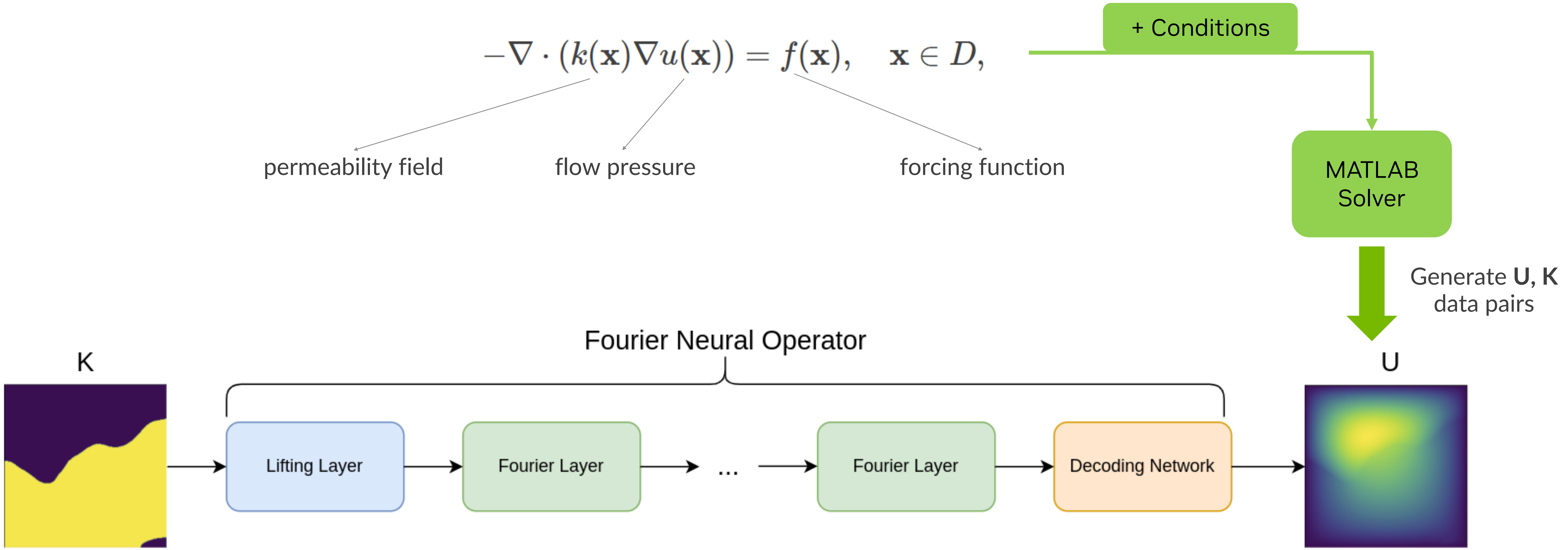
Data-driven approach

Darcy Flow Example

DATA: https://github.com/zongyi-li/fourier_neural_operator

$$-\nabla \cdot (k(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}), \quad \mathbf{x} \in D,$$



This problem develops a surrogate model that learns the mapping between a permeability field and the pressure field, K→U, for a distribution of permeability fields K~p(K). This is a key distinction of this problem from other examples, you are *not* learning just a single solution but rather a distribution.

A NOTE: Math behind Neural Operator

[Anima Anandkumar - Neural operator: A new paradigm for learning PDEs]

[Neural Operator: Learning Maps Between Function Spaces]

Nickola K., Zongyi L. et. al., Caltech.

[https://zongyi-li.github.io/neural-operator/]

Green Function

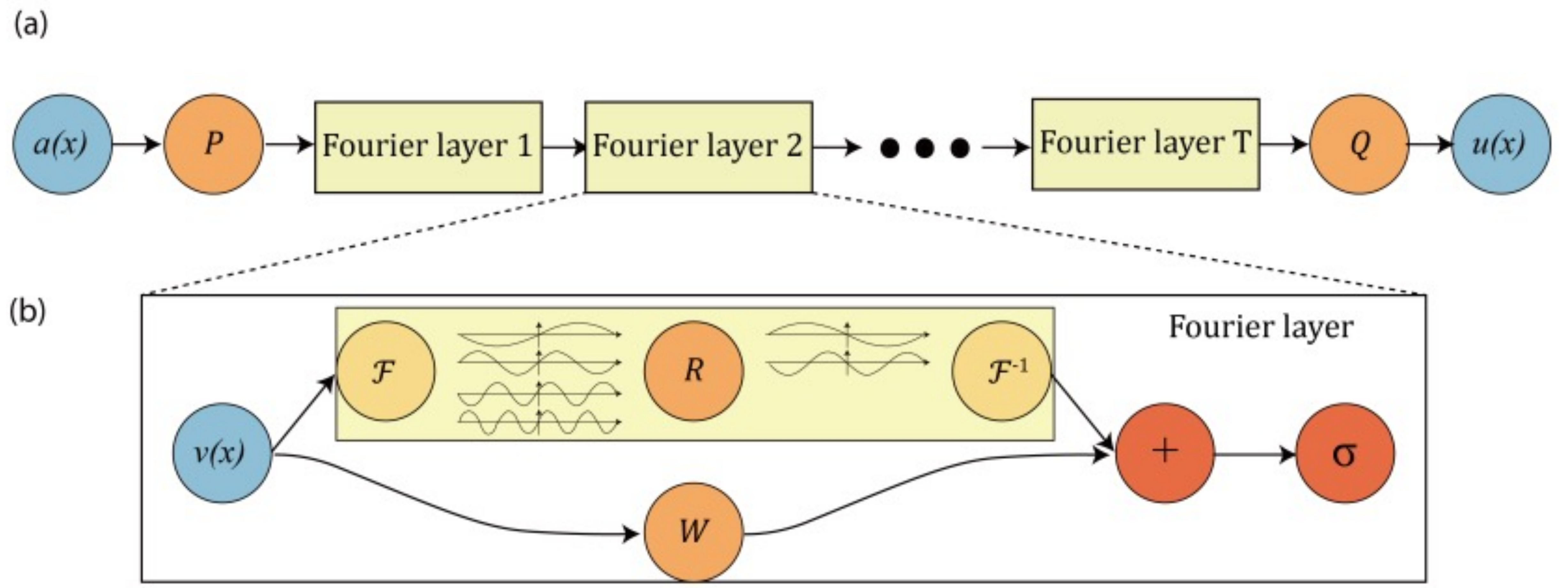
$$
\begin{aligned}
Goal: \quad & Lu = f \\
Def: \quad & L^{\dagger}G = \delta(x-\xi) \\
\langle Lu, G\rangle &= \langle f, G\rangle \\
\langle u, L^{\dagger}G\rangle &= \langle f, G\rangle \\
\langle u, \delta(x-\xi)\rangle &= \langle f, G\rangle \\
u(\xi) &= \langle f, G\rangle
\end{aligned}
$$

- Note
  - $\langle , \rangle$: function space inner product, project one function to another, which is the integral of multiplying two functions.
  - L: a linear operator.
  - $L^{\dagger}$: a "adjoint" operator of L, if $L = L^{\dagger}$ then they should has the same boundary conditions, named "self-adjoint operator".
  - δ: a `dirac function` use to `silter` function `u` at value ξ.
  - u: target solution. and ξ is just a dummy variable, actually we could simply change it here $u(\xi) \to u(x)$, we get the final solution with proper form.

