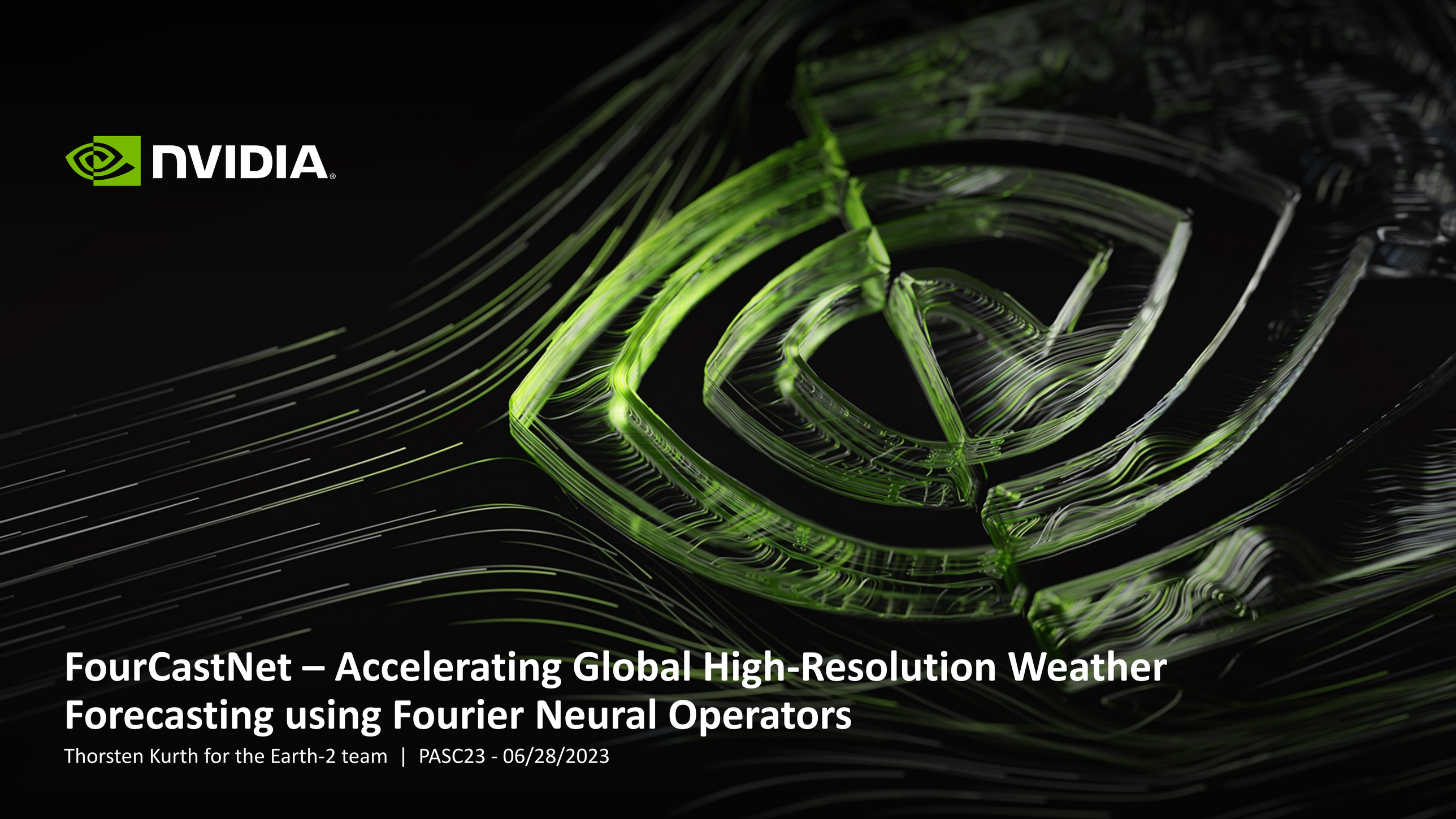


FourCastNet – Accelerating Global High-Resolution Weather Forecasting using Fourier Neural Operators

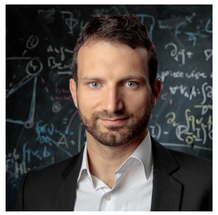
Thorsten Kurth for the Earth-2 team | PASC23 - 06/28/2023

The FourCastNet Team

Jaideep P.
NVIDIA

Shashank S.
LBL

Peter H.
LBL



Boris B.
NVIDIA

Morteza M.
NVIDIA

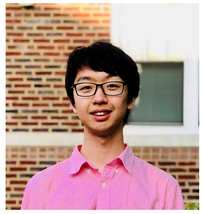
Thorsten K.
NVIDIA

Noah B.
NVIDIA

Sanjeev R.
U Michigan

Ashesh C.
Rice U.

David H.
NVIDIA



Zongyi L.
Caltech

Yair C.
NVIDIA

Kamyar A.
Purdue

Pedram H.
Rice U.

