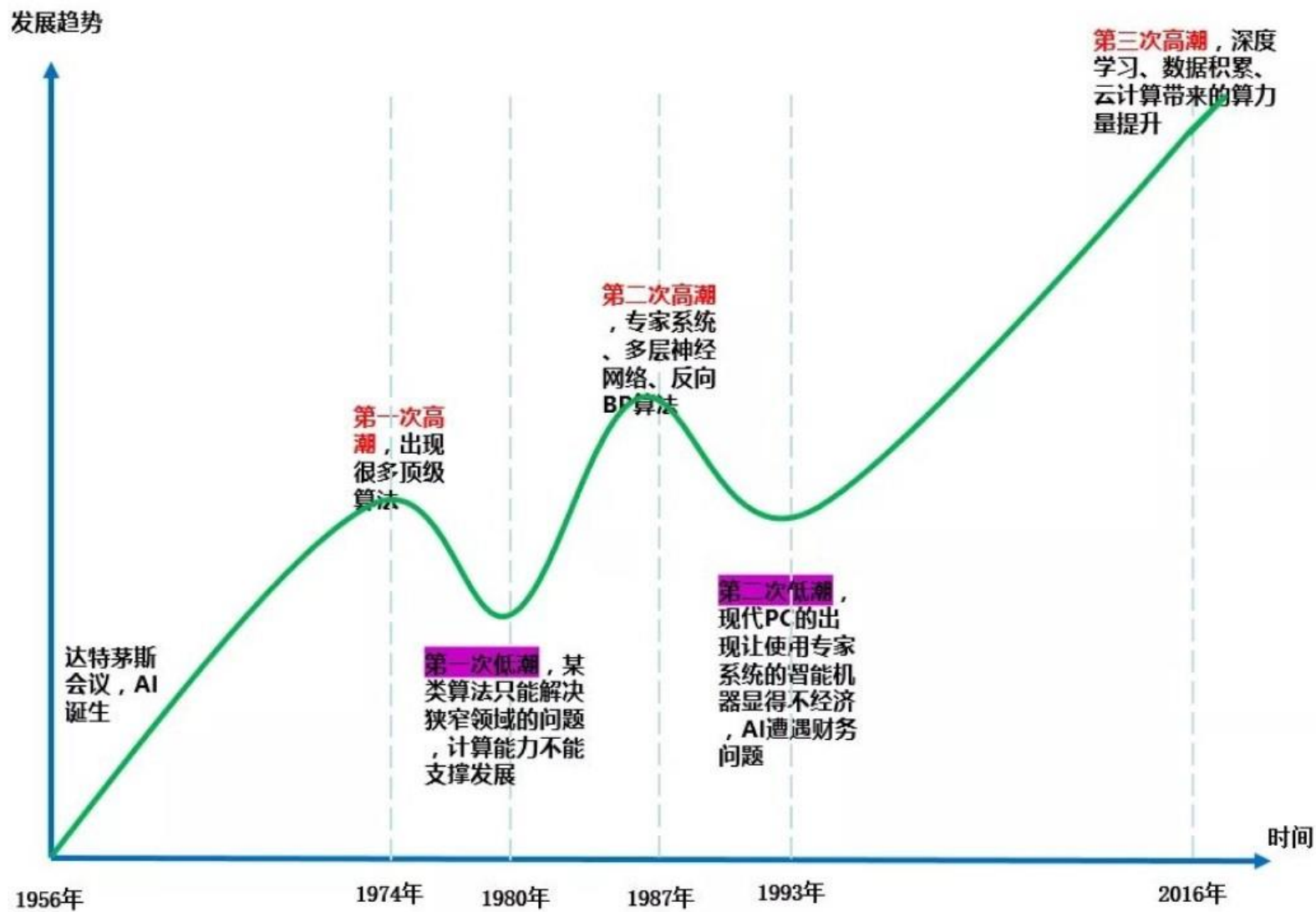
$$\psi' = 10^{-4} \sin\left(\frac{2\pi k}{L} x\right) e^{-\frac{y^2}{a^2}}$$

You may try other forms of perturbations.

Step 3: consider multiple scenarios with different values of a and k and compare your results with the theoretical prediction mentioned in the class. Hint: $a * k$ should be small enough for the emergence of instability. See chapter 6.2 (page 256) in Vallis's book.

Homework requirement: please summarize all you findings in a report, particularly including several snapshots of the vorticity field to demonstrate the barotropic instability. A thorough discussion about the choice of a and k in terms of barotropic instability is appreciated.

2.1 AI预测前沿进展



详细描述请参考 <https://datawhalechina.github.io/thorough-pytorch/第零章/0.1 人工智能简史.html>



John McCarthy

The Proposal states:

We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that **every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.** An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

第一次浪潮

达特茅斯会议之后，人工智能进入了一个高速发展的时期，也就是所谓的“第一次浪潮”。这次浪潮一直持续到二十世纪六十年代中期。在这近10年的时间里，计算机本身的“智能”发展，快速进步的是人工智能的一面。

很多对后来人工智能发展起到奠基作用的发明都诞生在这个时间段。感知机是神经网络理论中神经元的最早模型，人工神经网络理论得到了巨大的发展。强化学习的雏形也是在那段时间

彼时的科学界都弥漫着快乐的气氛，大家都认为，只要坚持走下去，人工智能就一定会得到跨越式的发展。

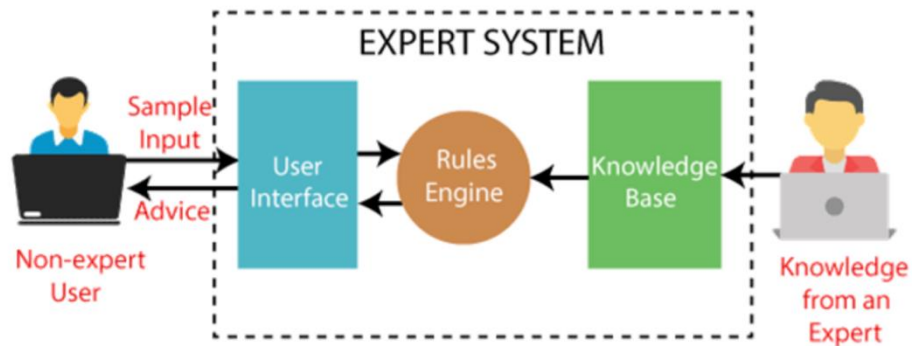
“认知系统”研究部门主任。在那里，他于1958年模拟大脑进行图像识别而设计了成名作品“感知器”（Perceptron）。这是人工智能发展史上一块不可或缺的里程碑。他的感知器 Mark I Perceptron 就是现在称为人工神经网络（Artificial Neural Network）的雏形装置，基于生物神经网络原理设计而成，在一台 IBM-704 计算机上实现。感知器包括一个含

1966年前后，AI遭遇了瓶颈。人们发现逻辑证明器、感知器、强化学习等等只能做很简单、非常专门且很窄的任务，稍微超出范围就无法应对。当时的计算机有限的内存和处理速度不足以解决任何实际的AI问题。研究者们很快就意识到，要求程序对这个世界具有儿童水平的认识这个要求都太高了——那时没人能够做出人工智能需要的巨大数据库，也没人知道一个程序怎样才能学到如此丰富的信息。另一方面，有很多计算复杂度以指数程度增加，这成为了不可能完成的计算任务。

一个由512台小型步进电动机组成的隐藏层以及一个元的输出层。在这个单隐藏层网络结构的装置中，罗森布拉特为它设计了一套相应的感知算法，让计算机通过反复迭代而自动调整参数，直至能正确识别输入。这种单层神经网络在处理线性可分离模式识别方面具有某种初步的自我学习能力。



罗森布拉特和他的感知器



时间来到了20世纪80年代。经过了数十年的研究，科学家们逐渐放弃了初代的符号学派思路，**改用统计学的思路**来研究人工智能。研究思路的改变再加上硬件技术的升级，人工智能的发展又一次迎来的新的契机。在那个时代，基于人工智能的“专家系统”受到了绝对的热捧。特定领域的“**专家系统**”程序被更广泛的采纳，该系统能够根据领域内的专业知识，推理出专业问题的答案，人工智能也由此变得更加“实用”，专家系统所依赖的知识库系统和知识工程成为了当时主要的研究方向。

但由于专家系统仅适用于某些特定场景，很快人们就对这一系统由狂热的追捧逐渐走向巨大的失望。与此同时，现代电子计算机的出现让“知识查询”的费用进一步降低，人们更加深刻的意识到专家系统是如此的古老陈旧。因此，政府部门下调了专家系统的研发资金。缺少了资金的支持，由专家系统再次兴起的人工智能研究又一次陷入了低谷之中。



第三次浪潮

1993年后，新的数学工具，理论和摩尔定律的出现，使得**计算机的算力**进一步提高，以**深度学习为核心的机器学习算法**获得发展，新的芯片和云计算的发展使得可用的计算能力获得飞跃式提高，**大数据的发展**使得海量数据的储存和分析成为可能。在这样的技术背景下，人工智能的第三次浪潮即将到来。

人工智能的第三次浪潮有两个重要的时间节点：2006年和2016年。2006年是深度学习发展史的分水岭。杰弗里辛顿在这一年发表了《一种深度置信网络的快速学习算法》，其他重要的深度学习学术文章也在这一年被发布，在基本理论层面取得了若干重大突破。而2016年3月，**谷歌DeepMind研发的AlphaGo在围棋人机大战中击败韩国职业九段棋手李世乜**，“人工智能”一词正式进入普通民众的视野并被逐渐熟知。至此，人工智能正式迈向了从“科研领域的应用型工具”到“实用性，功能性工具”的转变，人工智能有了新的研究方向和研究模式，即从过去的学术主导型研究逐渐走向了商业主导型研究。随着人类社会对智能化工具的不断追求和探索，人工智能的发展迎来了全新的时代。



持续突破，还是第三次低谷？

ChatGPT >

Can you write a python code for solving transport equation $u_t + c*u_x=0$

The transport equation $u_t + cu_x = 0$ is a simple linear partial differential equation (PDE). Here's a Python code using finite difference methods (FTCS) to numerically solve it:

Python Code

```
python Copy
import numpy as np
import matplotlib.pyplot as plt

# Parameters
c = 1.0 # Wave speed
L = 10.0 # Length of the domain
T = 5.0 # Total time
nx = 100 # Number of spatial grid points
nt = 500 # Number of time steps
dx = L / (nx - 1) # Spatial step size
dt = T / nt # Time step size
sigma = c * dt / dx # CFL condition

# Initial condition: a Gaussian profile
def initial_condition(x):
    return np.exp(-0.5 * (x - L / 2) ** 2)

# Spatial grid and initial state
x = np.linspace(0, L, nx)
u = initial_condition(x)

# Store the solution at each time step for visualization
u_history = [u.copy()]

# Time-stepping loop (FTCS method)
for n in range(1, nt):
    u_new = u.copy()
    for i in range(1, nx-1):
        u_new[i] = u[i] - sigma * (u[i] - u[i-1])
    u = u_new.copy()
    u_history.append(u)

# Plotting the results
plt.figure(figsize=(8, 6))
plt.plot(x, u_history[0], label="Initial Condition")
plt.plot(x, u_history[nt//4], label="1/4 Time Step")
plt.plot(x, u_history[nt//2], label="1/2 Time Step")
plt.plot(x, u_history[-1], label="Final Time Step")
plt.title('Solution to the Transport Equation $u_t + c u_x = 0$')
plt.xlabel('x')
plt.ylabel('u')
plt.legend()
plt.grid()
plt.show()
```

Explanation:

1. Parameters:

- c: Wave speed.
- L: Length of the spatial domain.
- T: Total time to simulate.
- nx: Number of spatial grid points.
- nt: Number of time steps.
- dx: Spatial step size.
- dt: Time step size.
- sigma: Courant-Friedrichs-Lewy (CFL) condition, necessary for stability.

2. Initial Condition:

- A Gaussian profile is used as the initial condition, which is centered at $x = L/2$.

3. Numerical Scheme:

- The finite difference method used is Forward-Time Central-Space (FTCS). This is an explicit scheme to update the solution at each time step.

4. Visualization:

- The code plots the initial profile, intermediate steps, and the final profile.

Let me know if you need further customization!

Can you show me how to derive shallow water equation

The shallow water equations (SWEs) are a set of partial differential equations that describe fluid flow in situations where the horizontal length scale is much larger than the vertical depth scale, such as oceans or atmospheric flows. These equations are derived from the basic principles of fluid mechanics—conservation of mass (continuity equation) and conservation of momentum (Euler's equations)—under the assumption of hydrostatic pressure.

1. Assumptions:

- The flow is incompressible.
- The fluid is inviscid (i.e., viscosity is neglected).
- The vertical acceleration is small, and hydrostatic balance applies in the vertical direction.
- The pressure at the free surface of the fluid is constant (usually atmospheric pressure).

Let's derive the shallow water equations in two dimensions (x and y), assuming a depth $h(x, y, t)$.

2. Conservation of Mass (Continuity Equation):

Consider a small fluid element in a domain. The rate of change of the total mass in this element must equal the net flux of mass into or out of the element.

The water column height is given by $h(x, y, t) = H + \eta(x, y, t)$, where H is the constant mean water depth, and $\eta(x, y, t)$ is the deviation of the free surface from the mean depth.

Let $u(x, y, t)$ and $v(x, y, t)$ represent the horizontal velocity components in the x and y directions.

The mass conservation equation is:

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} = 0$$

This is the **continuity equation** in the shallow water approximation.

3. Conservation of Momentum (Euler's Equations):

The shallow water momentum equations are derived from the Euler equations under the hydrostatic assumption. In the x - and y -directions, the momentum equations take the following form:

a) Momentum Equation in the x -Direction:

Using the incompressible Euler equations for momentum and applying the shallow water approximation:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -g \frac{\partial \eta}{\partial x}$$

where g is the acceleration due to gravity and $\eta(x, y, t)$ is the surface height deviation from mean depth.

Multiplying by h (to account for the height of the water column):

$$\frac{\partial(hu)}{\partial t} + \frac{\partial(hu^2)}{\partial x} + \frac{\partial(huv)}{\partial y} = -gh \frac{\partial \eta}{\partial x}$$

b) Momentum Equation in the y -Direction:

Similarly, for the y -direction:

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -g \frac{\partial \eta}{\partial y}$$

Multiplying by h :

$$\frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial(hv^2)}{\partial y} = -gh \frac{\partial \eta}{\partial y}$$

4. Summary of the Shallow Water Equations:

1. Continuity equation (Conservation of mass):

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} = 0$$

2. Momentum equation in the x -direction:

$$\frac{\partial(hu)}{\partial t} + \frac{\partial(hu^2)}{\partial x} + \frac{\partial(huv)}{\partial y} = -gh \frac{\partial \eta}{\partial x}$$

3. Momentum equation in the y -direction:

$$\frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial(hv^2)}{\partial y} = -gh \frac{\partial \eta}{\partial y}$$

These are the **two-dimensional shallow water equations**, which describe how the fluid height and horizontal velocity evolve over time. In these equations, u and v represent the velocity components, h is the height of the fluid, and g is the acceleration due to gravity.

Can you draw a picture for the sun



Here is a bright and radiant depiction of the sun. Let me know if you'd like any changes or further details!



can you add a face on top of that



Here is the sun with a cheerful face added! Let me know if you'd like any further adjustments.

Can you explain to me what is the Madden-Julian Oscillation?

The **Madden-Julian Oscillation (MJO)** is a large-scale atmospheric phenomenon that plays a crucial role in tropical weather patterns, especially in influencing rainfall and storm activity across different regions. Here's a basic breakdown of what it is:

1. Definition:

The MJO is an eastward-moving pulse of cloudiness, rainfall, and wind that travels around the globe near the equator, particularly in the Indian and Pacific Oceans. It's different from other climate systems like El Niño, which are stationary and last for long periods. The MJO moves across the tropics, typically over a 30- to 60-day period.

2. Phases:

The MJO consists of two main phases:

- **Convective Phase:** This is the active phase, where there is increased cloud cover, rain, and thunderstorm activity. During this phase, moisture builds up in the atmosphere.
- **Suppressed Phase:** This is the quieter phase, with less cloud cover and reduced rainfall. The atmosphere tends to be drier.

As these phases move across the globe, they influence weather patterns in the areas they pass over.

3. Global Influence:

- **Tropical Regions:** The MJO can impact monsoon systems, especially in South Asia and Southeast Asia, as well as tropical cyclones.
- **Mid-Latitudes:** Though it primarily affects the tropics, the MJO can indirectly influence weather in the mid-latitudes, such as triggering storm systems in North America or Europe.

4. Interaction with Other Climate Systems:

The MJO interacts with other large-scale climate systems like **El Niño** and **La Niña**. For example, during El Niño events, the MJO may strengthen or weaken depending on its phase, which can affect weather predictability.

5. Importance:

- **Weather Forecasting:** Because the MJO moves relatively predictably over weeks, it is useful for medium-range weather forecasts, particularly in regions like India, Australia, and the Americas.
- **Impact on Extreme Weather:** It can trigger or enhance extreme weather events like floods, heatwaves, or tropical cyclones, making it a critical factor in disaster preparedness.

In summary, the MJO is an important part of the Earth's climate system, particularly for influencing tropical weather and global atmospheric circulation.



Earth-2

Accelerated, AI-augmented, high-resolution climate and weather simulations with interactive visualization.



Earth-2 Data: DKRZ/MPI-M, ICON Model

Accelerating Climate Change Predictions

NVIDIA Earth-2 is a full-stack, open platform that accelerates climate and weather predictions with interactive, AI-augmented, high-resolution simulation. It includes physical simulation of numerical models like ICON; machine learning models such as FourCastNet, GraphCast, and Deep Learning Weather Prediction (DLWP) through [NVIDIA Modulus](#); and data federation and visualization with [NVIDIA Omniverse™](#). Running on [NVIDIA DGX™ GH200](#), [HGX™ H100](#), and [OVX™](#) supercomputers, Earth-2 will provide a path to simulate and visualize the global atmosphere at unprecedented speed and scale.

详情请查看 <https://www.nvidia.com/en-us/high-performance-computing/earth-2/>

High-Performance Computing

- Products
- Solutions
- Resources
- For Developers
- Where to Buy

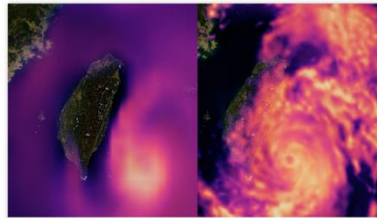
- Overview
- Services
- Adopters
- Demos



Earth-2 Goes Down to Street Level

City-scale simulation data is now incorporated into the Earth-2 planetary digital twin. In this video, we demonstrate the combination of high-resolution simulation data from ICON, WRF, and PALM with Google Photoreal Tiles served by Cesium using the Earth-2 visualization service to help address questions in the urban environment.

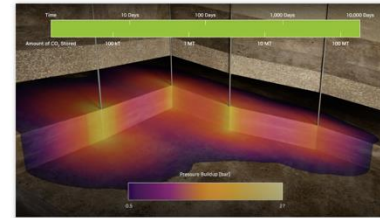
[Watch Demo](#) > (02:50)



Visualize AI-Augmented Weather Simulations

Researchers are using interactive visualization of AI weather forecasts, simulation data, and archived data to analyze extreme weather.

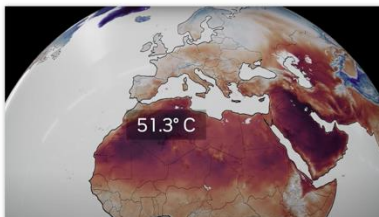
[Watch Generative AI Super-Resolution Local Prediction of Weather Extremes Demo](#) > (01:27)



Accelerating Carbon Capture and Storage with Fourier Neural Operator and NVIDIA Modulus

By accelerating analysis 700,000X, NVIDIA Omniverse and Modulus can help engineers with planning and operating carbon capture and storage, ensuring safe operation and long-term storage and reducing the amount of carbon dioxide released into our atmosphere.