Fourier Neural Operator (FNO)

https://arxiv.org/abs/2010.08895, ICLR 2021.
FNO: FFT + NO, Zongyi L. et. al., Caltech+NVIDIA

- Contribution
  - Learned an entire family of PDEs instead of solving only one instance such as FEM, FDM, PINN.
  - Resolution-invariant, can do Zero-Shot super-resolution.
  - Outperformed all existing DL methods with 30%+ lower error rate.
  - Sped up more than 440x comparing to Spectral Method.



- Method

Definition 1 (Iterative updates) *Define the update to the representation $v_t \mapsto v_{t+1}$ by*

$$v_{t+1}(x) := \sigma\Big(W v_t(x) + \big(\mathcal{K}(a;\phi)v_t\big)(x)\Big), \qquad \forall x \in D \tag{2}$$

Definition 2 (Kernel integral operator $\mathcal{K}$) *Define the kernel integral operator mapping in (2) by*

$$\big(\mathcal{K}(a;\phi)v_t\big)(x) := \int_D \kappa\big(x,y,a(x),a(y);\phi\big)v_t(y)\mathrm{d}y, \qquad \forall x \in D \tag{3}$$

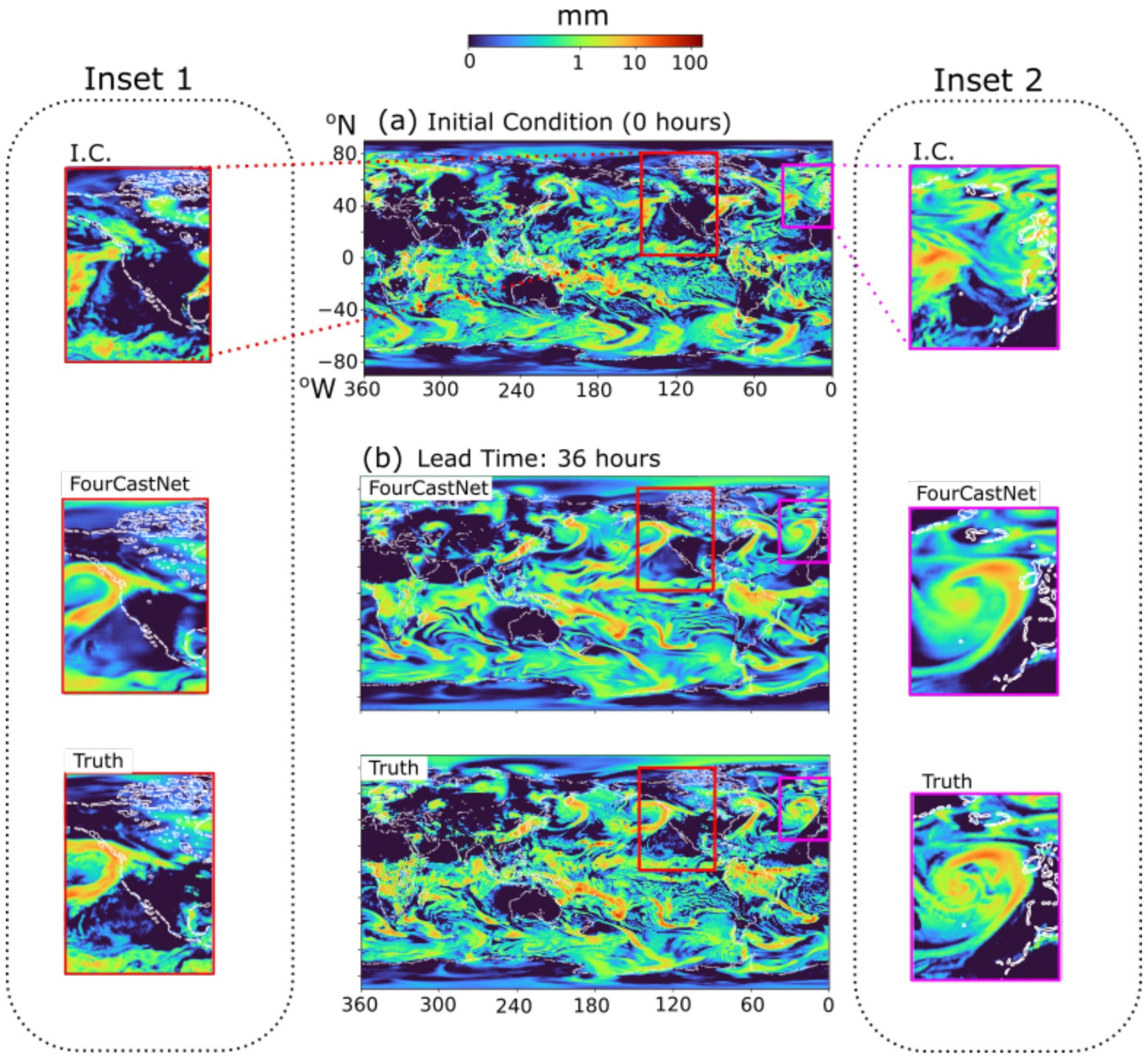
Definition 3 (Fourier integral operator $\mathcal{K}$) *Define the Fourier integral operator*