Andrea M.
Apple

David P.
NVIDIA

Justin L.
NVIDIA

Karthik K.
NVIDIA

Christian H.
NVIDIA

Max B.
NVIDIA

Anima A.
NVIDIA / Caltech

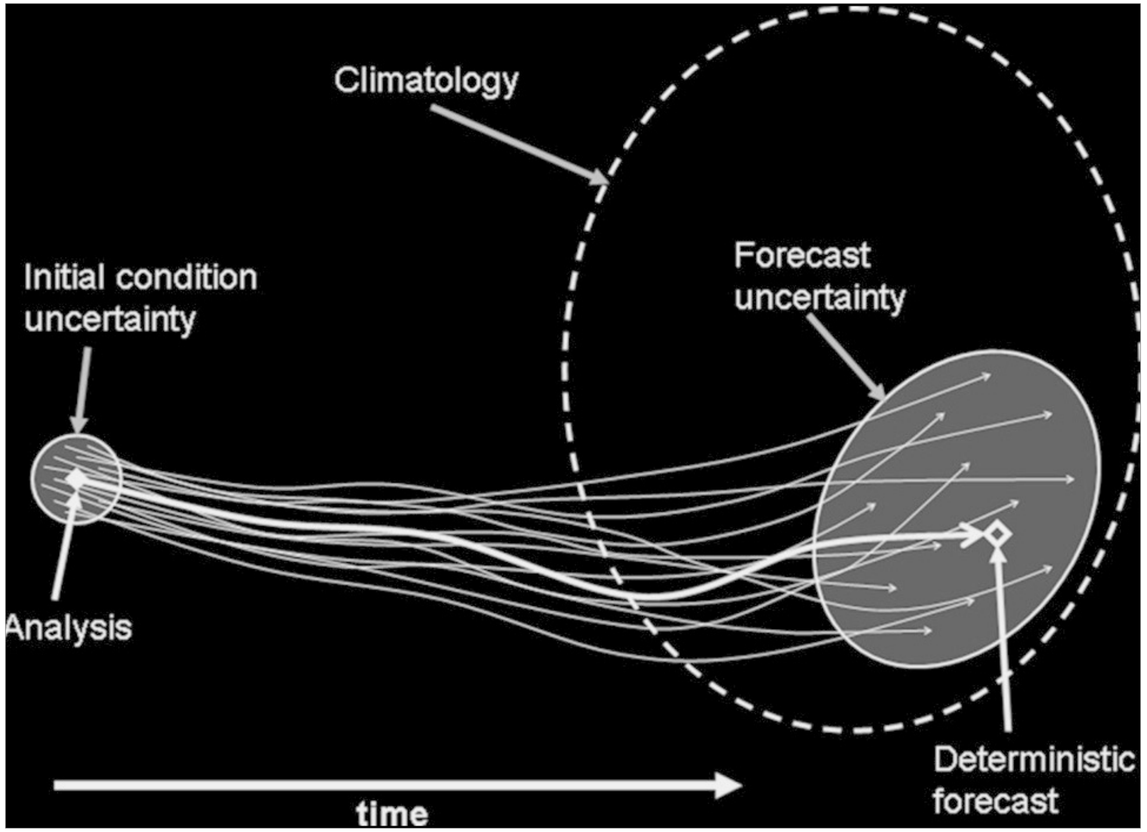
Mike P..
NVIDIA

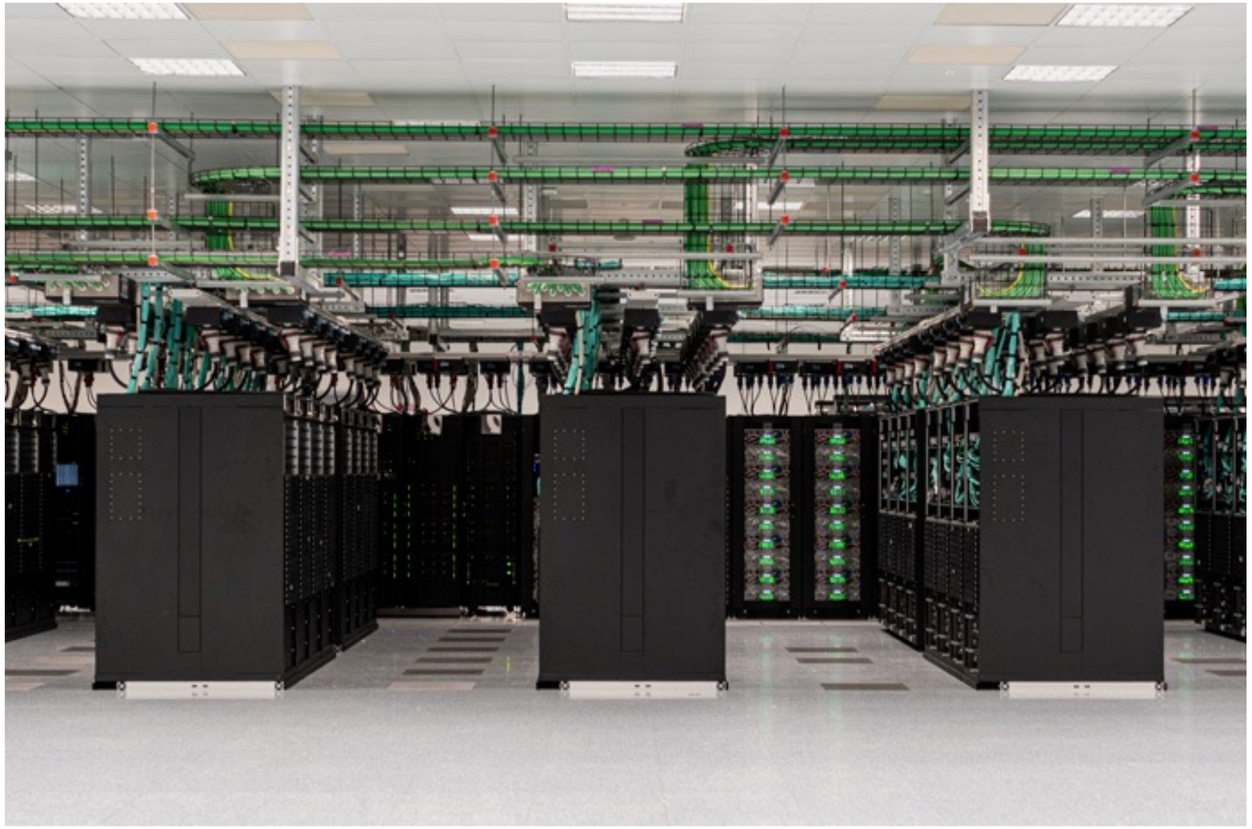
Outline

- Data-driven Weather Prediction
- Adaptive Fourier Neural Operator
- From Weather Prediction to Climate Science:
  Fourier Neural Operators on the Sphere
- Outlook

Traditional Deterministic Numerical Weather Prediction (NWP)

Solving equations of motion for incompressible fluid on a rotating sphere + lots of tricks



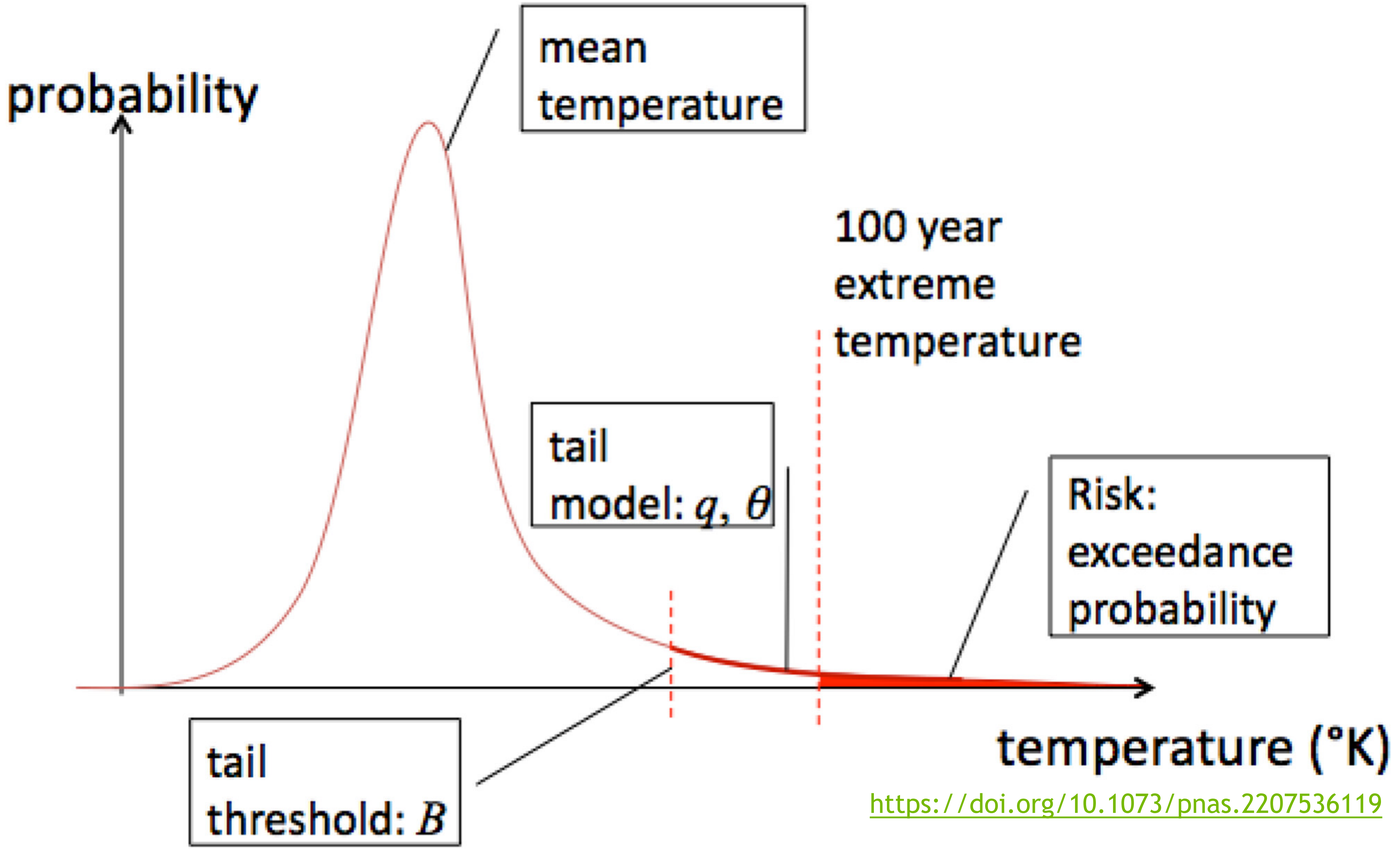


HPC-intensive: European Center for Medium Range Weather Forecasting

- Numerical models and heuristics developed over decades, requiring lot of domain expertise to implement new features/physics
- **Producing long prediction rollouts/large ensembles is computationally demanding**

Massive Ensembles required to capture Low Likelihood/High Impact Extremes

Multiple atmospheric rivers landing in California Dec-Jan 2023

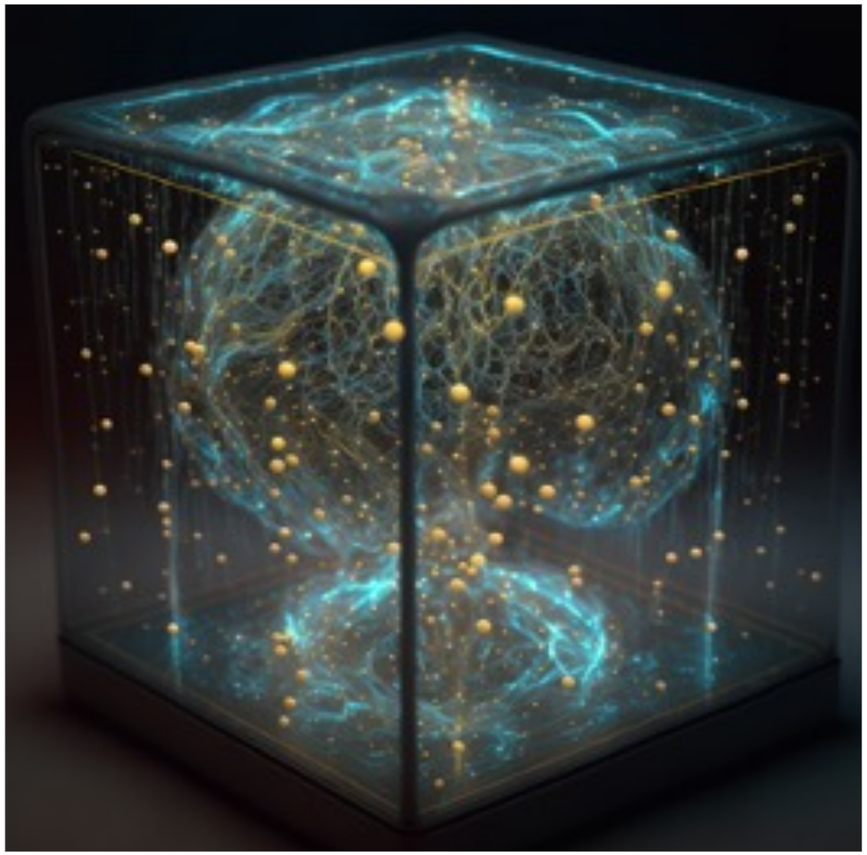


https://doi.org/10.1073/pnas.2207536119

Capture the exponentially suppressed tail: huge ensemble and thus fast sampling needed!

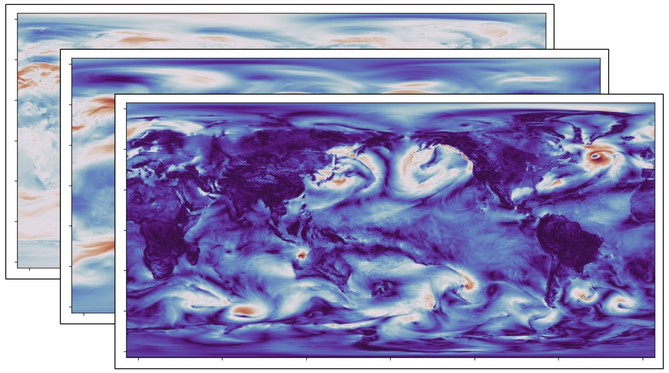
Data-driven Weather Prediction, what's different?

Training transformers to predict the next few weeks temperatures, winds, surface pressure with data.