[Watch Carbon Capture Storage Demo](#) > (01:17)



Predicting Extreme Weather Events Three Weeks in Advance With FourCastNet

By running FourCastNet in NVIDIA Modulus, we were able to generate 21-day weather trajectories of 1,000 ensemble members in a tenth of the time it previously took to do a single ensemble—and with 1,000X less energy consumption.

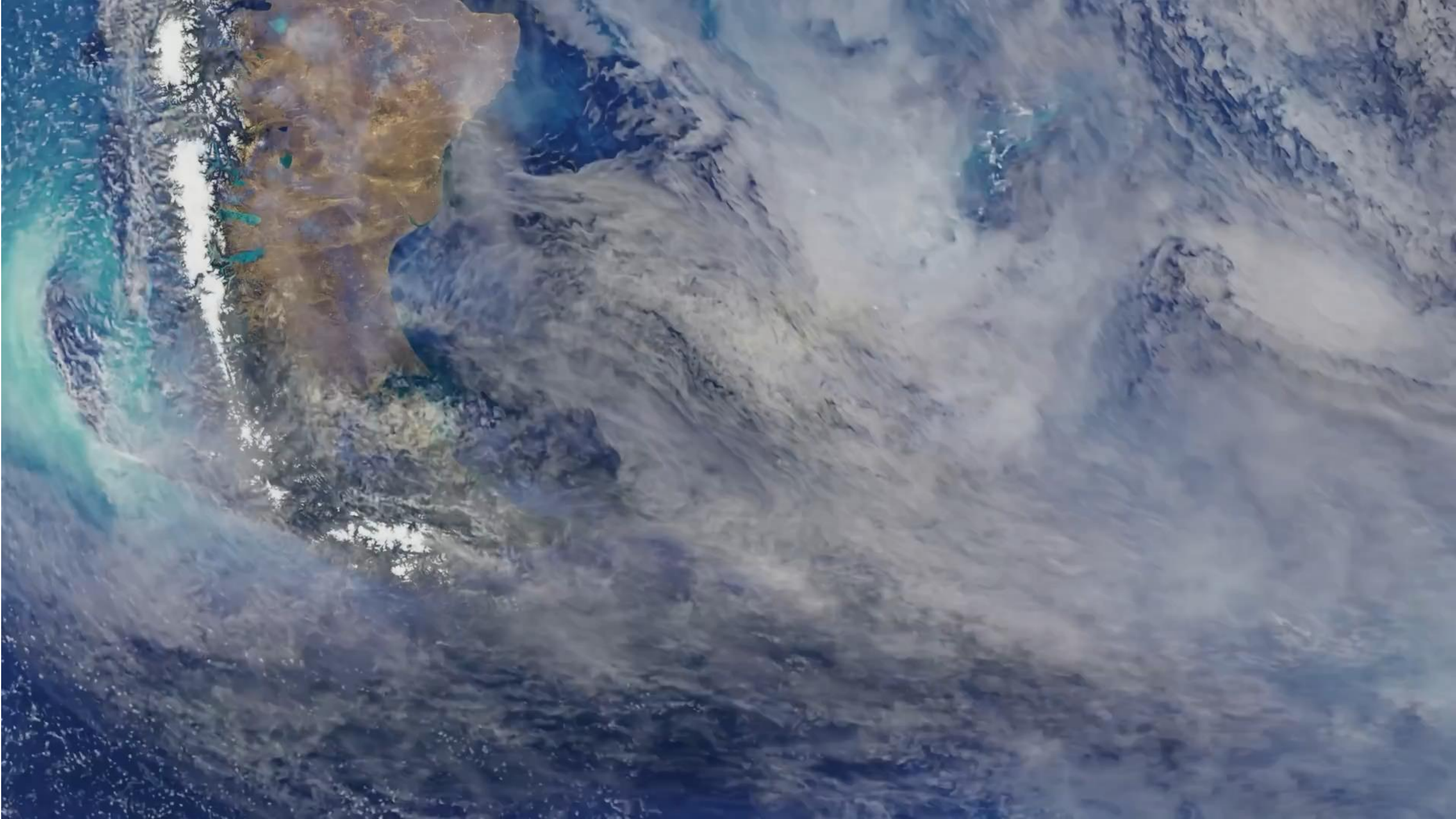
[Watch Predict Extreme Weather Events with FourCastNet Demo](#) > (01:40)



Interactive Visualization of High-Resolution, Global-Scale Climate Data in the Cloud

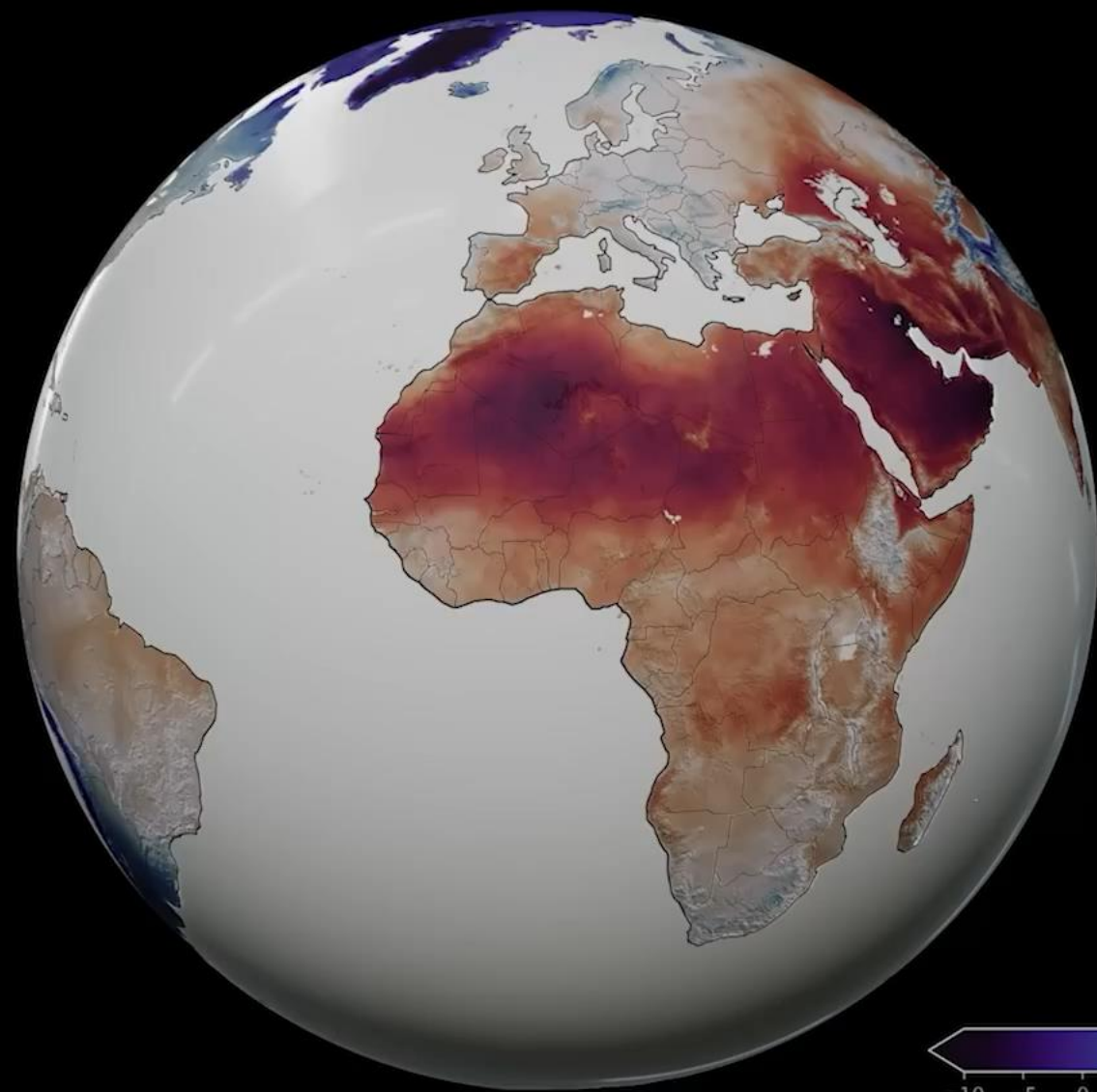
Built on NVIDIA Omniverse and the OpenUSD 3D framework, the Earth-2 platform enables aggregation and visualization of diverse, global-scale climate simulation and geospatial datasets. Made possible with cloud-native technology, visualizations can be explored by anyone around the globe.