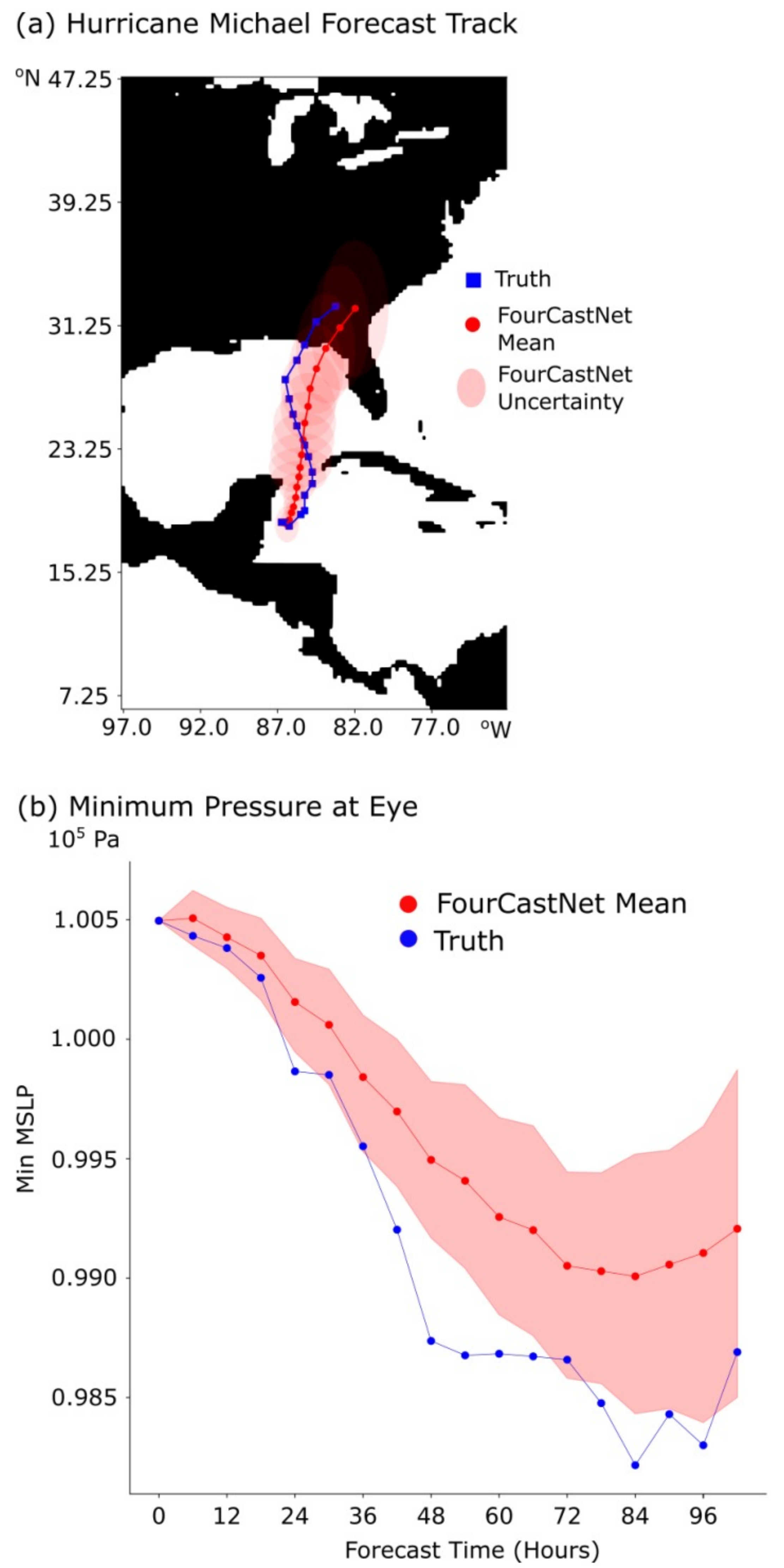
$$\big(\mathcal{K}(\phi)v_t\big)(x) = \mathcal{F}^{-1}\Big(R_\phi \cdot (\mathcal{F}v_t)\Big)(x) \qquad \forall x \in D \tag{4}$$

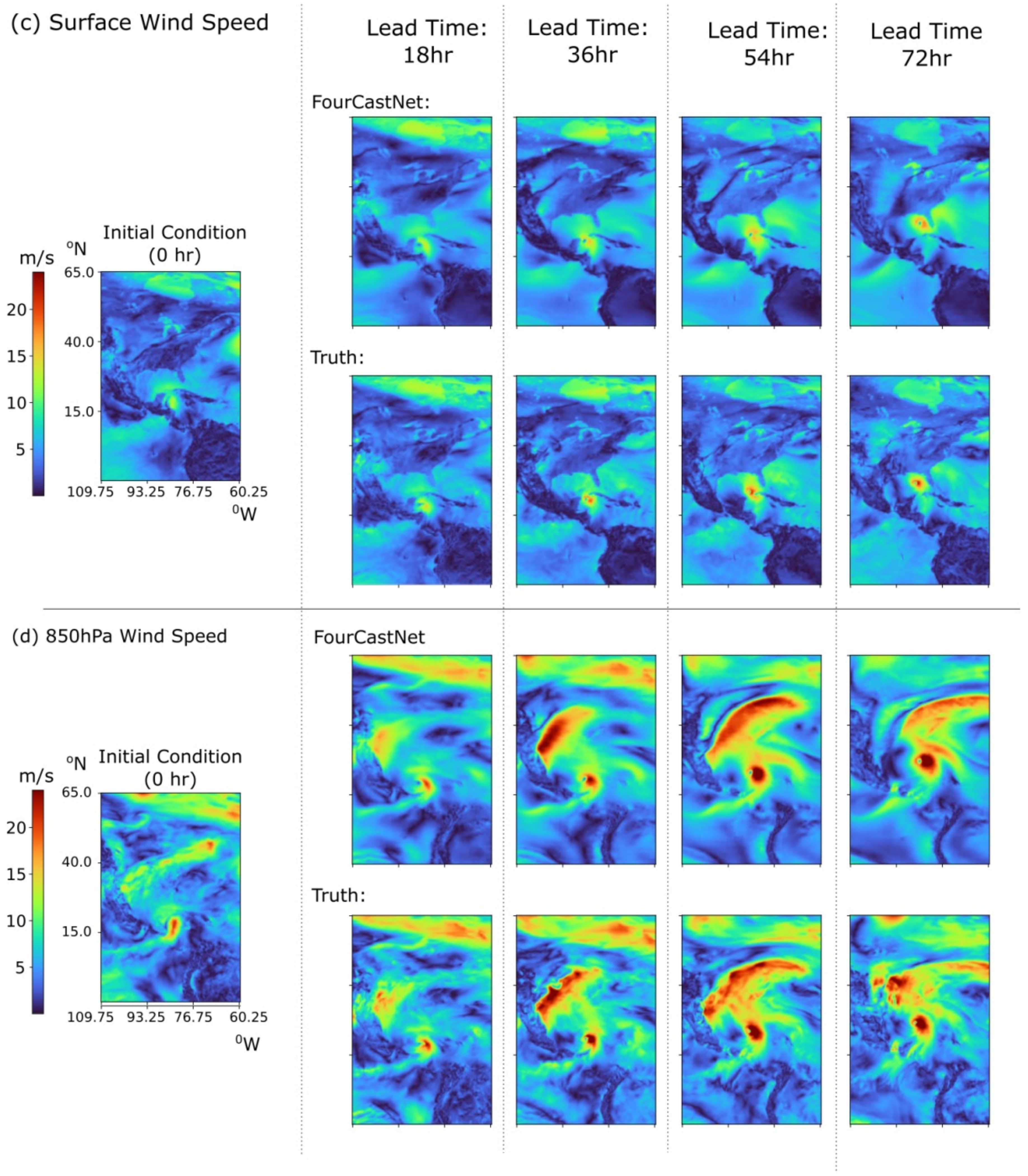
FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators

https://arxiv.org/abs/2202.11214

NVIDIA, Lawrence Berkley National Lab, University of Michigan, Rice University, Caltech, Purdue University







FOURCASTNET: A GLOBAL DATA-DRIVEN HIGH-RESOLUTION WEATHER MODEL USING ADAPTIVE FOURIER NEURAL OPERATORS

- Contribution
  - Super Accuracy
    - Outperforming IFS for small-scale precipitation prediction within 48hr.
    - FourCastNet resolves extreme events such as tropical cyclones and atmospheric rivers that prior DL models are failed owing to their coarser grids
  - Super Fast
    - FourCastNet is about 45000 times faster than traditional NWP models on a node-hour basis.
    - Enabling the generation of very large ensembles (1000+)
    - FourCastNet generates a week-long forecast in less than 2 seconds, enabling rapid testing.
  - Super Efficient and Scalable
    - Once trained FourCastNet uses about 12,000 times less energy to generate a forecast than the IFS model
    - FourCastNet has 8 times greater resolution than the SOTA DL model.