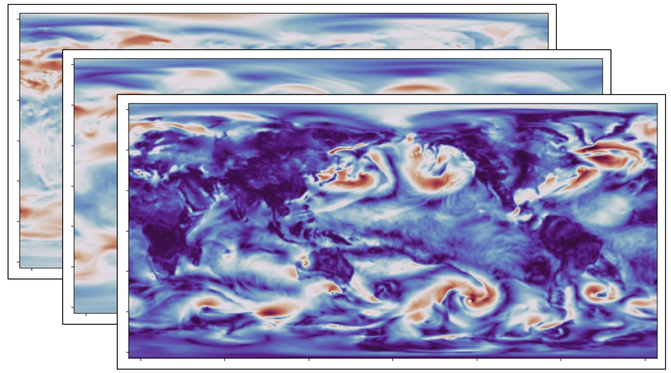
- Very similar to time series image prediction task
  (e.g., in movies, video games)
- Instead of RGB channels we use physical fields
  (temperatures, winds, pressures…)
- # training samples scales with temporal resolution and length of recording
- Can be stood up by small teams within tech companies (lot of engagement from a broader community with great results,
  e.g. Pangu-Weather, Graphcast, FuXi)
- Does not require in-depth domain science knowledge
  to modify model or parameters
- Is producing skill gains rapidly
- **Producing long prediction rollouts and large ensembles is cheap**

Data-driven Weather Prediction in a Nutshell



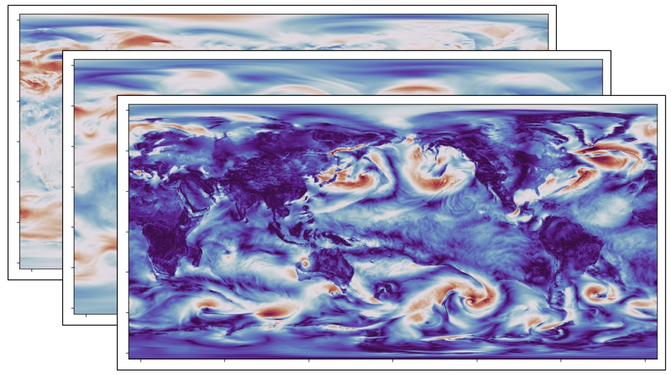
state at time t

Deep Neural Network



predicted state at time t + dt

loss



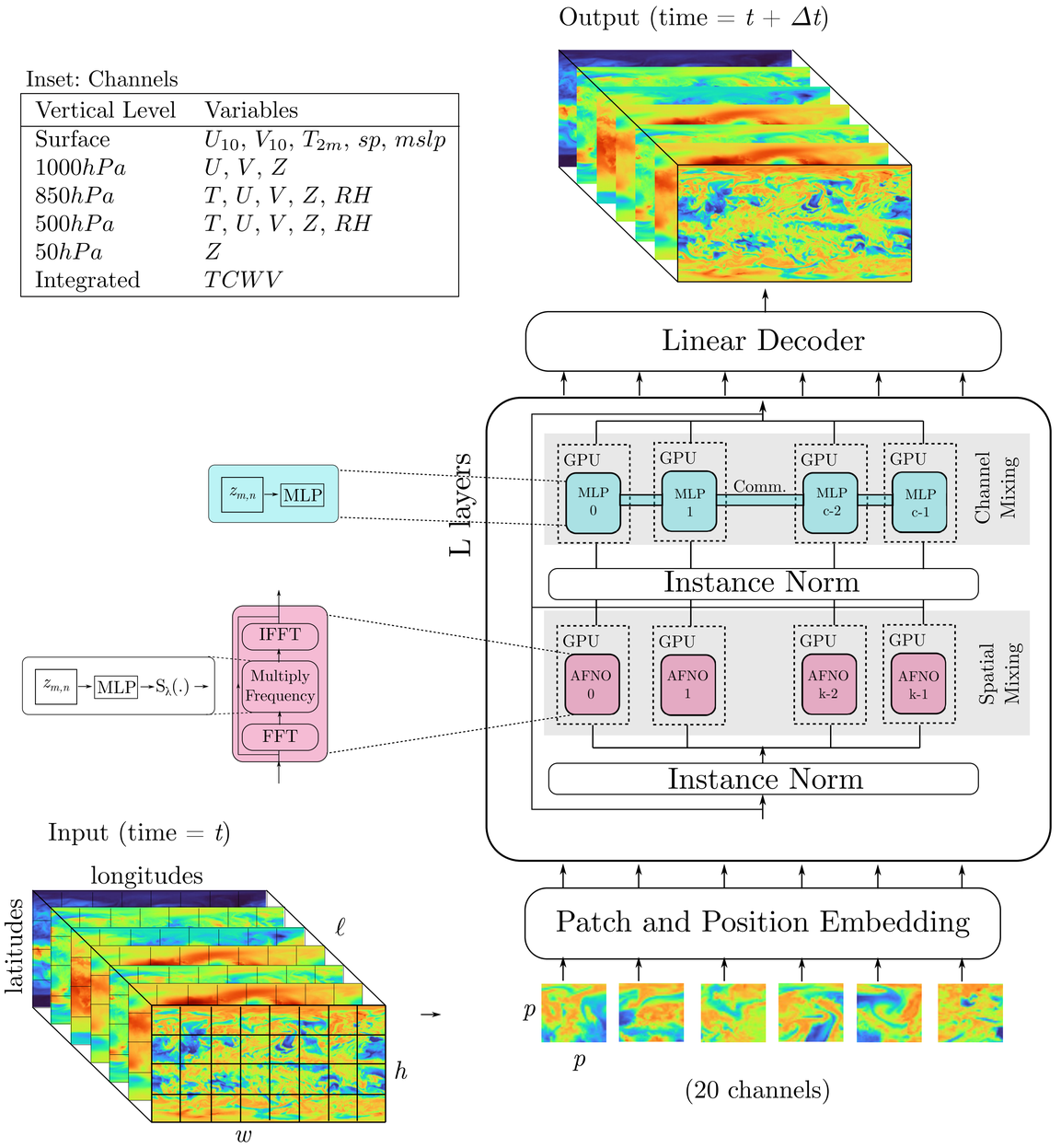
ground truth at time t + dt

- Model fully learned from data (could be augmented with physics inputs)
- Incorporating new features is straightforward
- Requires expertise in training DL models, training is expensive
- Inferred predictions are inexpensive
- Relies on high quality ground truth data
  (e.g. from simulations, reanalyzes, observations, etc.)

Adaptive Fourier Neural Operator

Bringing ViT and FNO together

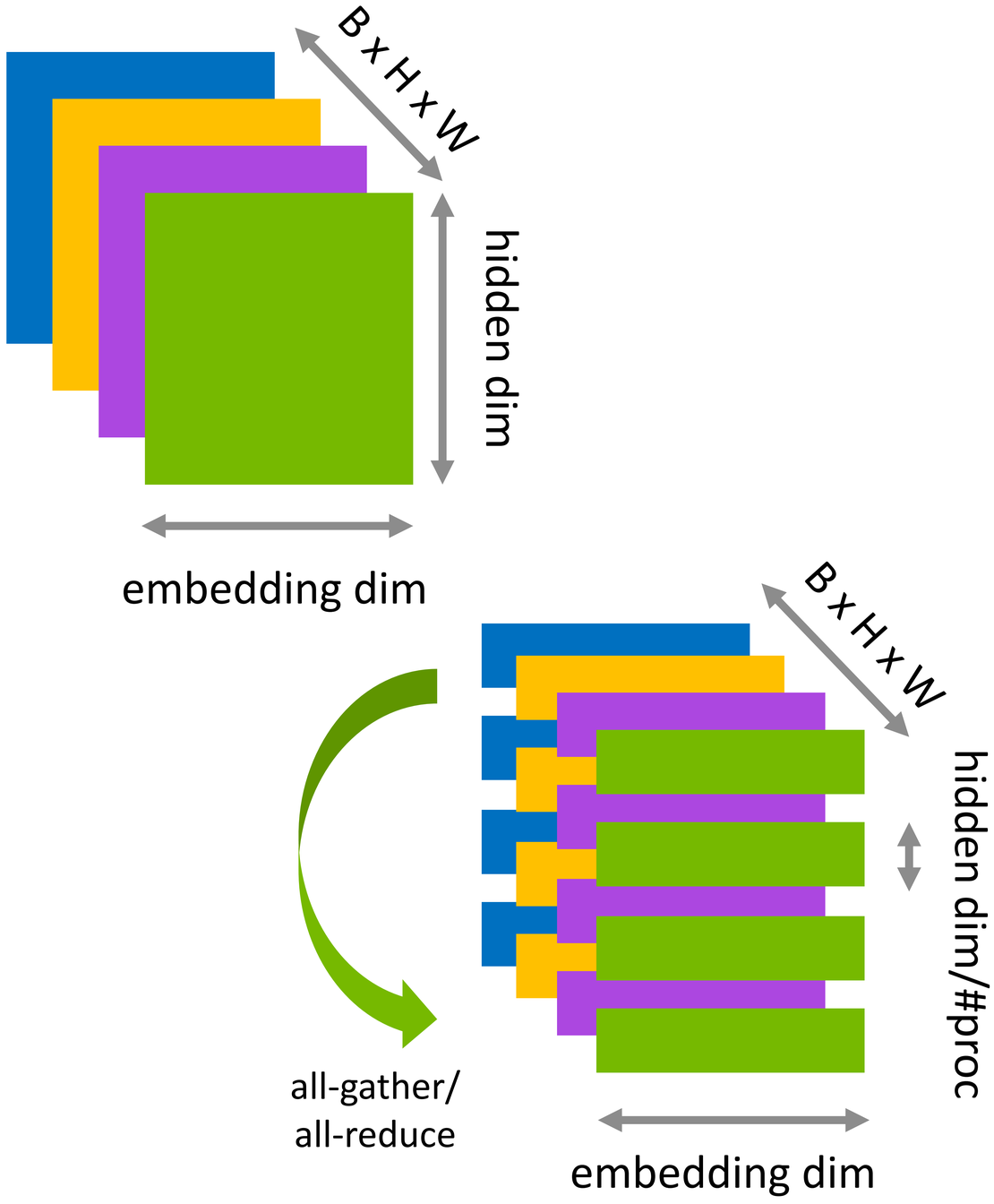
- Tokenization (patch embedding) of input data to reduce spatial complexity and memory footprint
- Learnable positional encoding
- Transformer Block (12x):
  - Inter-token mixing using 2D Fast Fourier Transforms (large scale features)
  - Intra-token mixing using fully connected layers (small scale features)
  - Normalization of spatial features
- Predictions performed on full resolution input grid
- Publication: arxiv.org/abs/2202.11214
- Available on ECMWF-lab: github.com/ecmwf-lab/ai-models-fourcastnet
- NVIDIA Modulus implementation available