[Watch Global Climate Visualization Demo](#) > (01:31)

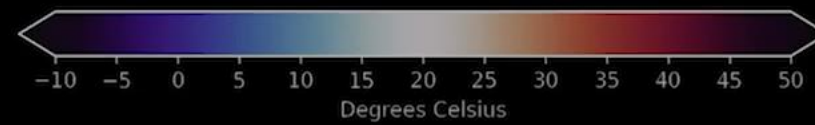








25-June-2018



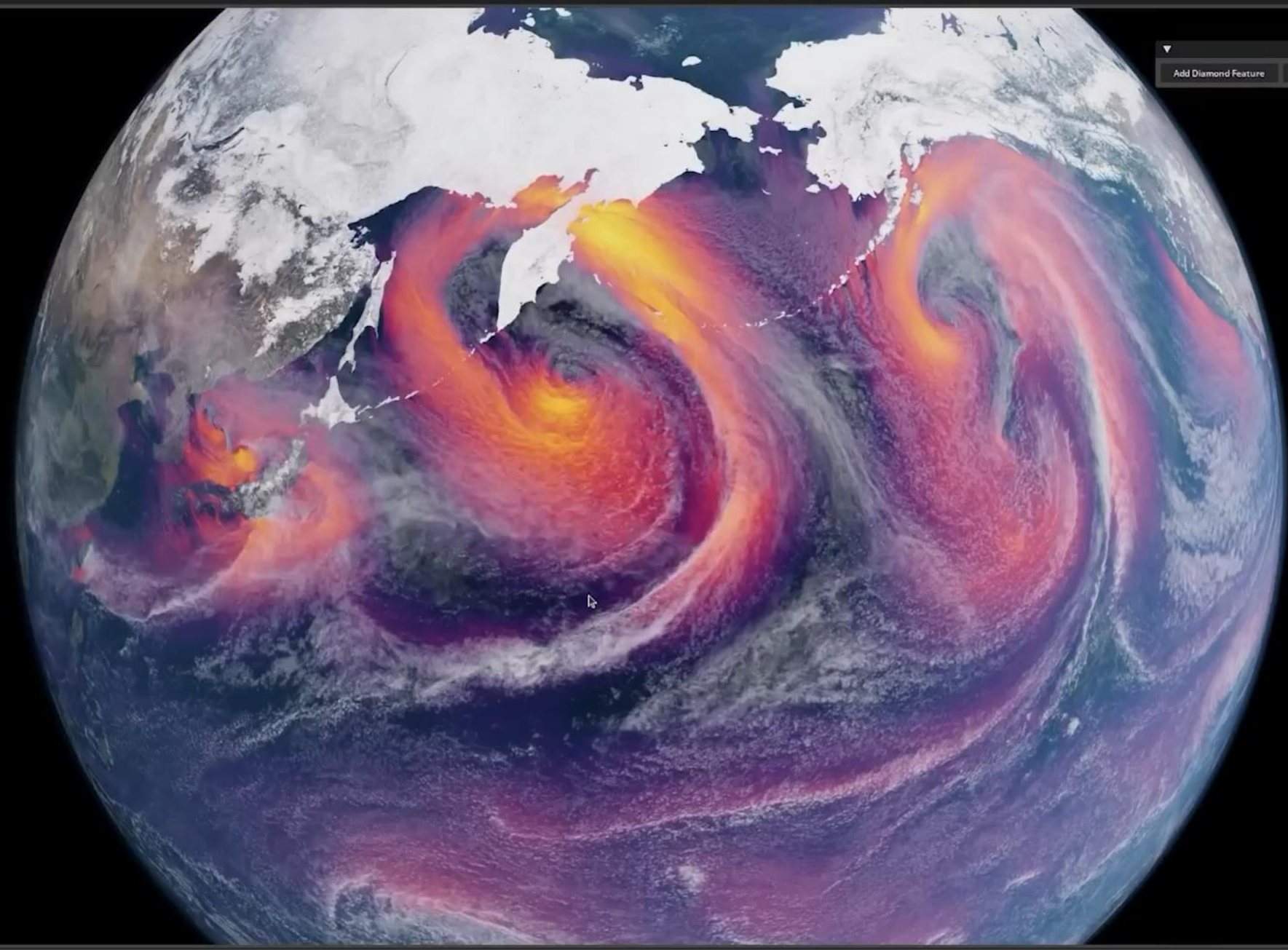


Image Feature Tests

Add Diamond Feature Add LatLong Feature Clear Features



盘古大模型：解决行业难题，释放AI生产力

盘古大模型：解决行业难题，释放AI生产力

L2 场景模型

X

传送带异物检测

政务热线

台风路径预测

自动驾驶研发

报告解读

数字人直播

智能测试

+

重介选煤洗选

城市事件处理

降水预测

车辆辅助设计

辅助医疗

智能问答

智能运维

N

L1 行业大模型

盘古矿山大模型

盘古政务大模型

盘古气象大模型

盘古汽车大模型

盘古医学大模型

盘古数字人大模型

盘古研发大模型

+

L0 基础大模型

盘古自然语言大模型

盘古多模态大模型

盘古视觉大模型

盘古预测大模型

盘古科学计算大模型

5

大模型行业开发套件

重塑行业

技术扎根

开放同飞

[nature](#) > [articles](#) > article

Article | [Open access](#) | Published: 05 July 2023

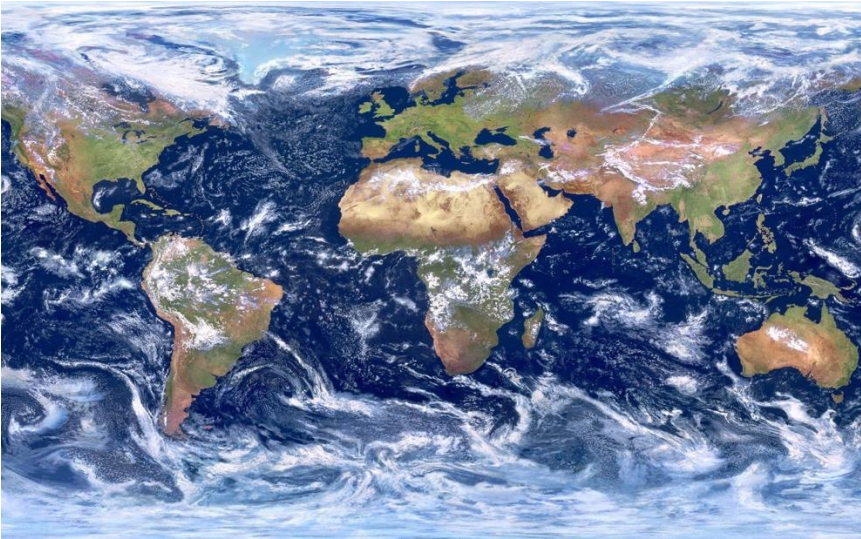
Accurate medium-range global weather forecasting with 3D neural networks

[Kaifeng Bi](#), [Lingxi Xie](#), [Hengheng Zhang](#), [Xin Chen](#), [Xiaotao Gu](#) & [Qi Tian](#) 

[Nature](#) **619**, 533–538 (2023) | [Cite this article](#)

209k Accesses | **191** Citations | **1728** Altmetric | [Metrics](#)

Question to be solved



What happened in 24 hrs



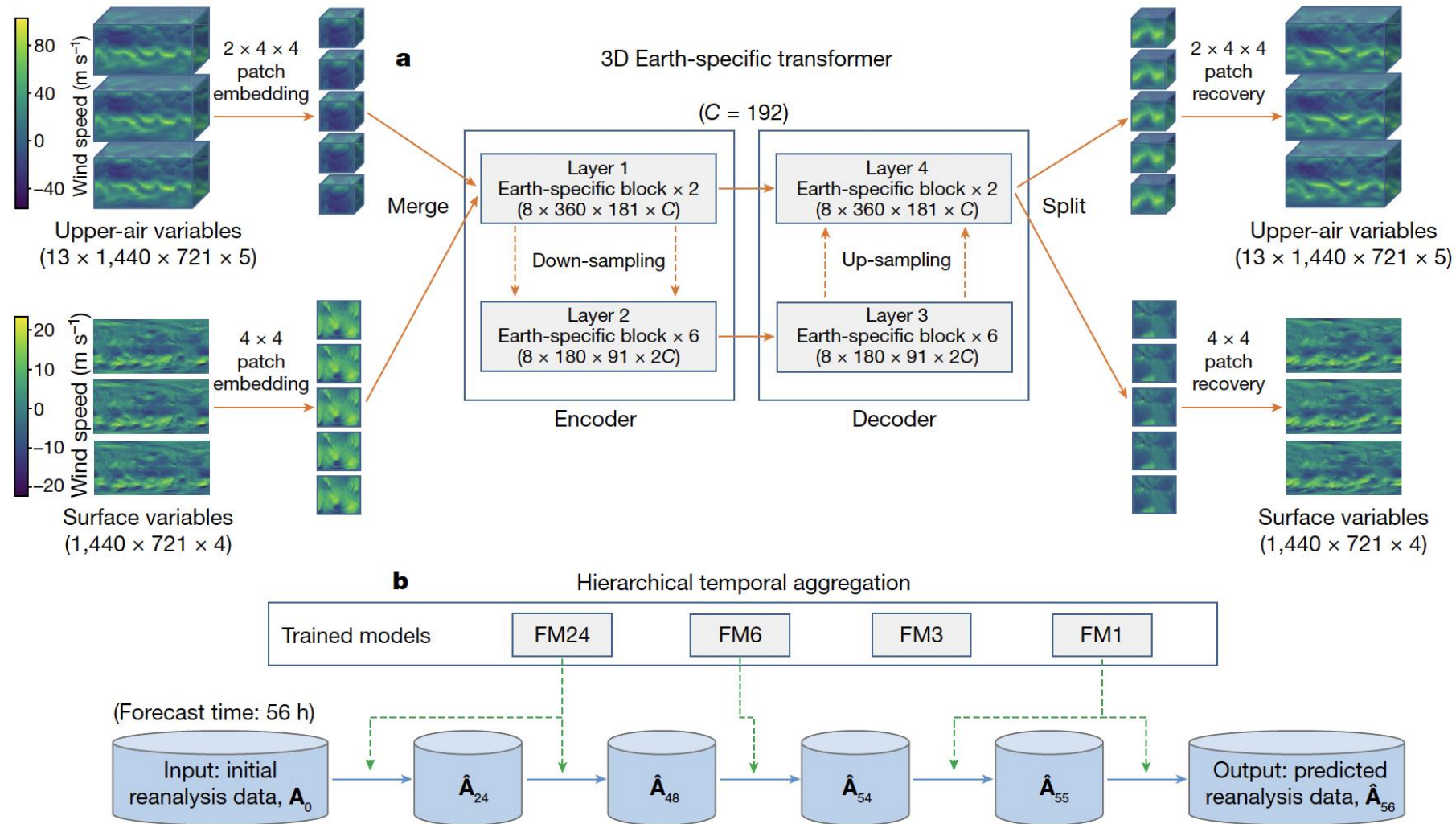


Fig. 1 | Network training and inference strategies. a, 3DEST architecture. Based on the standard encoder–decoder design of vision transformers, we adjusted the shifted-window mechanism¹⁹ and applied an Earth-specific positional bias. **b**, Hierarchical temporal aggregation. Once given a lead time,

we used a greedy algorithm to perform forecasting with as few steps as possible. We use FM1, FM3, FM6 and FM24 to indicate the forecast models with lead times being 1 h, 3 h, 6 h or 24 h, respectively. \mathbf{A}_0 is the input weather state and $\hat{\mathbf{A}}_t$ denotes the predicted weather state at time t (in hours).