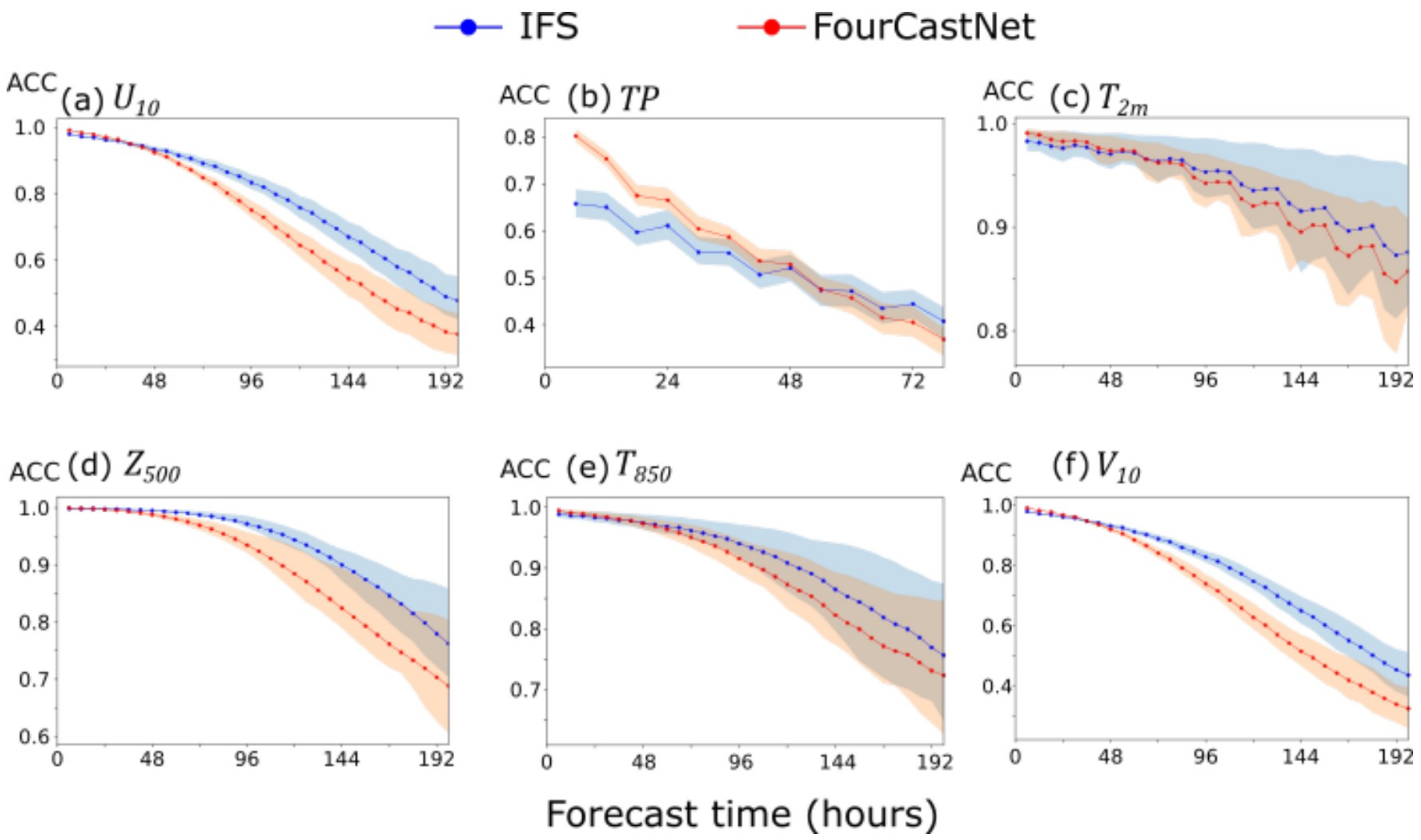
- Method
  - Training Time: 16 hours on a 64 NVIDIA A100 GPUs cluster.
  - Training Data: ERA5 Reanalysis dataset (link)
    - FourCastNet selects 20 variables at 5 vertical levels every 6 hors.
    - Training 54020 samples; Validation 2920 samples; Testing ~1k samples;
    - Resolution: 30km x 30km (720 x 1440 pixels x 20 variables), 8x8 patches
  - Comparison with NWP Model: ECMWF IFS
    - has 150+ variables at 50+ vertical levels, 50 emsembles.
  - Model:
    - Adaptive Fourier Neural Operator (AFNO) model with Fourier transform-based token-mixing scheme [Guibas et al., 2022]
    - Vision transformer (ViT) backbone [Dosovitskiy et al., 2021].
      - convolutional architectures showed poor performance on capturing small scales over many time steps in auto-regressive inference
  - Note that Earth-2
    - Use to help NWP, and will not possible to replace NWP. (at least for now)
    - has no Physics-Informed yet. (possible in future work)
    - Build on top Pytorch, not Modulus and Omniverse yet. (possible in future work)

Vertical Level	Variables
Surface	$U_{10}, V_{10}, T_{2m}, sp, mslp$
$1000hPa$	U, V, Z
$850hPa$	T, U, V, Z, RH
$500hPa$	T, U, V, Z, RH
$50hPa$	Z
Integrated	$TCWV$

FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators

https://arxiv.org/abs/2202.11214

NVIDIA, Lawrence Berkley National Lab, University of Michigan, Rice University, Caltech, Purdue University



Latency and Energy consumption for a 24-hour 100-member ensemble forecast				
	IFS	FCN - 30km (actual)	FCN - 18km (extrapolated)	IFS / FCN(18km) Ratio
Nodes required	3060	1	2	**1530**
Latency (Node-seconds)	984000	7	22	**44727**
Energy Consumed (kJ)	271000	7	22	**12318**

NVIDIA

WRAP UP

How does Modulus compliment PyTorch?