Parallelizing AFNO FourCastNet

Hybrid Parallelism Required for Fast Training

- Load balancing friendly: work homogeneous across layers
- **Domain parallelism**: split up spatial domain, using distributed FFT, expensive communication (all-to-all)
- **Feature parallelism**: embedding dimension (feature dimension of patch embedding/tokenization) is usually O(1K): good target for feature parallelism (cf. MEGATRON), moderately expensive communication (all-reduce/all-gather)
- **Data parallelism**: straightforward, but leads to significant generalization gap at moderate batch sizes (~256), cheap communication (all-reduce)

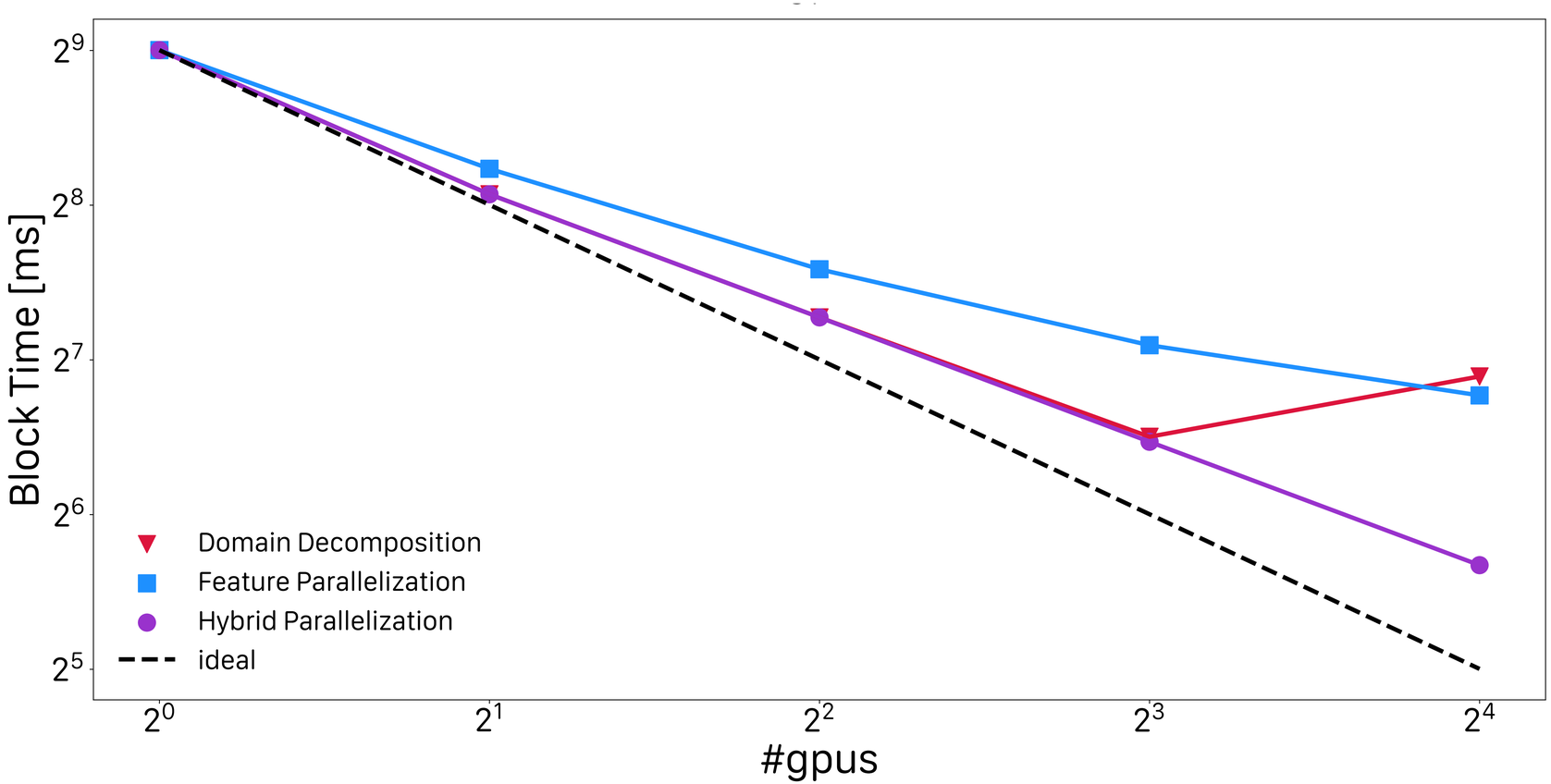


Implementation Overview in a Nutshell

PyTorch supports all necessary tools for parallelism

- Forward and Backward parallelism: torch.autograd.Function
- Gradient reductions modifications: DistributedDataParallel and
  - tensor backward hooks: all weights are shared among all ranks
  - comm hooks: full flexibility in implementing more complicated comm structures
- Data Loading: NVIDIA DALI with external source operator. cuPY arrays and pinned host memory for minimizing H2D transfer overhead, expensive CPU functions are numba compiled (recent addition)
- CUDA graphs to reduce jitter at scale, especially important with different comm groups in play and communication in the critical path
- Automatic Mixed Precision: FP16 and (recently) BF16
- Other important optimizations: optimizing tensor contractions in spectral space with torch.jit, ensuring generation of efficient CGEMM calls, using fused optimizers, using GPU-based LR schedulers, …