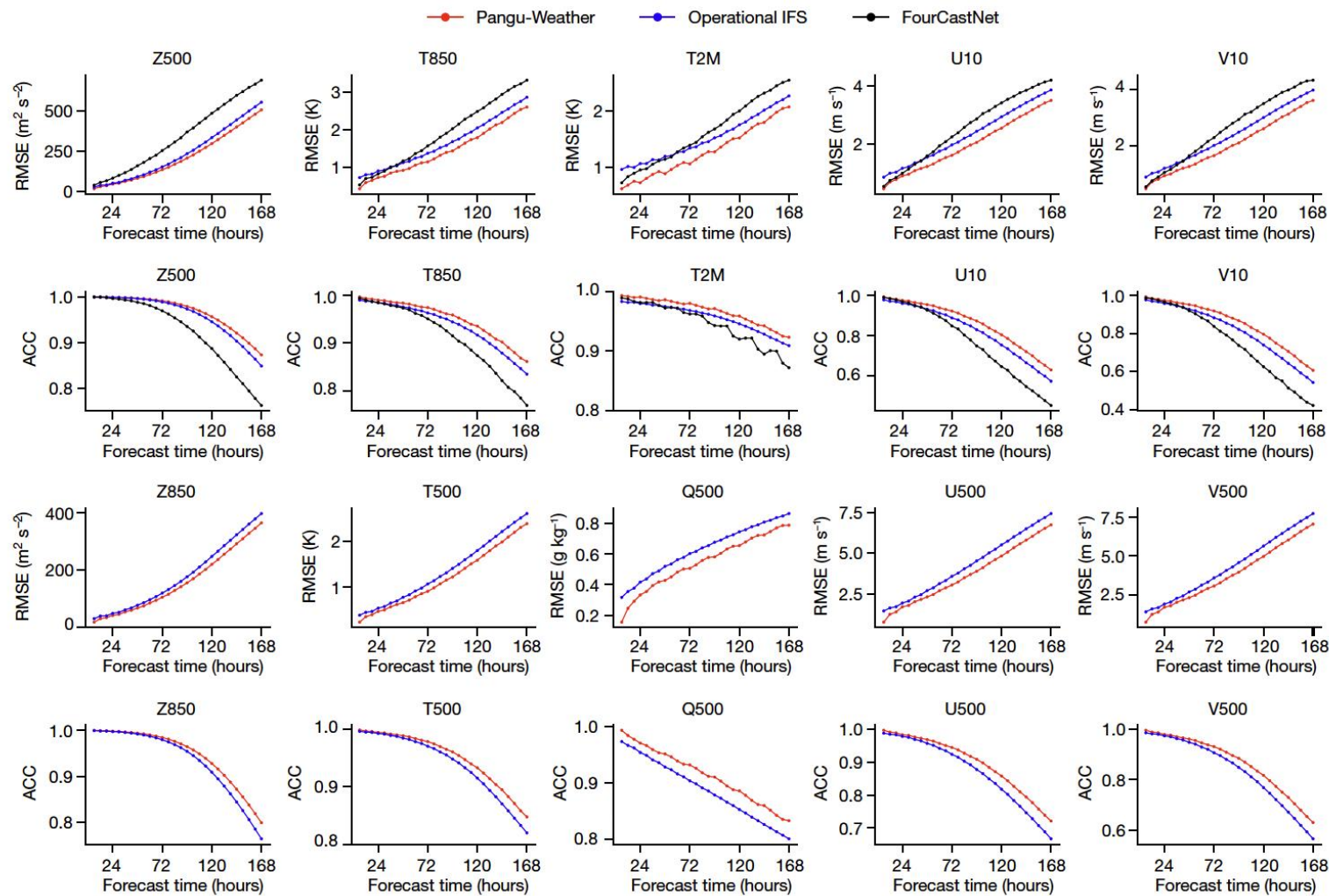


Fig. 2 | Pangu-Weather produces higher accuracy than the operational IFS and FourCastNet in deterministic forecasts on the ERA5 data. Ten variables were compared in terms of latitude-weighted RMSE (lower is better) and ACC (higher is better), where the first five variables were reported in FourCastNet and the last five were not. Here, Z500, T500, Q500, U500 and V500 indicate the

geopotential, temperature, specific humidity, and the u -component and v -component of wind speed at 500 hPa, respectively. Z850 and T850 indicate the geopotential and temperature at 850 hPa, respectively. T2M indicates the 2-m temperature, and U10 and V10 indicate the u -component and v -component of 10-m wind speed, respectively.

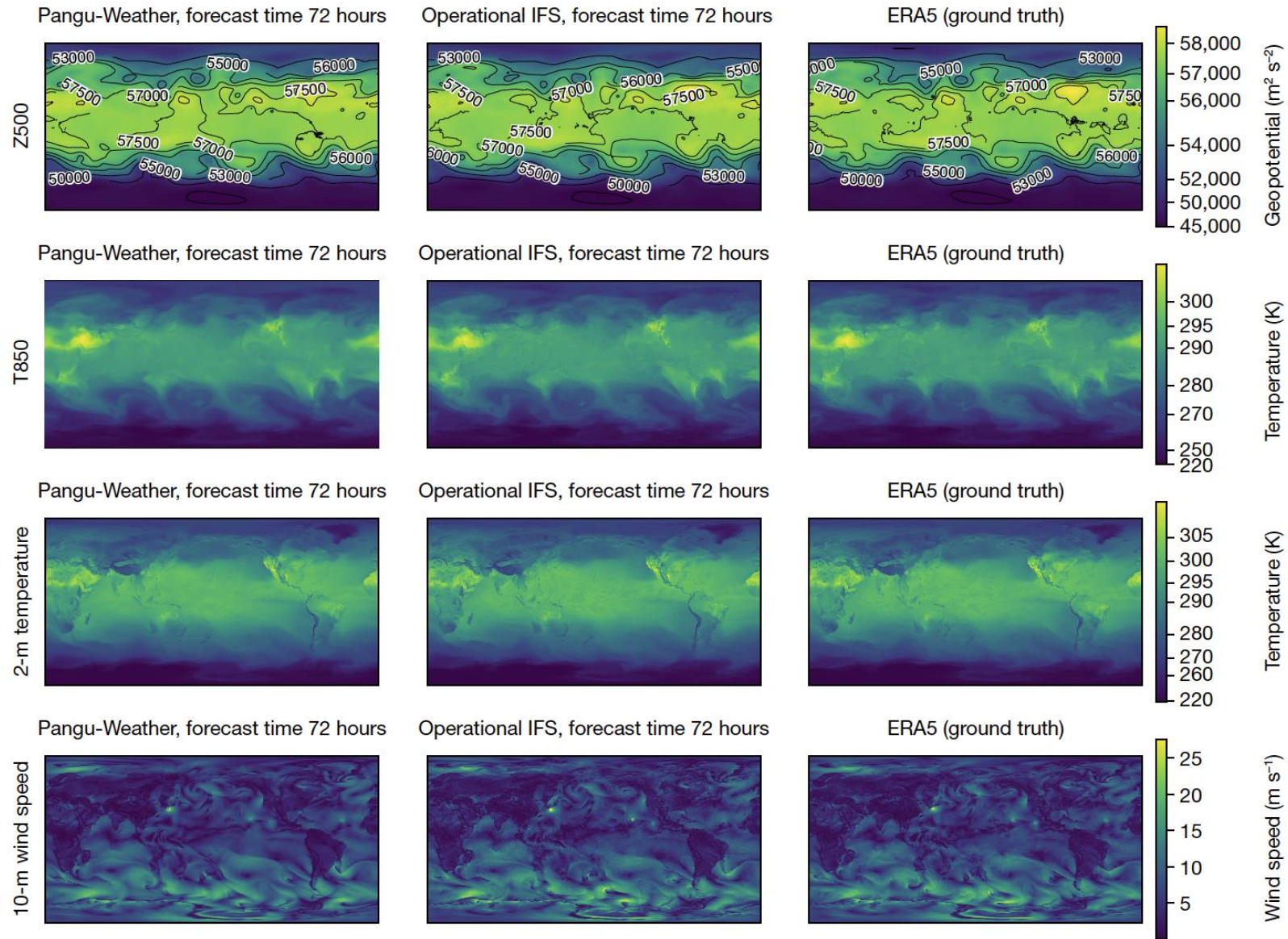


Fig. 3 | Visualization of forecast results. The 3-day forecast of two upper-air variables (Z500 and T850) and two surface variables (2-m temperature and 10-m wind speed). For each case, Pangu-Weather (left), the operational IFS³

(middle) and the ERA5 ground truth¹⁸ (right) are shown. For all cases, the input time is 00:00 UTC on 1 September 2018.