Features that can aid data and/or physics driven problems

Data & Physics oriented utils

- Performance Enhancements
CUDA graphs, kernel fusion, JIT compilation, data parallelism, model parallel, etc.
- Pre-built Network Architectures
Diffusion Models, Neural Hash Encoding, Neural Operators, Graph Networks, DCT-RNN, several variants of MLPs etc.
- Hydra Configuration
Hyper-parameter tuning and customization
- Data Pipeline
For very large data-driven problems using Zarr, NVComp, GDS
- Data and Inference Tools
Pre-defined datasets for common data formats (VTK, HDF5, …). Model export functions to TensorRT and Triton
- Integration with Other Products
Omniverse, PySDF, NVFuser, Triton, Tensor RT, DALI, Warp, etc.

Physics oriented utils

- Geometry Module
Integrated, parameterized geometry module with point cloud/SDF
- Symbolic PDE Loss Construction
Automated PDE loss construction using SymPy API
- Automated Optimized Gradient Calculations
Automatic gradient calculations for physics-informed learning with optimizations such as FuncTorch, AMP16, mesh free derivatives etc.
- Convergence and Stabilization Methods
Mass balance control planes, loss balancing schemes, AdaHessian support, learning rate annealing, etc.
- Exact Boundary Enforcement
Exact enforcement of continuity or boundary conditions
- Variational Learning
Solving PDE systems using variational formulations

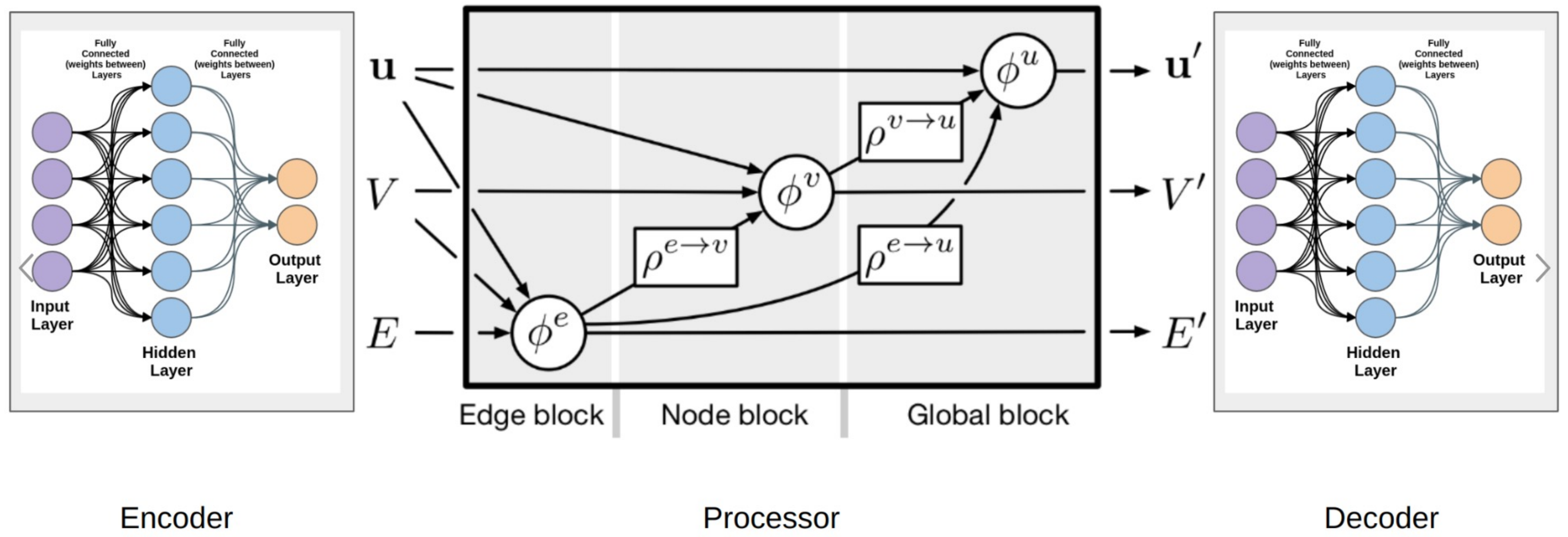
Summary: Modulus Framework

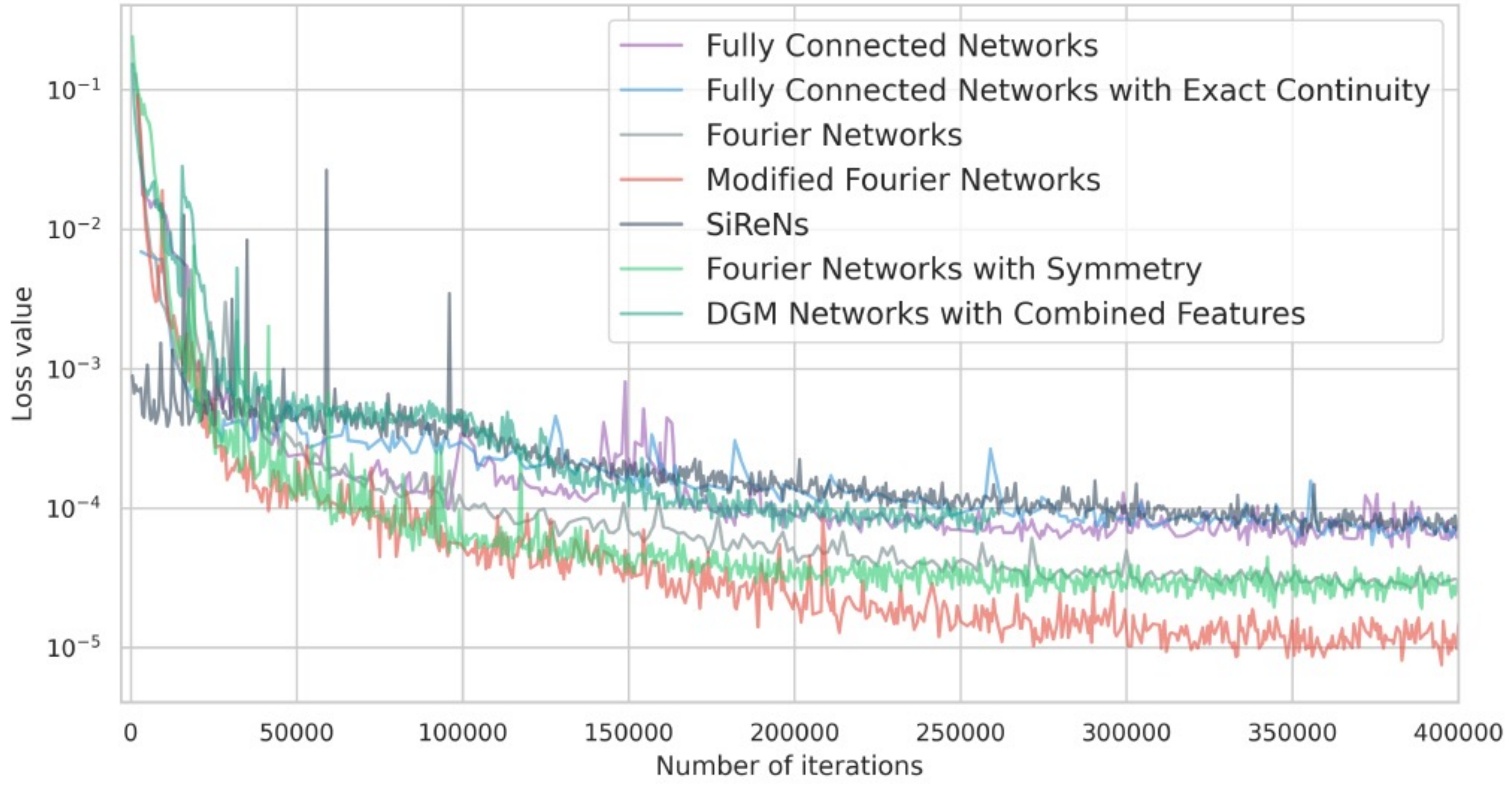
- Modulus is a framework to build and customize Physics-ML models and training workflows:
  - Built on top of PyTorch, interoperable with PyTorch
  - Utilities designed to enable DL researchers as well as Domain experts
  - A vast library of optimized models particularly suited for Physics-ML applications
  - Develop performant and scalable training pipelines with ease
  - Utilities to introduce Physics-based guard-railing
  - Several case studies across different domains to enable transition from POC to Industrial problems
  - Open Source and accessible via GitHub, NGC registry, PyPi