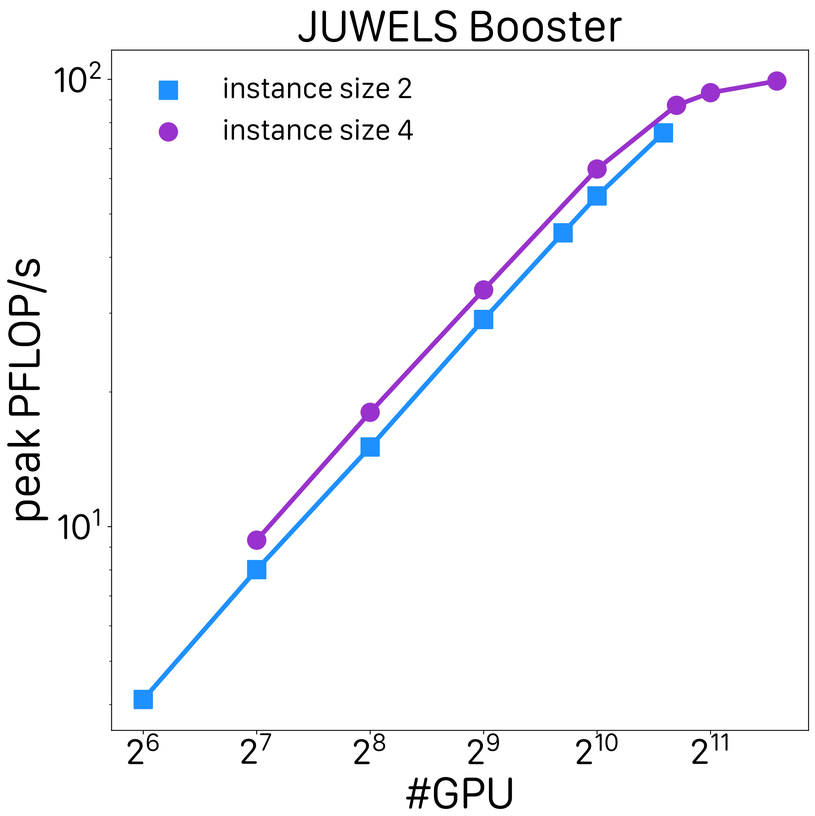
AFNO Scaling Example

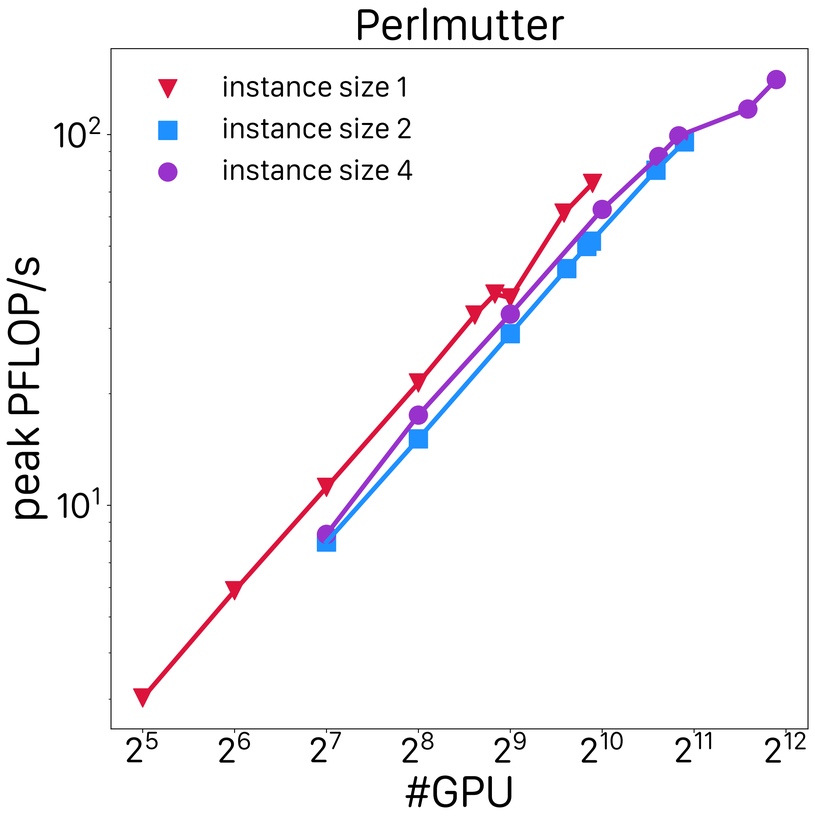


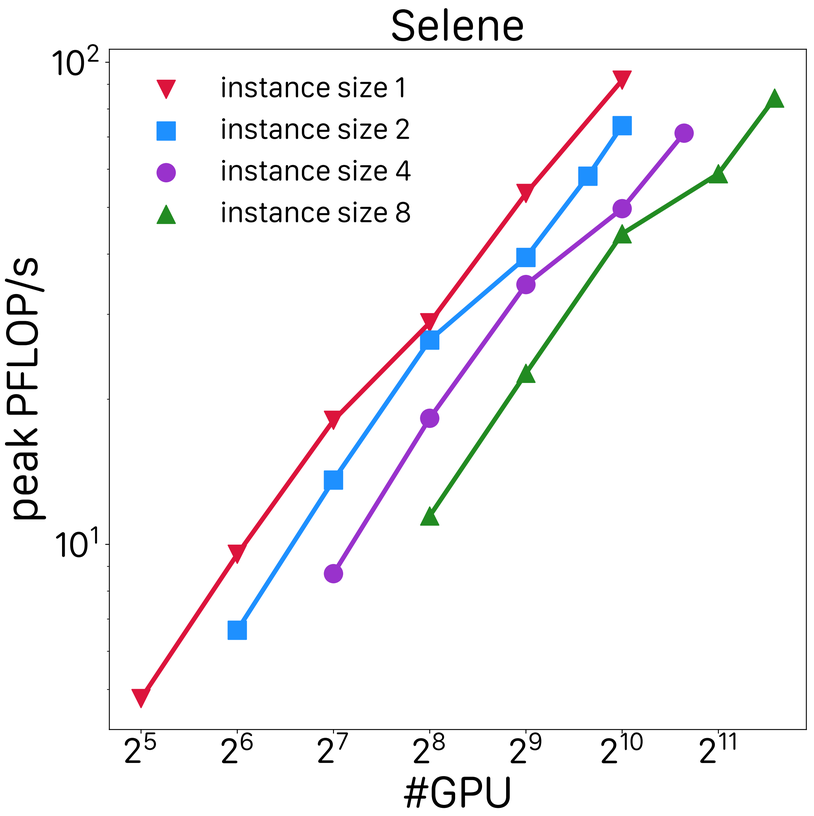
- Domain decomposition works well but only inside the NVLink island
- Hybrid parallelism is important for (strong) scalability
- When mixing with data parallelism, picture might change because of additional overhead of weight grad reductions: in practice, fewer parallelization dimensions work better, e.g.: feature + data

AFNO Performance Scalability

MEGATRON-like Fork-Join MLP Parallelization

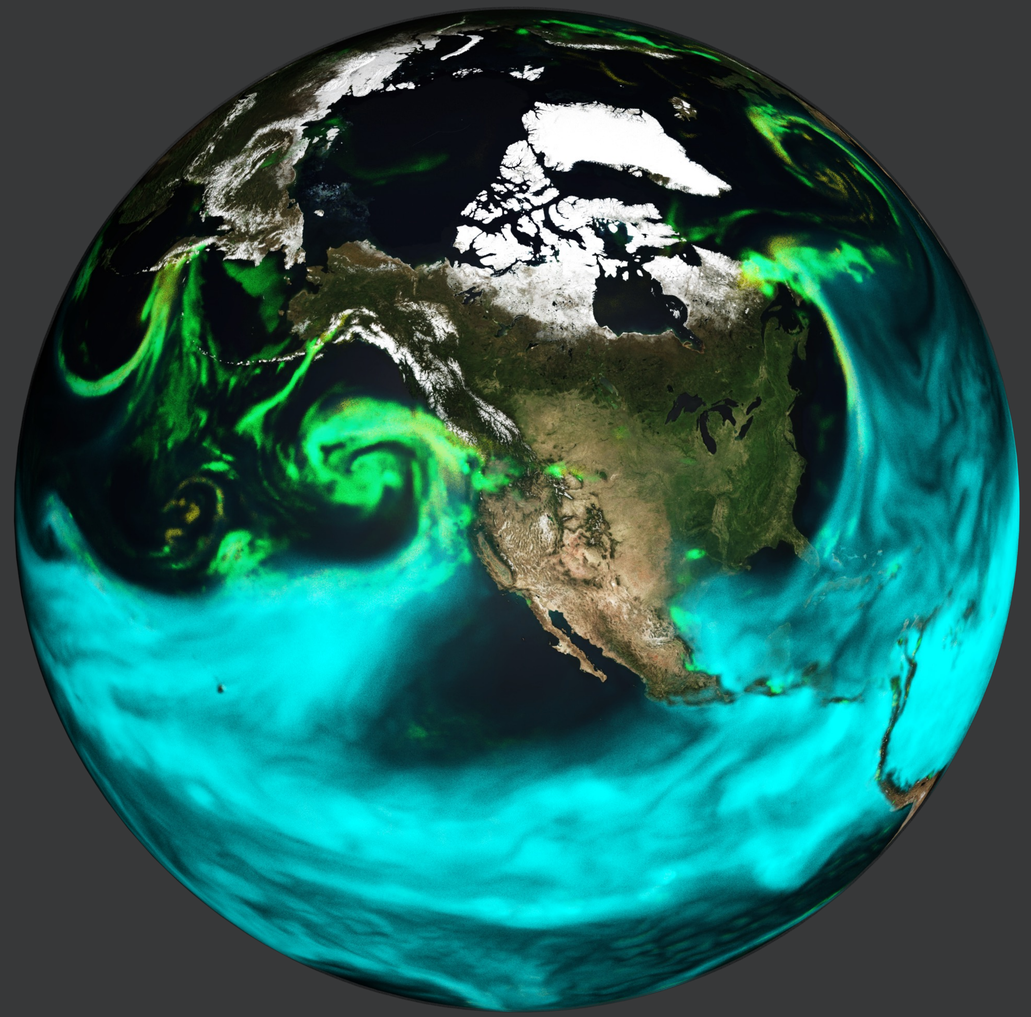






- Instance size: number of GPU running the same model instance (model parallel dimension size)
- Model parallel scaling close to perf modeling expectations

NVIDIA.



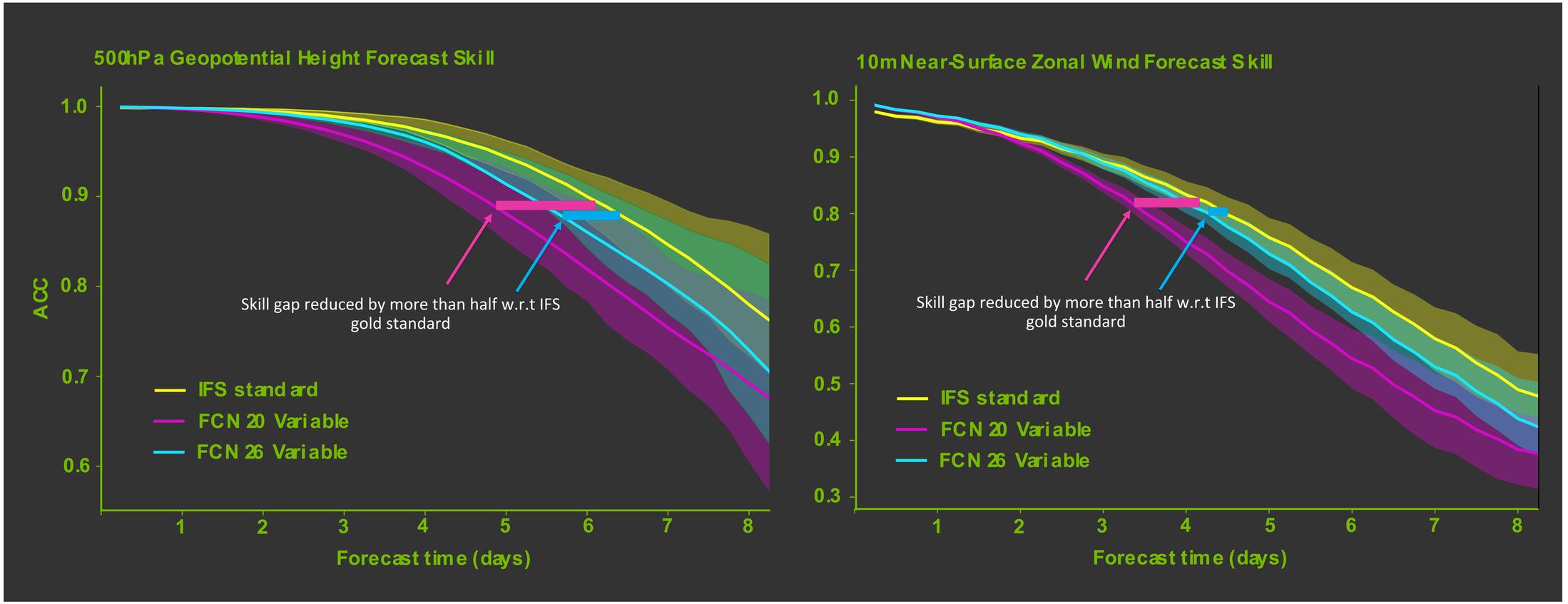
Fully Data-driven Weather Simulation with AFNO FourCastNet

• Scope	Global, Medium Range
• Model Type	Full-Model AI Surrogate
• Architecture	AFNO (Adaptive Fourier Neural Op.)
• Resolution:	25km
• Training Data:	ERA5 Reanalysis
• Initial Condition	GFS / UFS
• Training Time.	70 min @ 3072 A100
• Inference Time	70 sec @ 4 A100 (100-member, 10-days)
• **Speedup vs NWP**	**O(10^4-10^5)**
• **Power Savings vs NWP**	**O(10^4)**

NVIDIA.