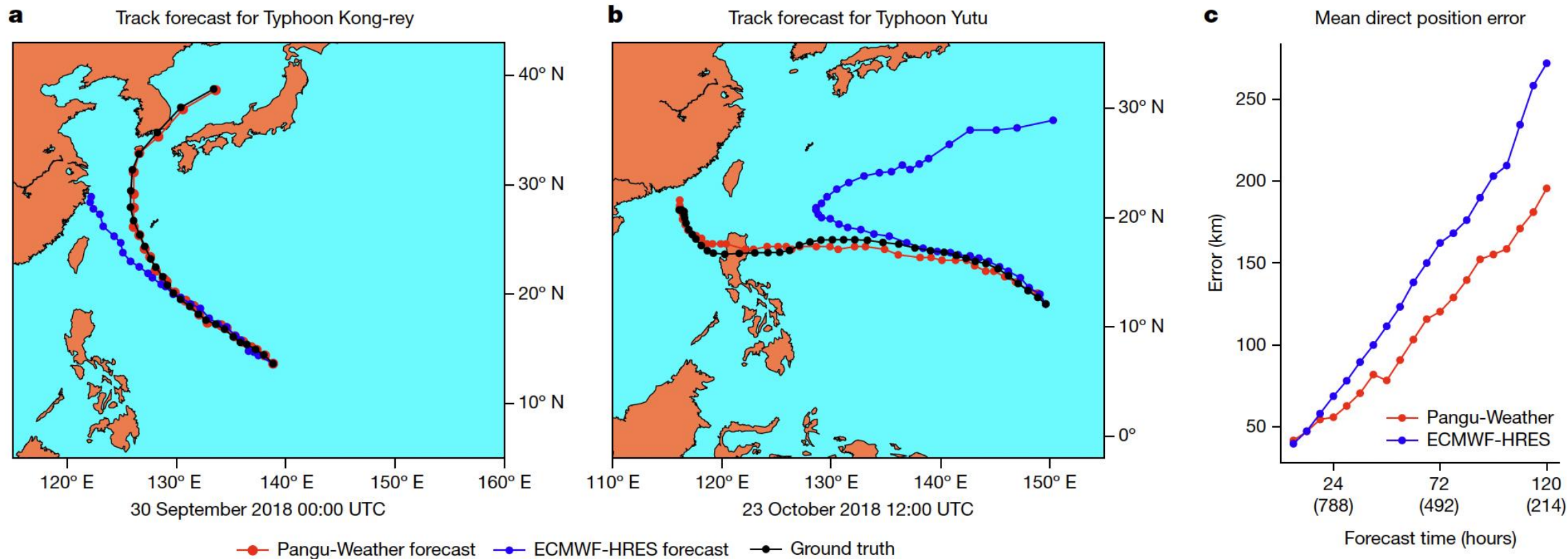


Fig. 4 | Pangu-Weather is more accurate at early-stage cyclone tracking than ECMWF-HRES. **a,b**, Tracking results for two strong tropical cyclones in 2018, that is, Typhoon Kong-rey (2018–25) and Yutu (2018–26). The initial time point is shown below each panel. The time gap between neighbouring dots is 6 h. Pangu-Weather forecasts the correct path of Yutu (that is, it goes to the Philippines) at 12:00 UTC on 23 October 2018, whereas ECMWF-HRES obtains the same conclusion 2 days later, before which it predicts that Yutu will make

a big turn to the northeast. **c**, A comparison between Pangu-Weather and ECMWF-HRES in terms of mean direct position error over 88 cyclones in 2018. Each number in brackets in the *x*-axis indicates the number of samples used to calculate the average. For example, '(788)' means that there are in total 788 initial points from which the typhoon lasts for at least 24 hours, and the 788 direct position errors of Pangu-Weather and ECMWF-HRES were averaged into the final results. Panels **a** and **b** were plotted using the Matplotlib Basemap toolkit.

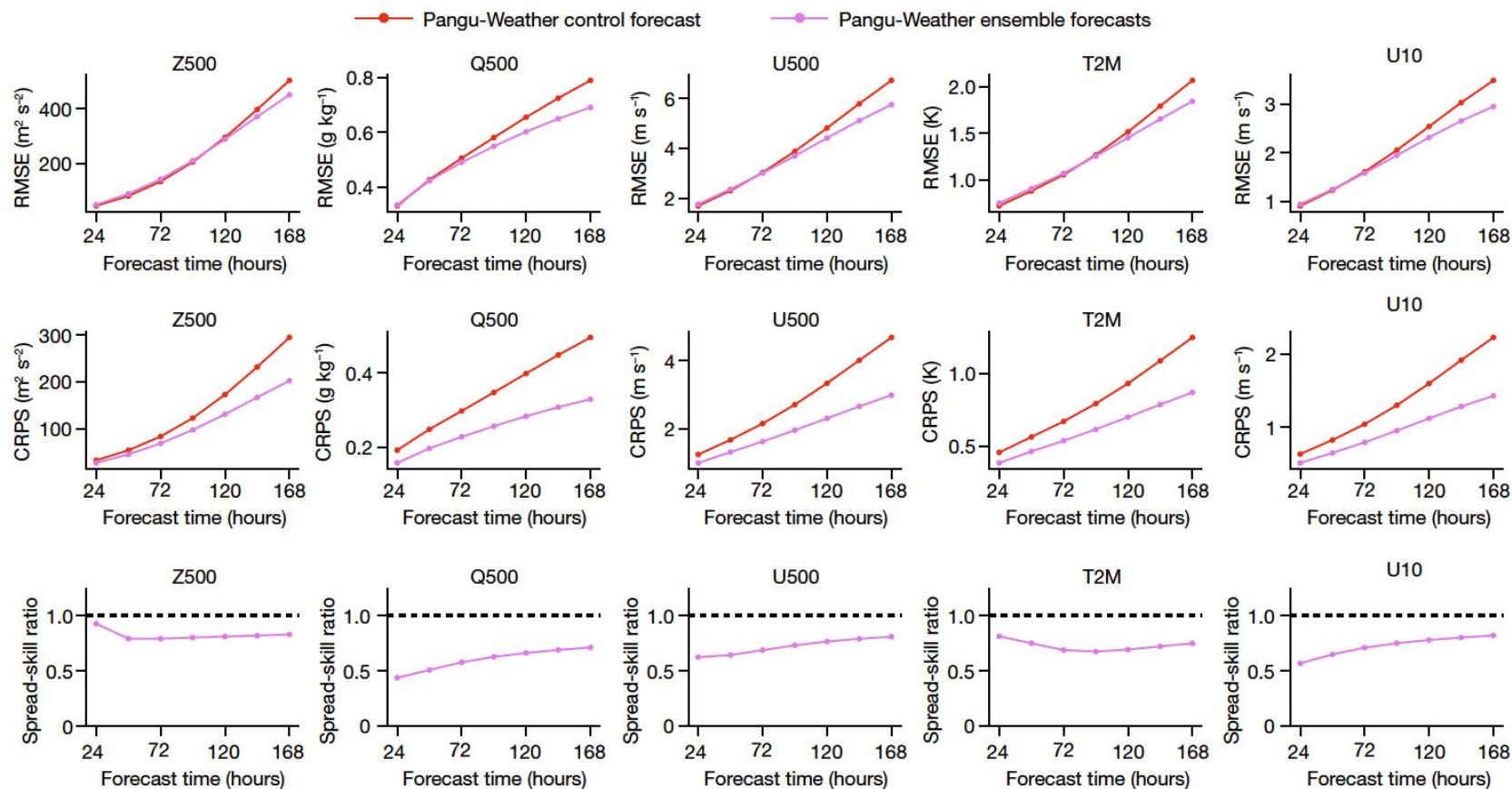


Fig. 5 | Ensemble forecast results of Pangu-Weather. The RMSE of the ensemble mean forecast (lower is better) for three upper-air variables (Z500, Q500 and U500) and two surface variables (T2M and U10). We also followed a recent work³⁵ to plot two metrics, the CRPS (lower is better) and the spread-skill ratio (an ideal ensemble model produces spread-skill ratios of 1.0, shown as the

dashed lines), which further demonstrate the properties of our ensemble forecast results. Here, Z500, Q500 and U500 indicate the geopotential, temperature and the u -component of wind speed at 500 hPa, respectively. T2M indicates the 2-m temperature and U10 indicates the u -component of 10-m wind speed.

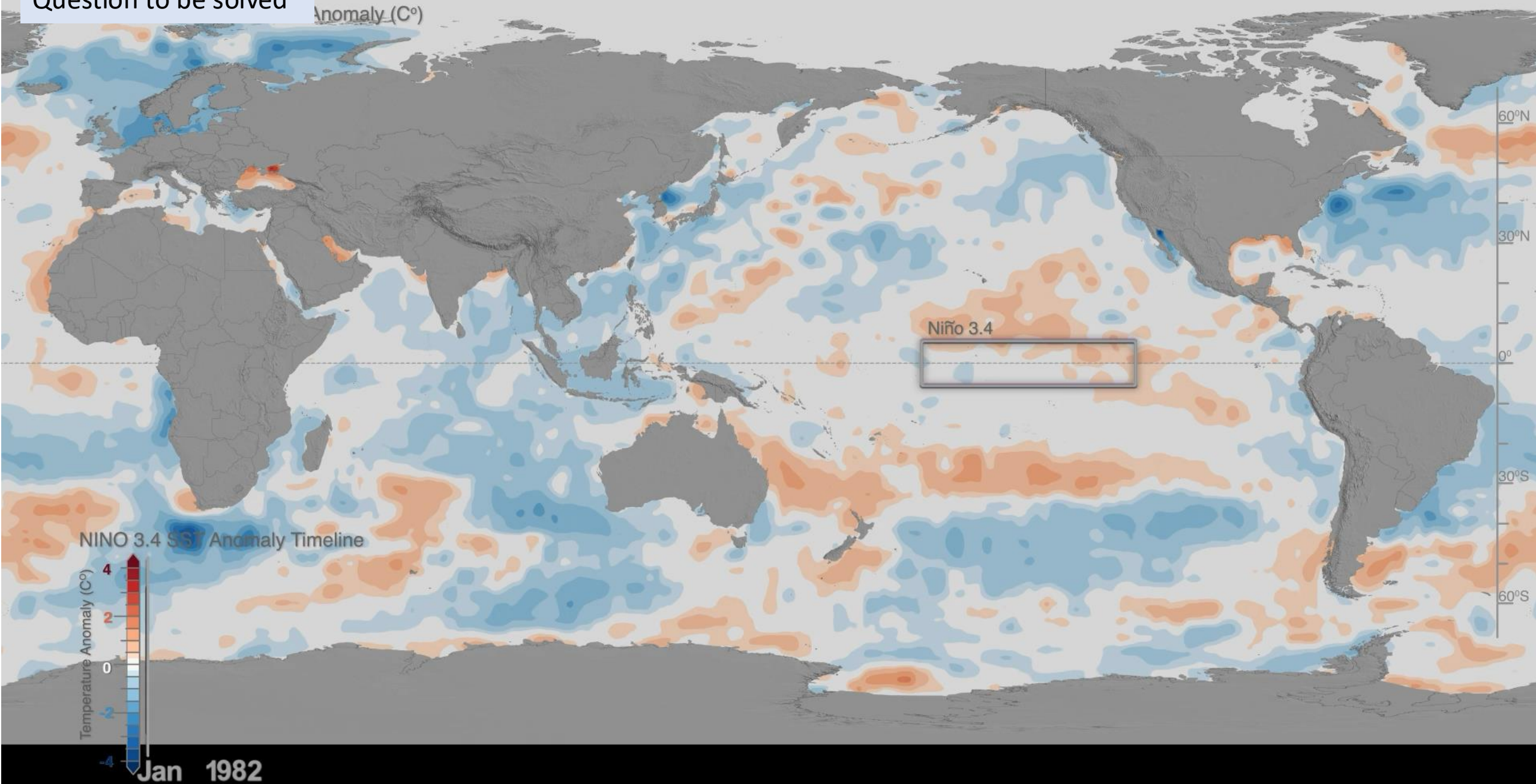
LETTER

<https://doi.org/10.1038/s41586-019-1559-7>

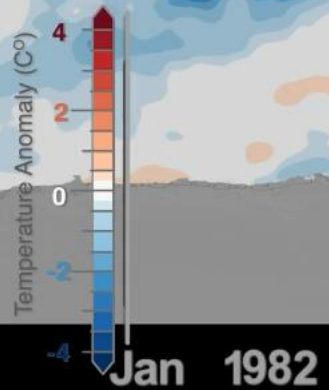
Deep learning for multi-year ENSO forecasts