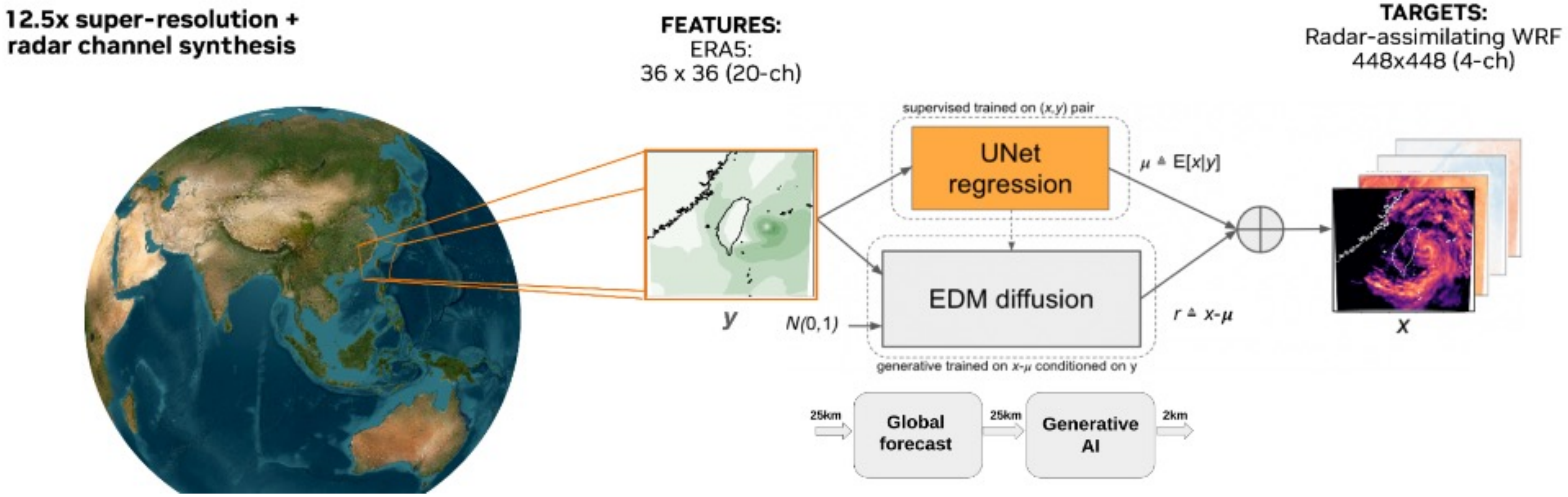
Summary: Modulus Framework

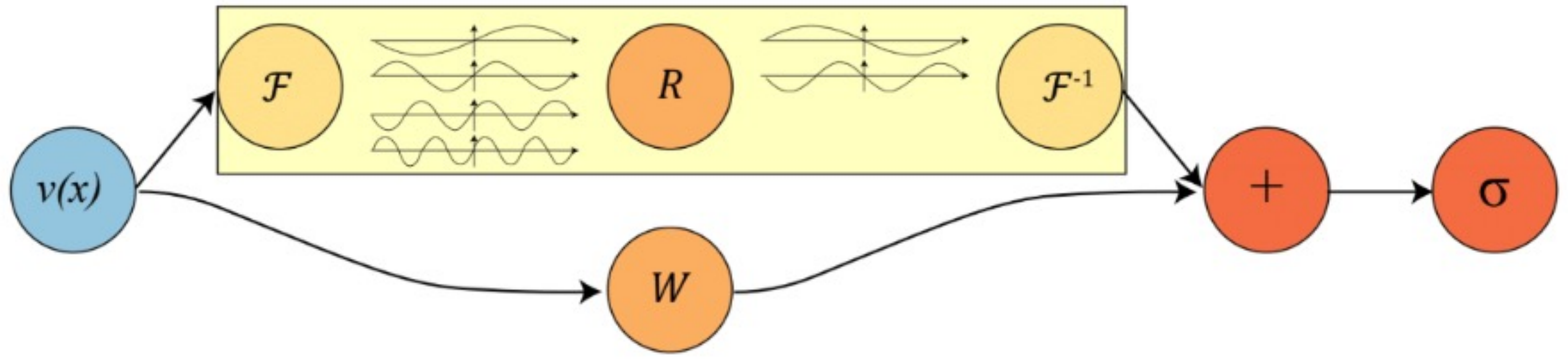
Variety of model architectures

- Modulus Model Zoo - Diverse Physics-ML approaches:
  - Fully Physics driven AI models
  - Fully data driven AI models
  - Hybrid (data + Physics) AI models
- Models like GNNs, RNNs, Diffusion Models, Neural Operators (FNO, DeepONet, etc.), MLPs and its variants available to use out-of-the-box or use optimized layers and components in your custom models
- Modulus Model enable features like:
  - Argument-less checkpoint loading
  - Model registry
  - …