Significant Skill Improvements in Short Amount of Time

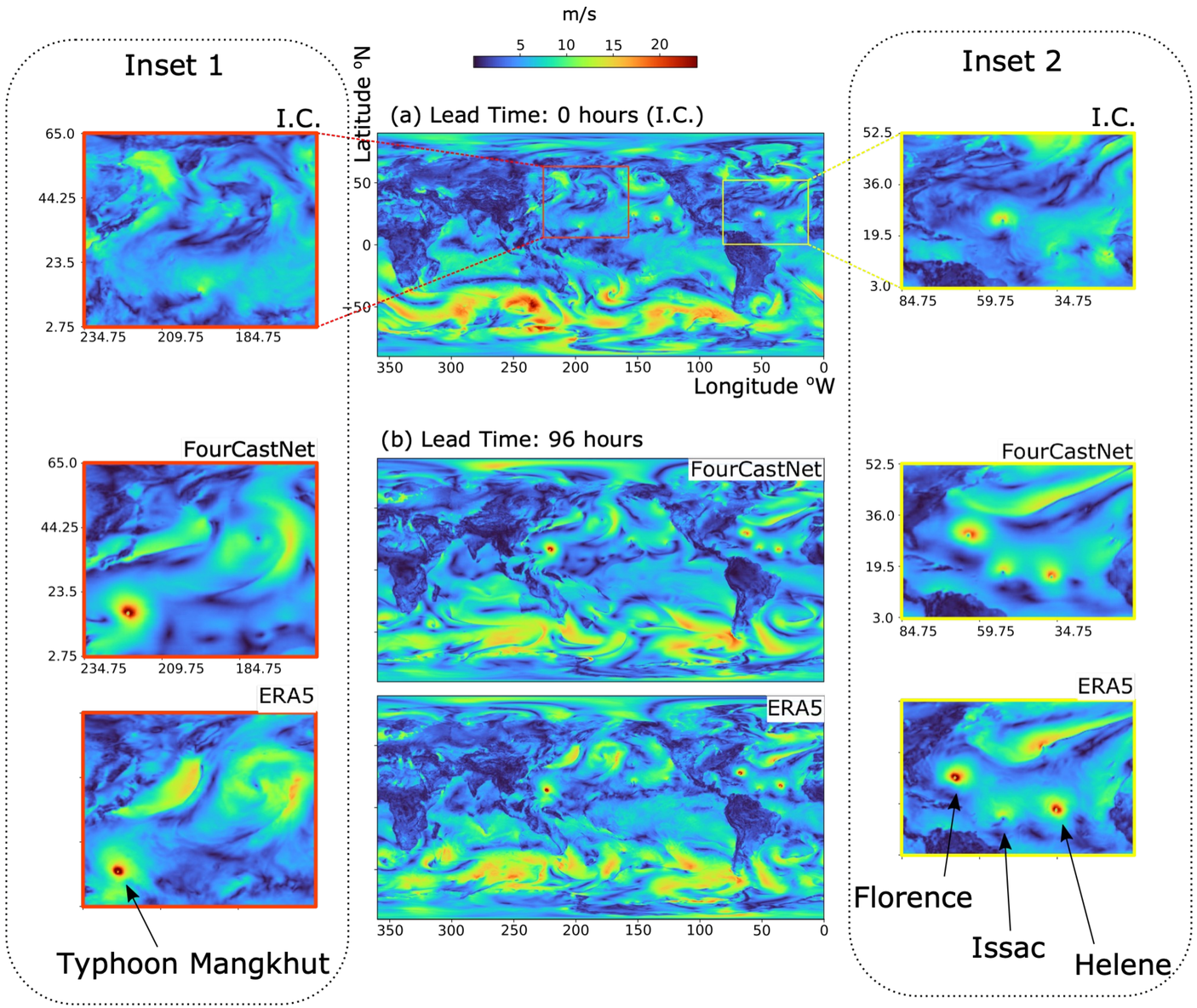
Despite small team of engineers



Acronyms:
ACC: Anomaly Correlation Coefficient (metric of weather skill)
IFS: The Integrated Forecast System
FCN: FourCastNet, our digital twin of weather

FCN has impressive skill on forecasting extremes

Including (extra-)tropical cyclones and atmospheric rivers



Switching Scopes: AI-driven Climate Modeling

Realistic climate simulation is a computational grand challenge

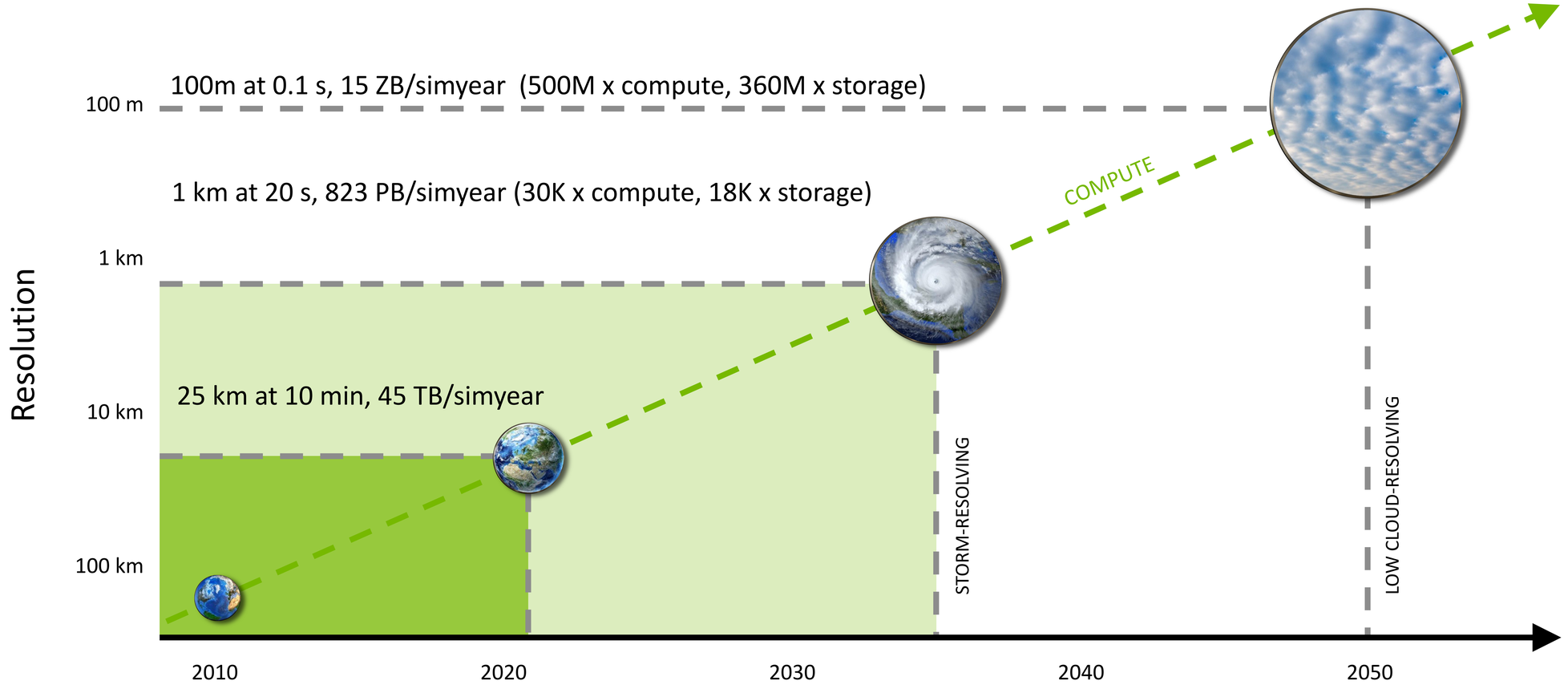
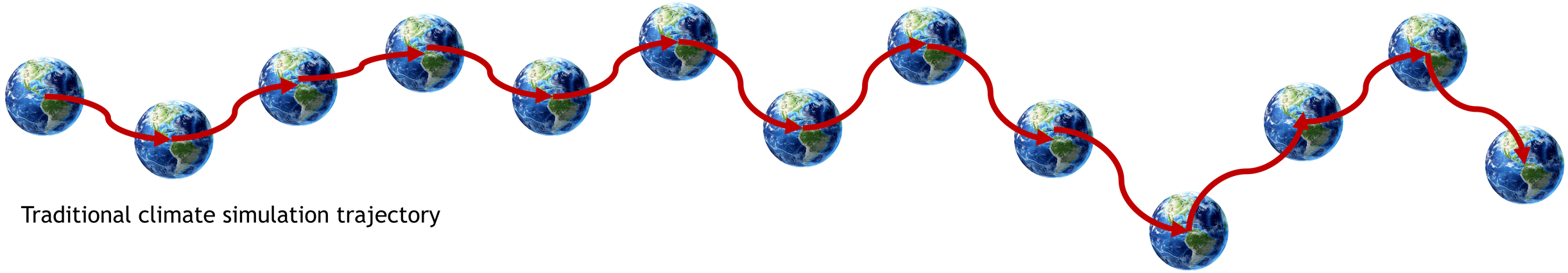


Figure adapted from: Schneider, T., Teixeira, J., Bretherton, C. *et al.* *"Climate goals and computing the future of clouds". Nature Climate Change* **7,** 3–5 (2017)