Yoo-Geun Ham^{1*}, Jeong-Hwan Kim¹ & Jing-Jia Luo^{2,3}

Question to be solved



Niño 3.4 SST Anomaly Timeline



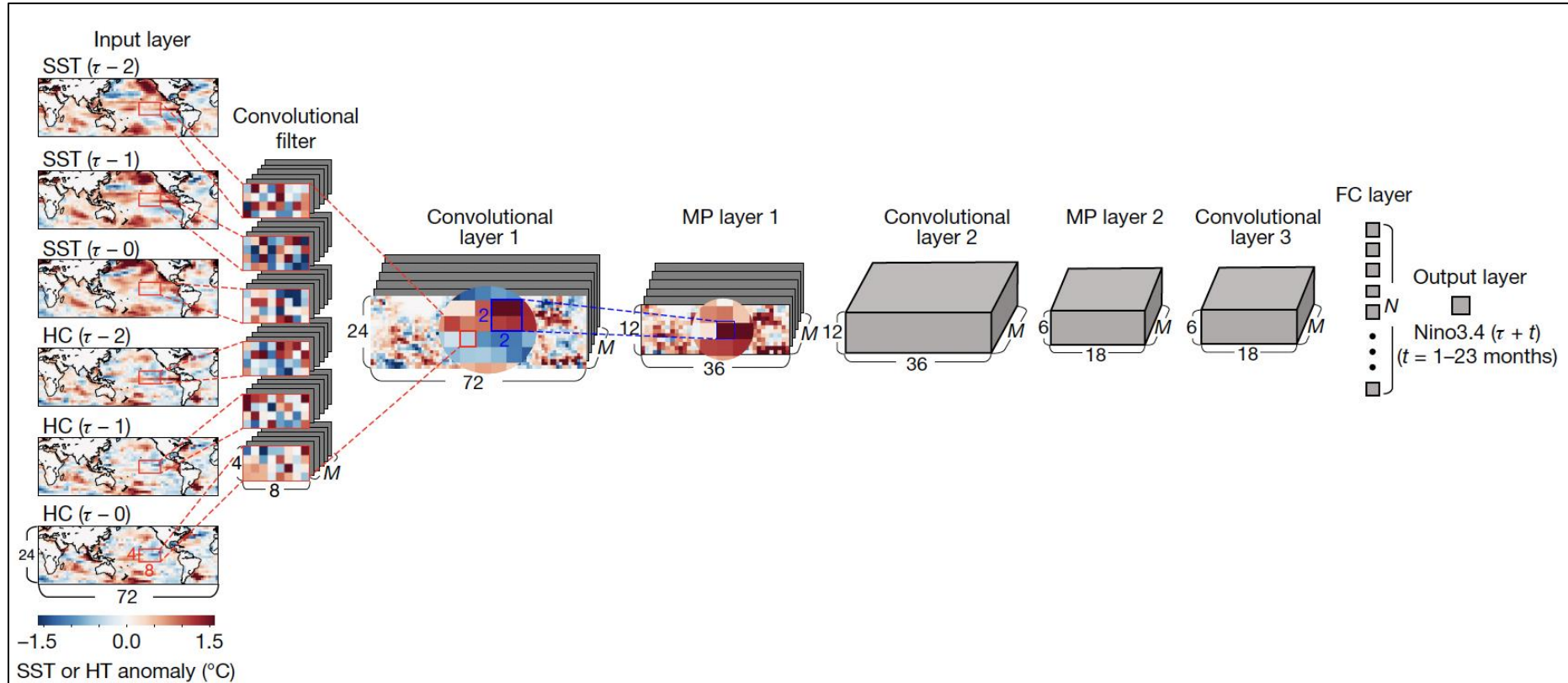


Fig. 1 | Architecture of the CNN model used for the ENSO forecasts. The CNN model consists of one input layer (the predictor), three convolutional layers, two max-pooling (MP) layers, one fully connected (FC) layer, and one output layer (the predictand). The variables of the input layer correspond to the SST (in units of $^{\circ}\text{C}$) and the oceanic heat content (HC, in units of $^{\circ}\text{C}$) anomaly maps from time $\tau - 2$ months to time τ (in months), between 0° – 360° E and 55° S– 60° N. The three-month-averaged Nino3.4 index from time $\tau + 1$ month to $\tau + 23$ months is used as a variable for the

output layer. The red boxes and lines highlight the convolutional filter and convolutional process, respectively; and the blue box and lines indicate the max-pooling window and the max-pooling process, respectively. M denotes the number of feature maps, while N denotes the number of neurons in the FC layer, which are set to be either 30 or 50 in this study. The global map is generated in Matplotlib³¹. The x and y dimensions of the map in each layer are denoted (6, 12, 18, 24, 36, 72) next to the map.

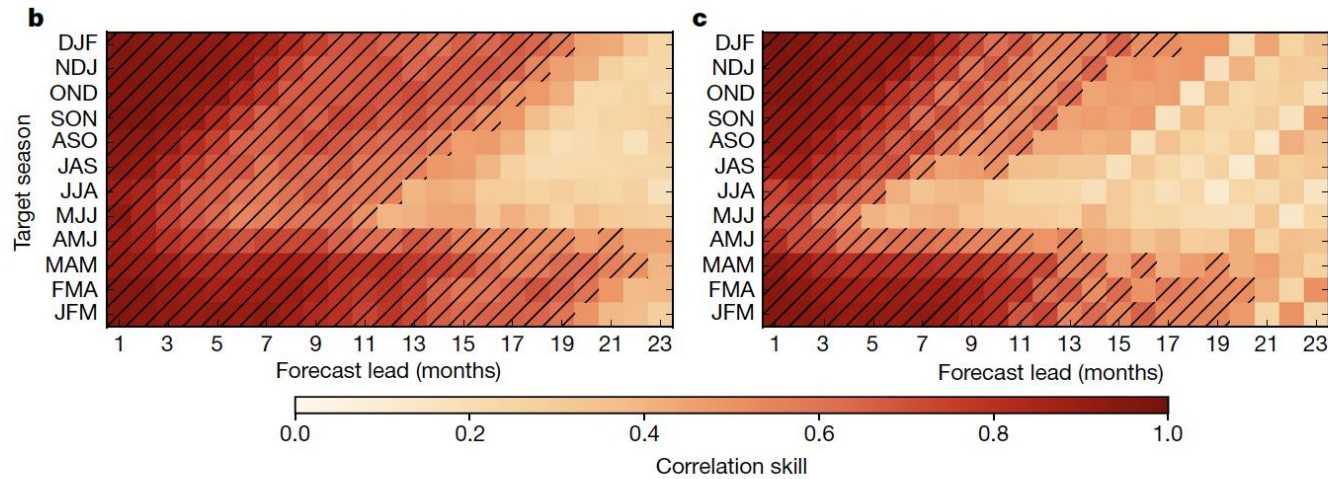
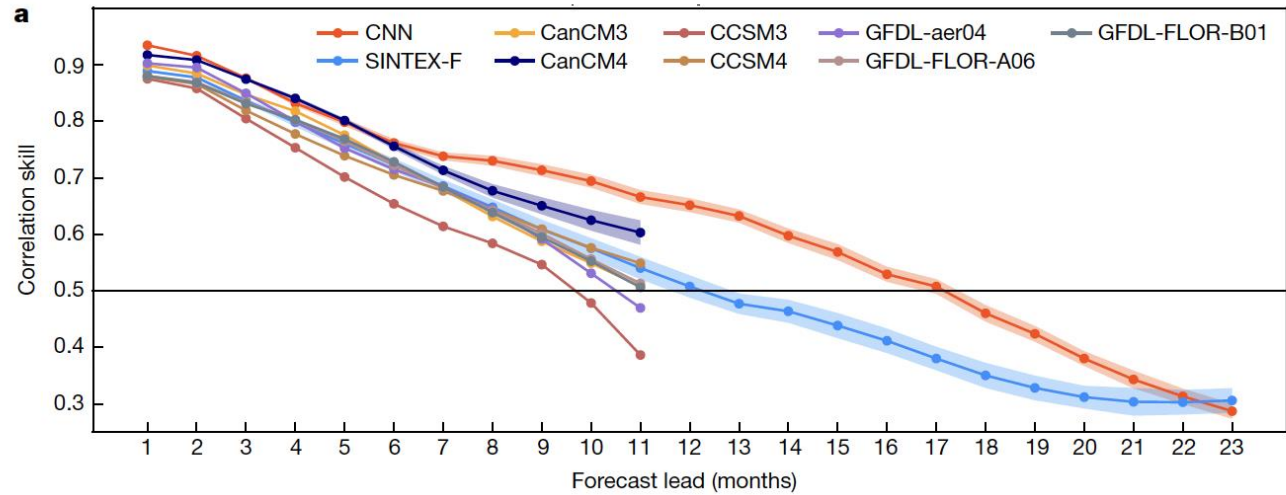


Fig. 2 | ENSO correlation skill in the CNN model. **a**, The all-season correlation skill of the three-month-moving-averaged Nino3.4 index as a function of the forecast lead month in the CNN model (red), SINTEX-F dynamical forecast system (blue), and dynamical forecast systems included in the North American Multi-Model Ensemble (NMME) project (the other colours). The validation period is between 1984 and 2017. The shading around the lines for CNN, CanCM4 and SINTEX-F denotes the 95% confidence interval, based on the bootstrap method. **b**, **c**, The correlation skill of the Nino3.4 index targeted to each calendar month in the CNN model (**b**) and the SINTEX-F dynamical forecast system (**c**). Hatching highlights the forecasts with correlation skill exceeding 0.5.