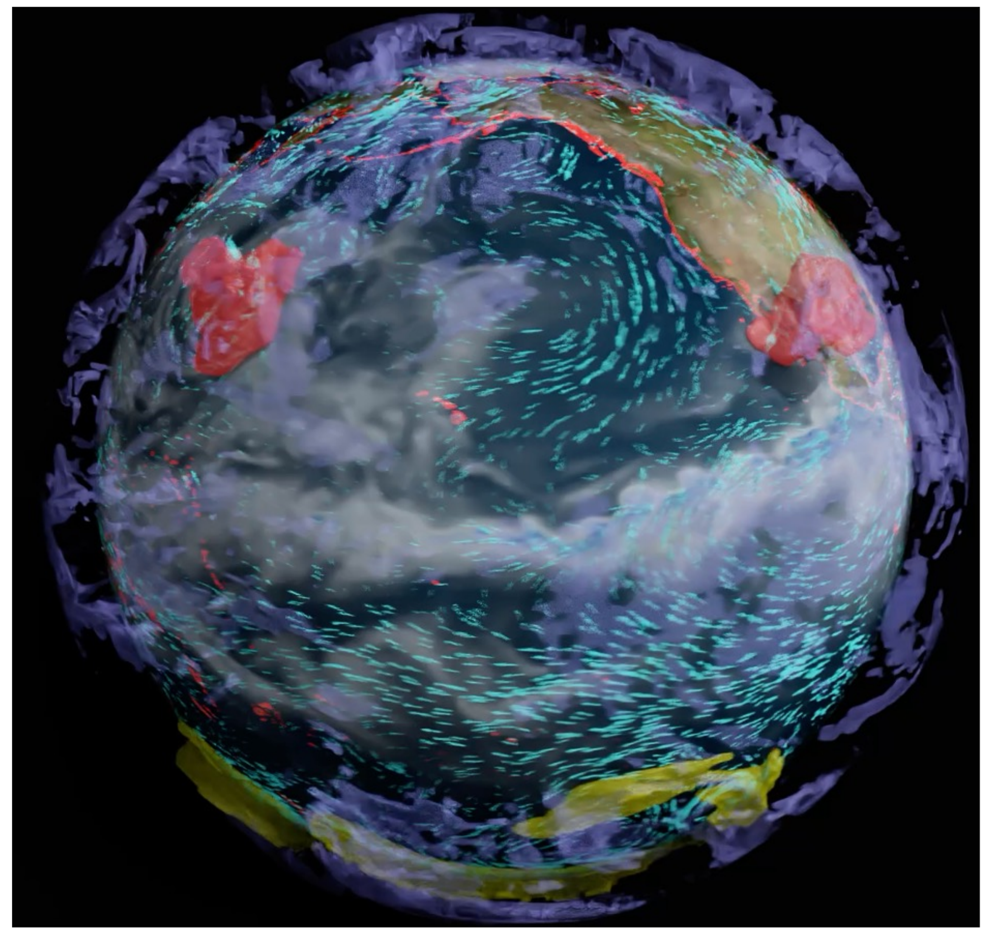




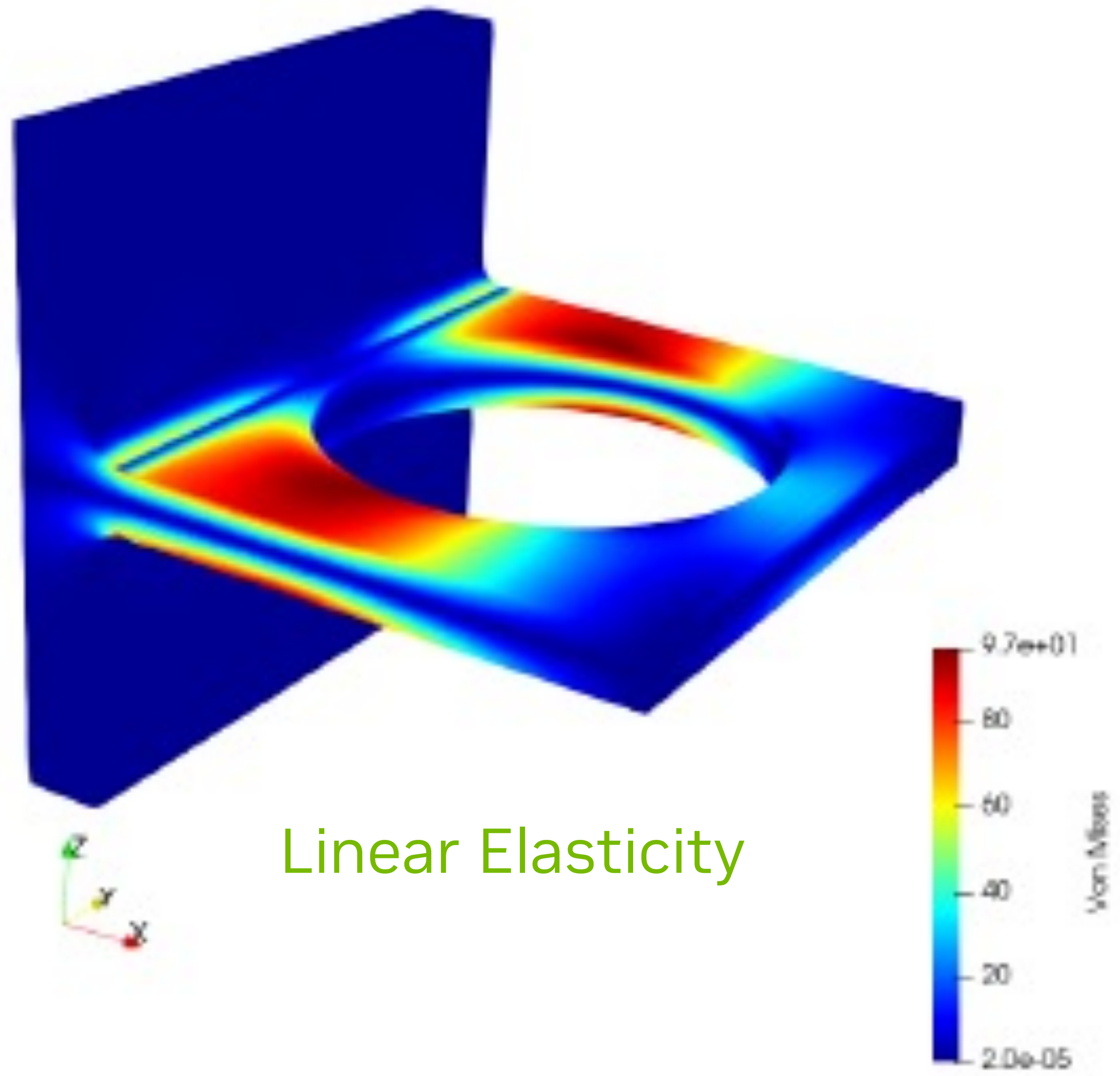
Summary: Modulus Framework

Variety of physics applications

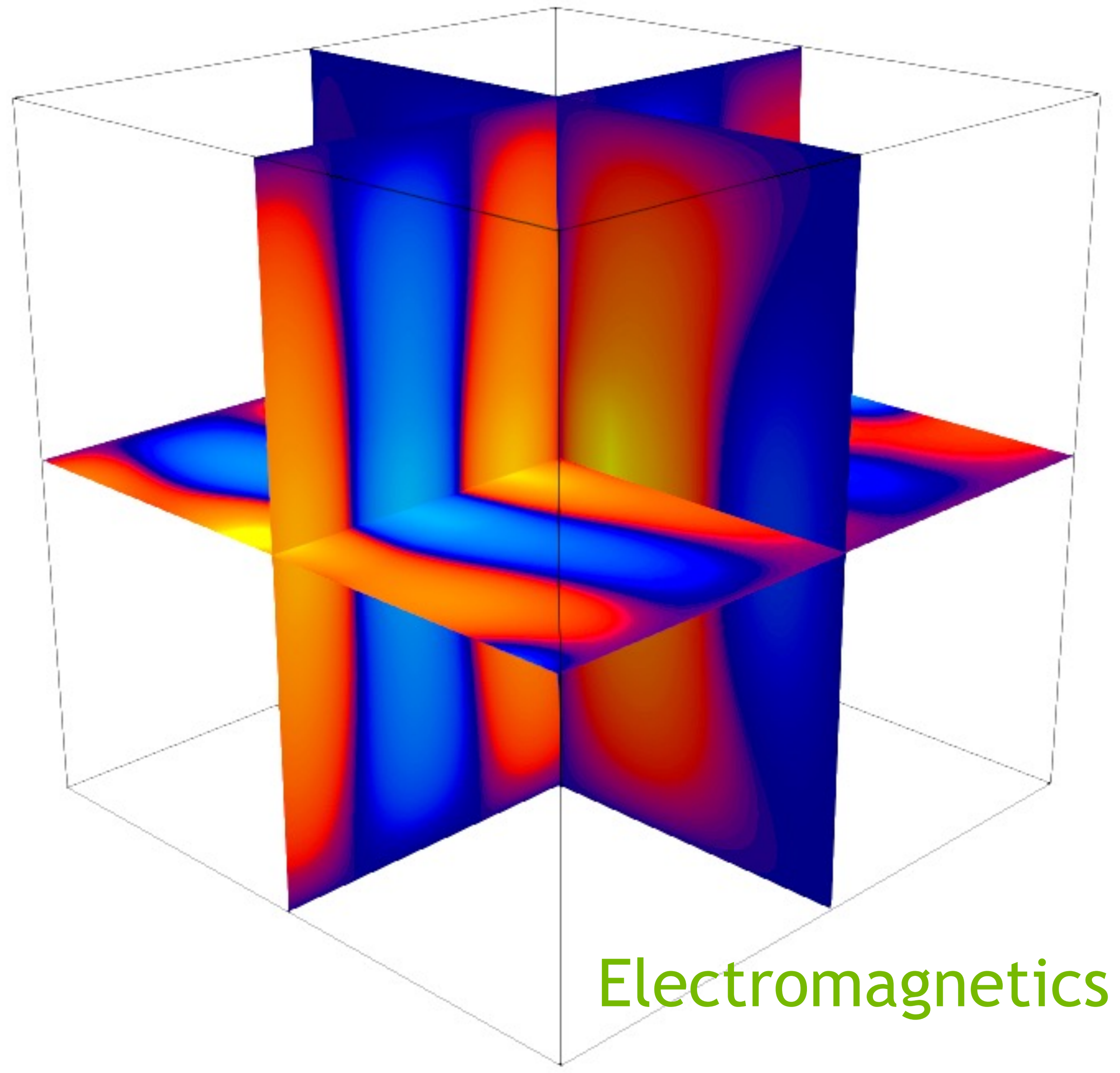
- Sample case studies and use cases covering diverse physical domains like:
  - Fluid dynamics
  - Heat transfer