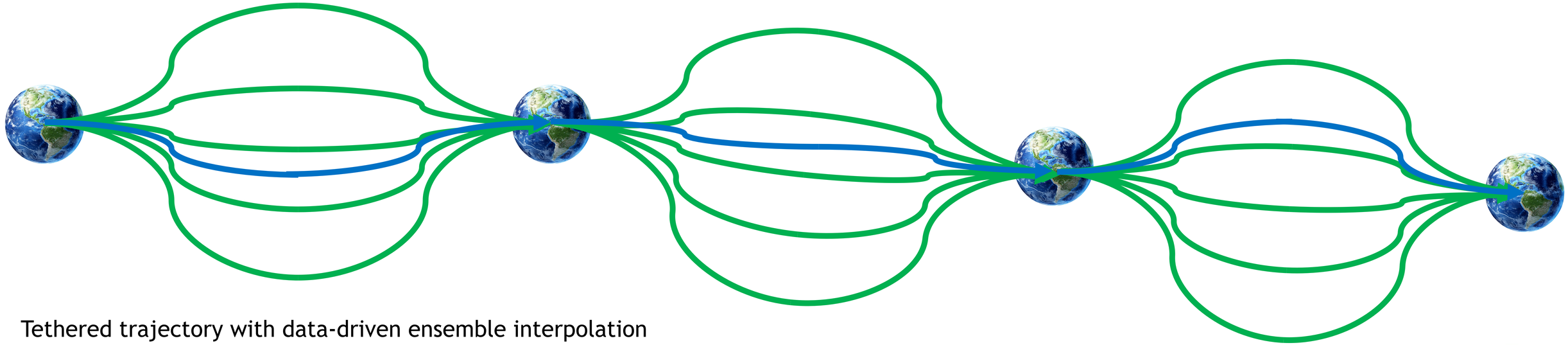
Tethering can solve storage, latency crisis facing high-res

AI nimbly generates details between "checkpoints" saved only infrequently from physics-based climate simulations

-- Bjorn Stevens, GTC 2021



Traditional climate simulation trajectory

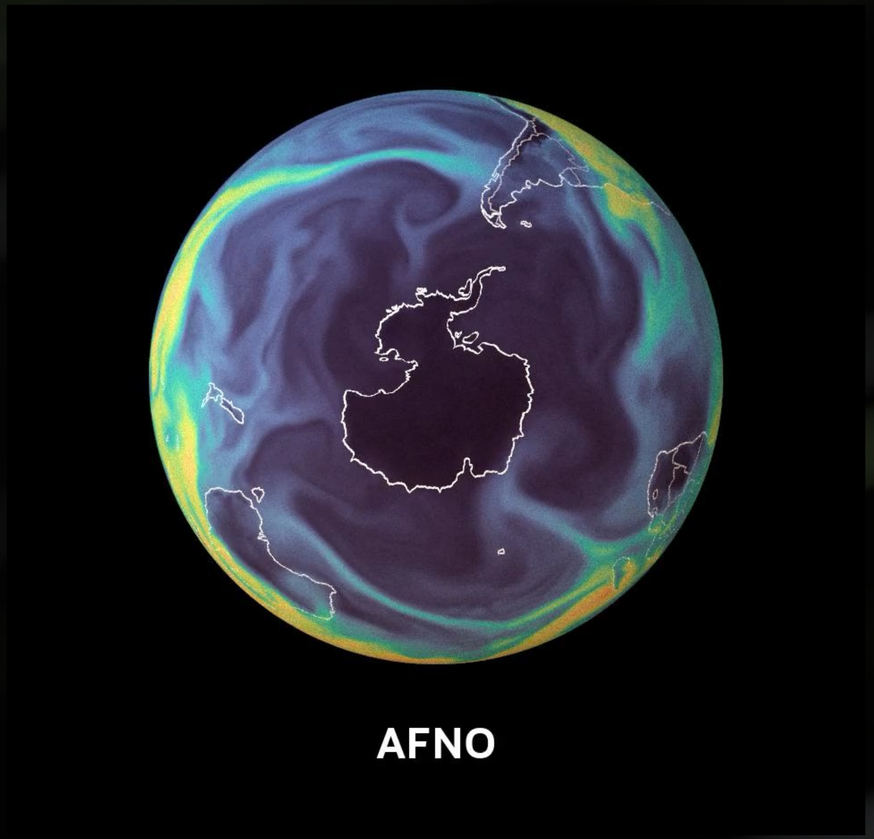


Tethered trajectory with data-driven ensemble interpolation

NVIDIA.

The Problem of Polar Instabilities

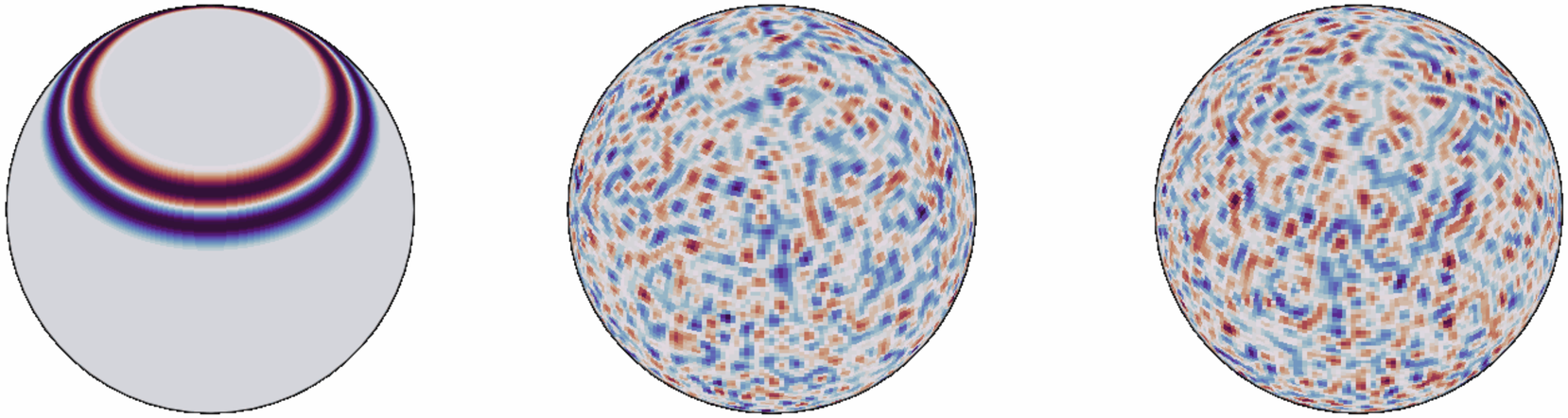
AFNO is treating spatial domain incorrectly



Correct topology is S2 and not S1xS1
(autoregressive feedback loop amplifies small errors over rollout steps)

harmonics

A PyTorch library

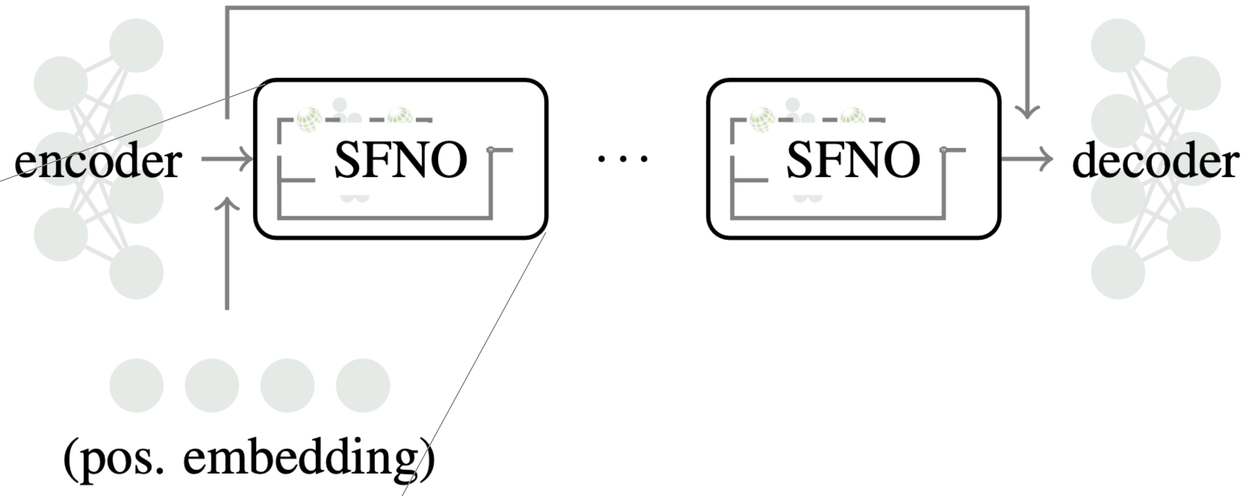


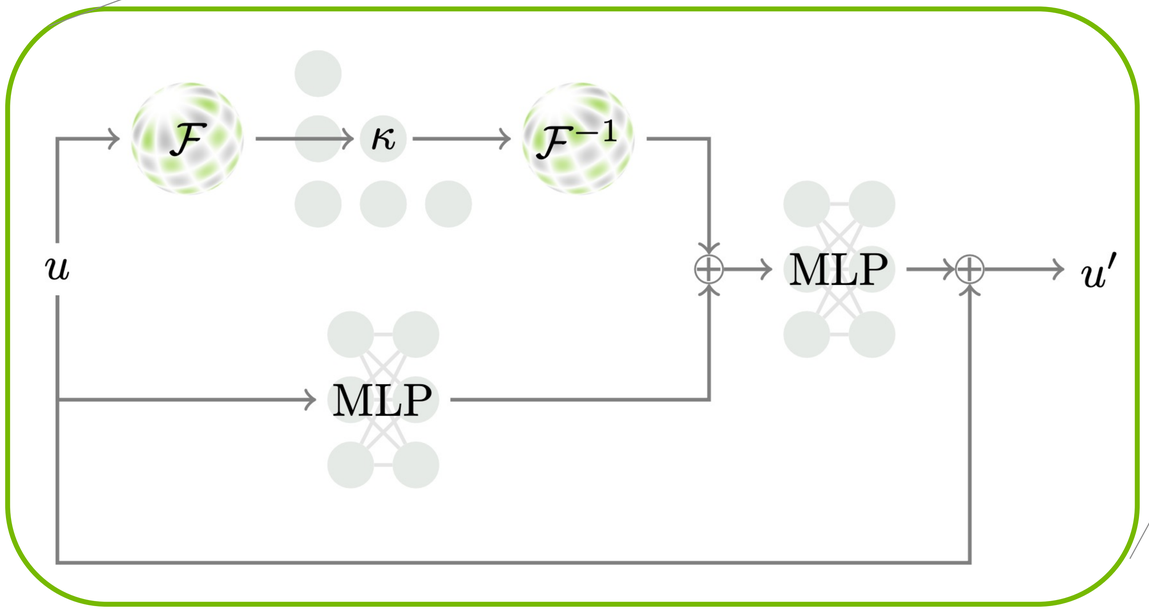
```python
import torch
import torch_harmonics as th

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

nlat = 512
nlon = 2*nlat
batch_size = 32
signal = torch.randn(batch_size, nlat, nlon)

# transform data on an equiangular grid
sht = th.RealSHT(nlat, nlon, grid="equiangular").to(device).float()

coeffs = sht(signal)
```