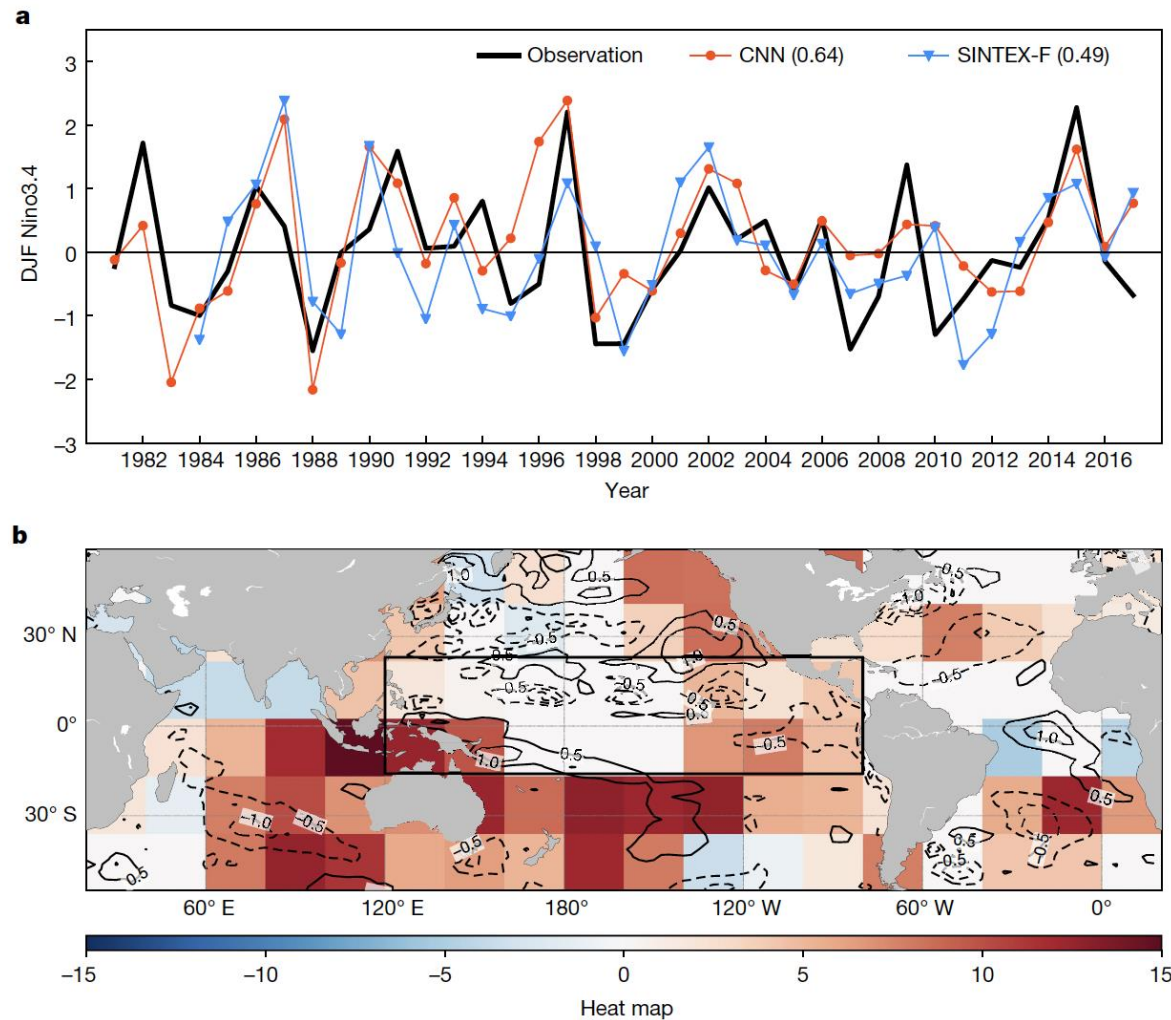


Fig. 3 | Physical interpretation of the CNN model ENSO forecasts.

a, Time series of DJF season Nino3.4 indexes for an 18-month-lead forecast using the CNN model (red) and the SINTEX-F model (blue). The observed Nino3.4 index is also shown (black). **b**, The heat map (shading), and SST or heat content anomalies (contours; dashed contours denote the negative values of SST or HC anomalies and solid contours denote the positive values of SST or HC anomalies) for the MJJ season in 1996 used

for the prediction of the DJF season during the 1997/98 El Niño event. Heat content anomalies (in units of °C) are shown over the tropical Pacific (within the black box), while SST anomalies (in units of °C) are denoted outside the tropical Pacific. Only the values with over 95% confidence level based on Student's *t*-test using the standard deviation of the heat map during 1984–2017 are shaded. The global map is generated in Matplotlib³¹.

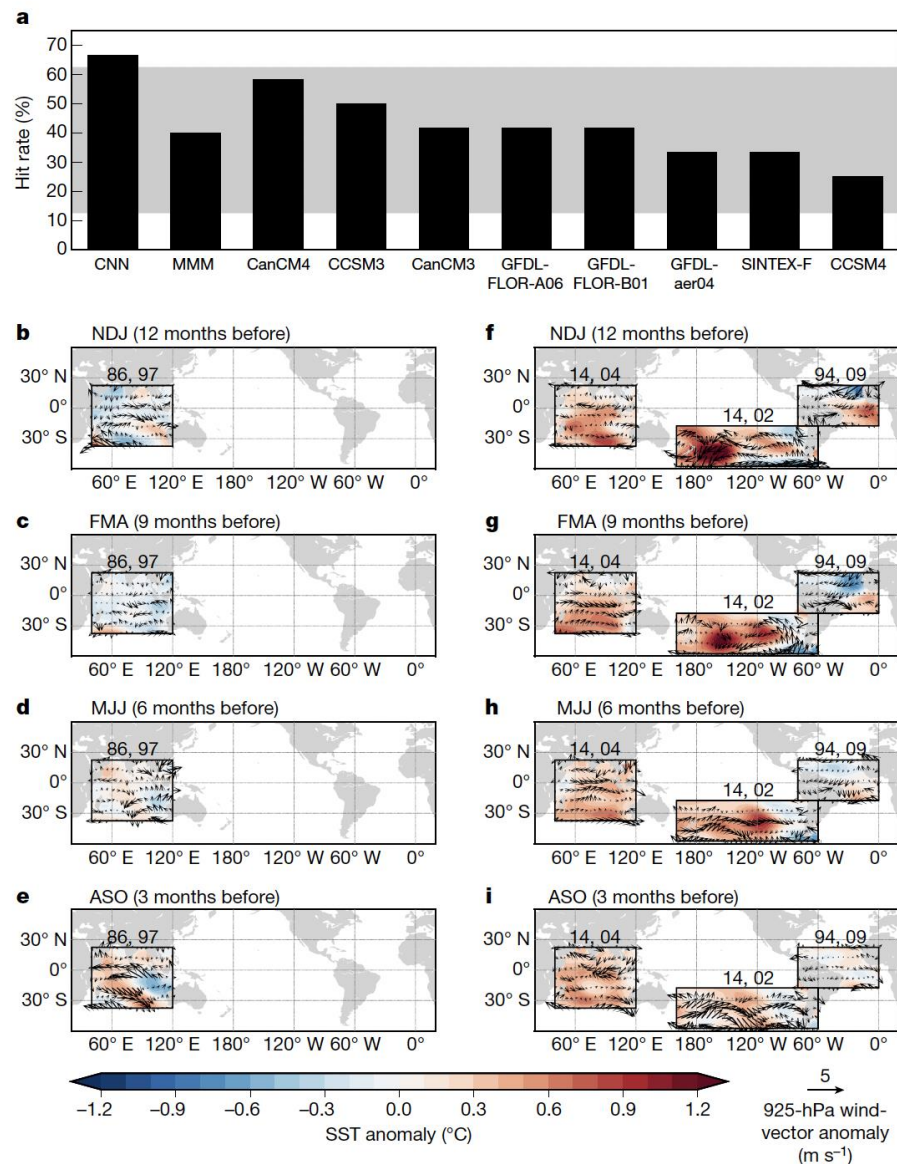


Fig. 4 | CNN forecast system for the prediction of El Niño type. **a**, The hit rate of the 12-month-lead prediction of El Niño types (EP-type, CP-type or mixed-type) during the DJF season using the CNN model and the SINTEX-F model, and models in the NMME project, for the validation period 1984–2017. The multi-model mean (MMM) of the hit rate is also shown. The grey shading denotes the 95% confidence interval of the random forecasts. **b–e**, The SST and 925-hPa wind-vector anomalies for the NDJ (**b**), FMA (**c**), MJJ (**d**) and ASO (**e**) seasons during the El

Niño events when the area-averaged heat map value is greatest for the EP-type El Niño. **f–i**, As for panels **b–e**, but for the CP-type El Niño. The numbers above each ocean basin box denote the El Niño years selected for drawing the anomalies in each basin, and for each type of El Niño event. The anomalies are visible only when the area-averaged heat map value in any ocean basin exceeds one standard deviation for El Niño events for all ocean basins (see Extended Data Fig. 5 for details). The global map is generated in Matplotlib³¹.