- Open-Source library under BSD-3 license: https://github.com/NVIDIA/torch-harmonics
- Efficient calls for forward and inverse spherical harmonic transformations
- Autograd support as differential layers in PyTorch
- Full support for distributed memory computation

SFNO Topology

Fully SO(3) Equivariant Architecture





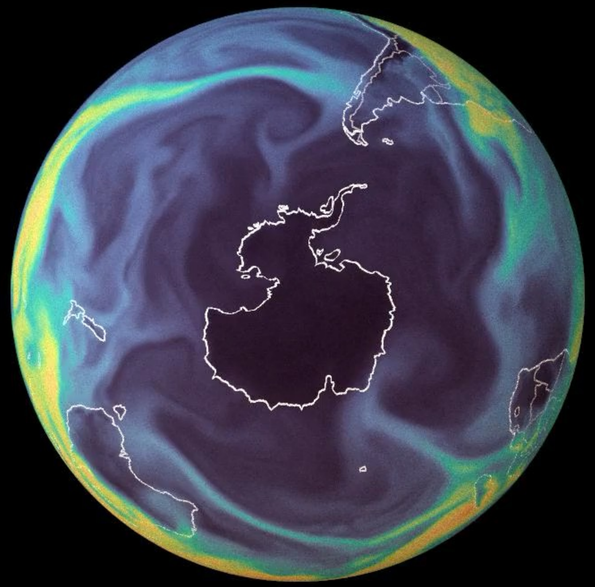
- Spherical Fourier layers derived from Convolution theorem on the Sphere
- Data preprocessing fully SO(3) equivariant
- Positional encoding SO(3) equivariant
- Instance norm instead of layer norm (elementwise affine operation in layer norm breaks SO(3) equivariance)
- Space-MLP (pointwise) and Fourier layers (Driscoll-Healy) are both SO(3) equivariant
- **Full architecture respects SO(3) symmetry**

NVIDIA.

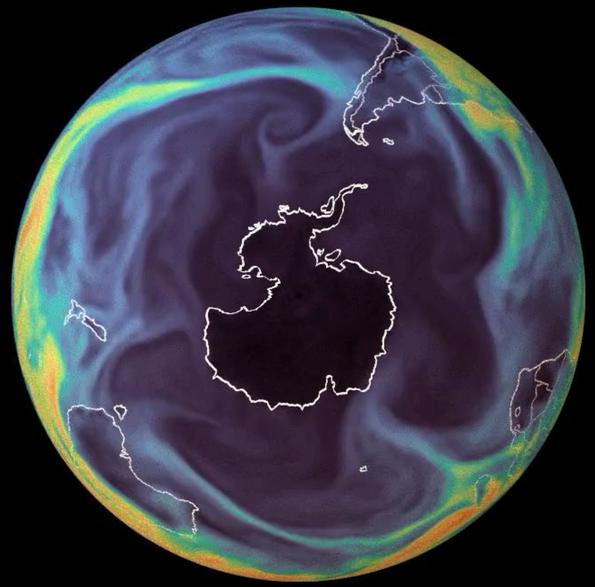
Polar instabilities

Equivariant treatment of spherical geometry overcomes instabilities

tcwv 2018-01-03 00:00:00



AFNO

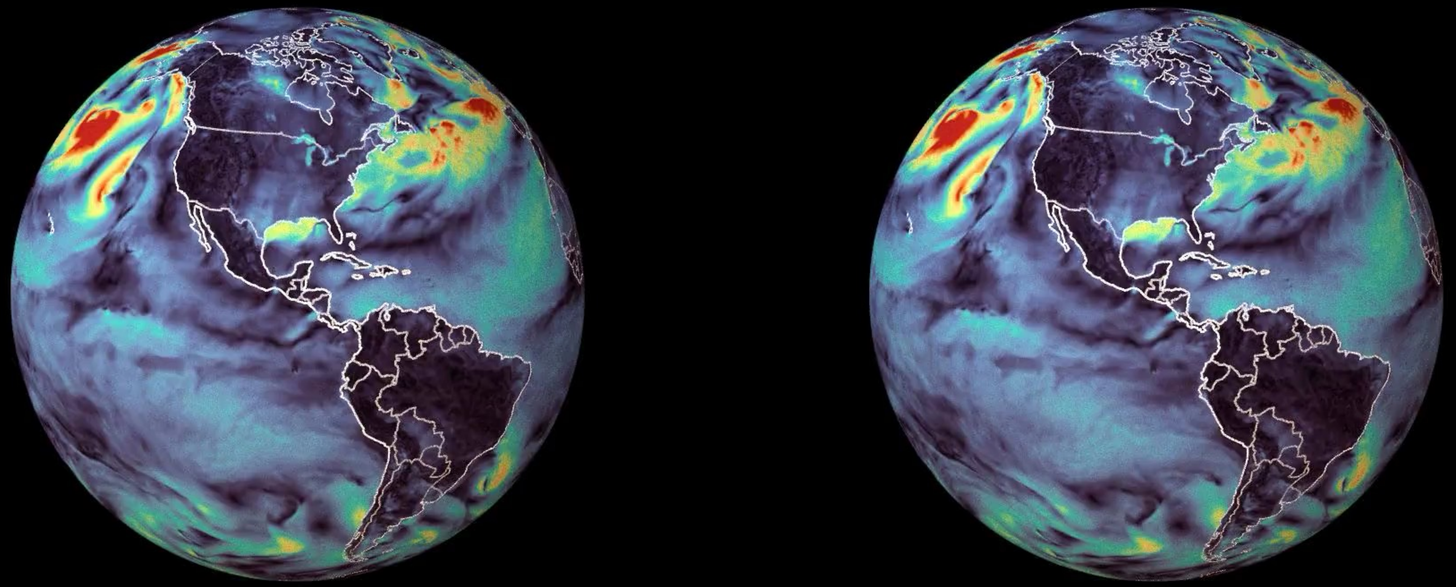


SFNO

Stable rollouts

SFNO rollouts remain stable over a year and generate visually plausible weather patterns

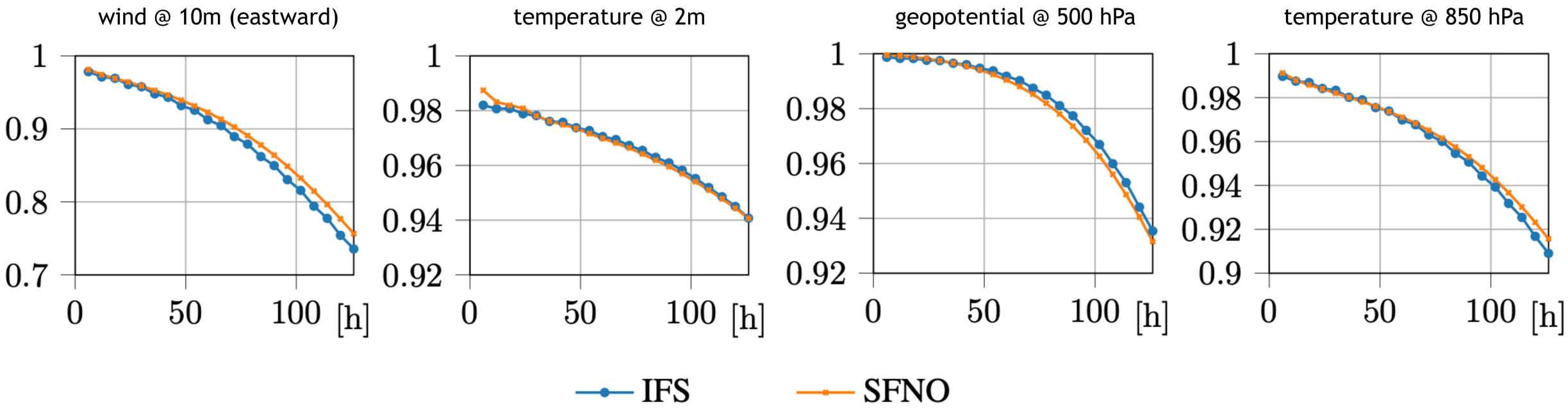
2018-01-01 00:00:00



* Stable one-year rollout (1460 autoregressive steps) computed in 13 minutes on a single NVIDIA RTX A6000

Skill compared to IFS

Preliminary Results



- FourCastNet-SFNO demonstrates excellent skill, comparable to IFS for relevant rollouts
- Development not finished, still many directions to explore
- Excellent skill with long rollout stability: candidate for S2S or climate science applications
- arXiv Link: Spherical Neural Operators: Learning Stable Dynamics on the Sphere, accepted for publication at ICML
- For more details:
  Anima Anandkumar (remote): Accelerating Earth System Emulation with Spherical Neural Operators,
  MS4C (Sertig), Tue, June 27, 16:00-16:30

Summary and Outlook

- Developed highly skilled, fast weather prediction model with stable long rollouts
  (including open source torch-harmonics, a library for differentiable spherical harmonics transforms in PyTorch)
- Next up: attacking subseasonal-to-seasonal predictions and climate science challenges
- After that: increasing spatio-temporal resolutions
- In process of making SFNO network architecture and trained weights available publicly via NVIDIA Modulus, stay tuned


NVIDIA
Thank You