  - Acoustics and waveform inversion
  - Linear Elasticity
  - Molecular Dynamics
  - Electromagnetics
  - Turbulence modeling
  - Weather and climate modeling
  - And more….
- Reference samples for Modulus, and Modulus-Sym



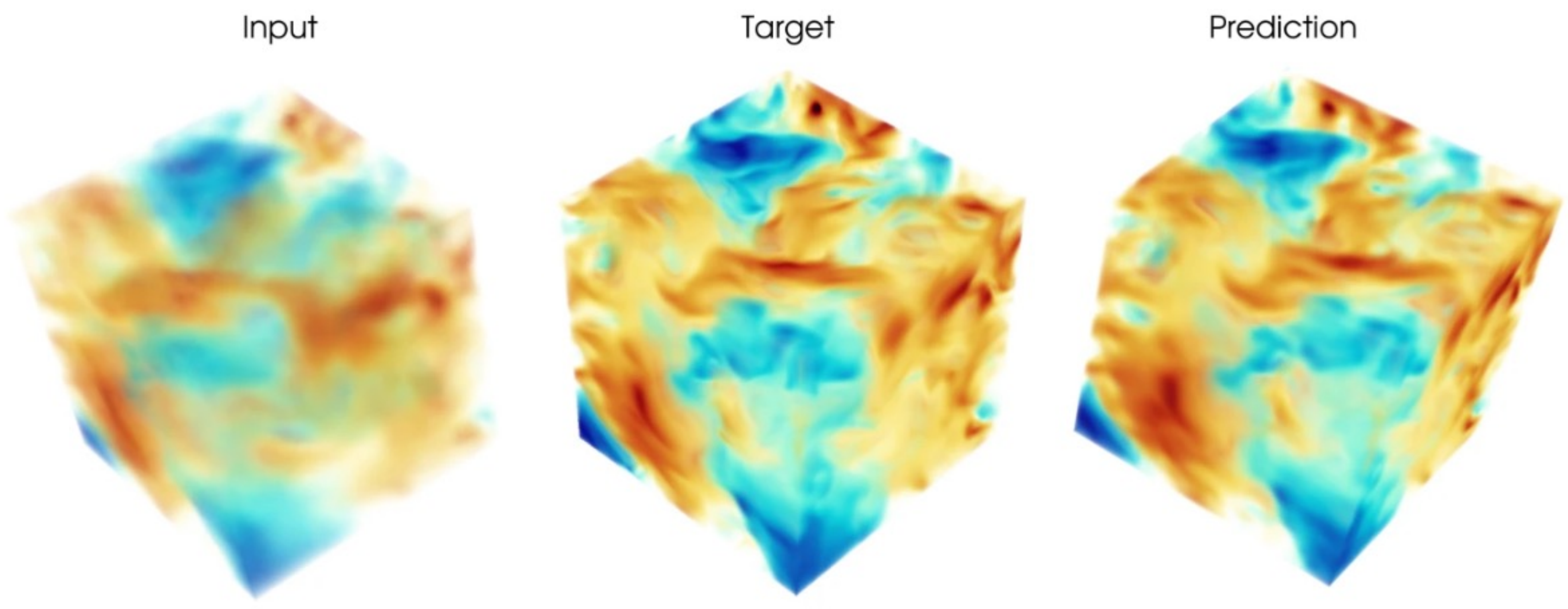
Weather/climate modeling



Linear Elasticity

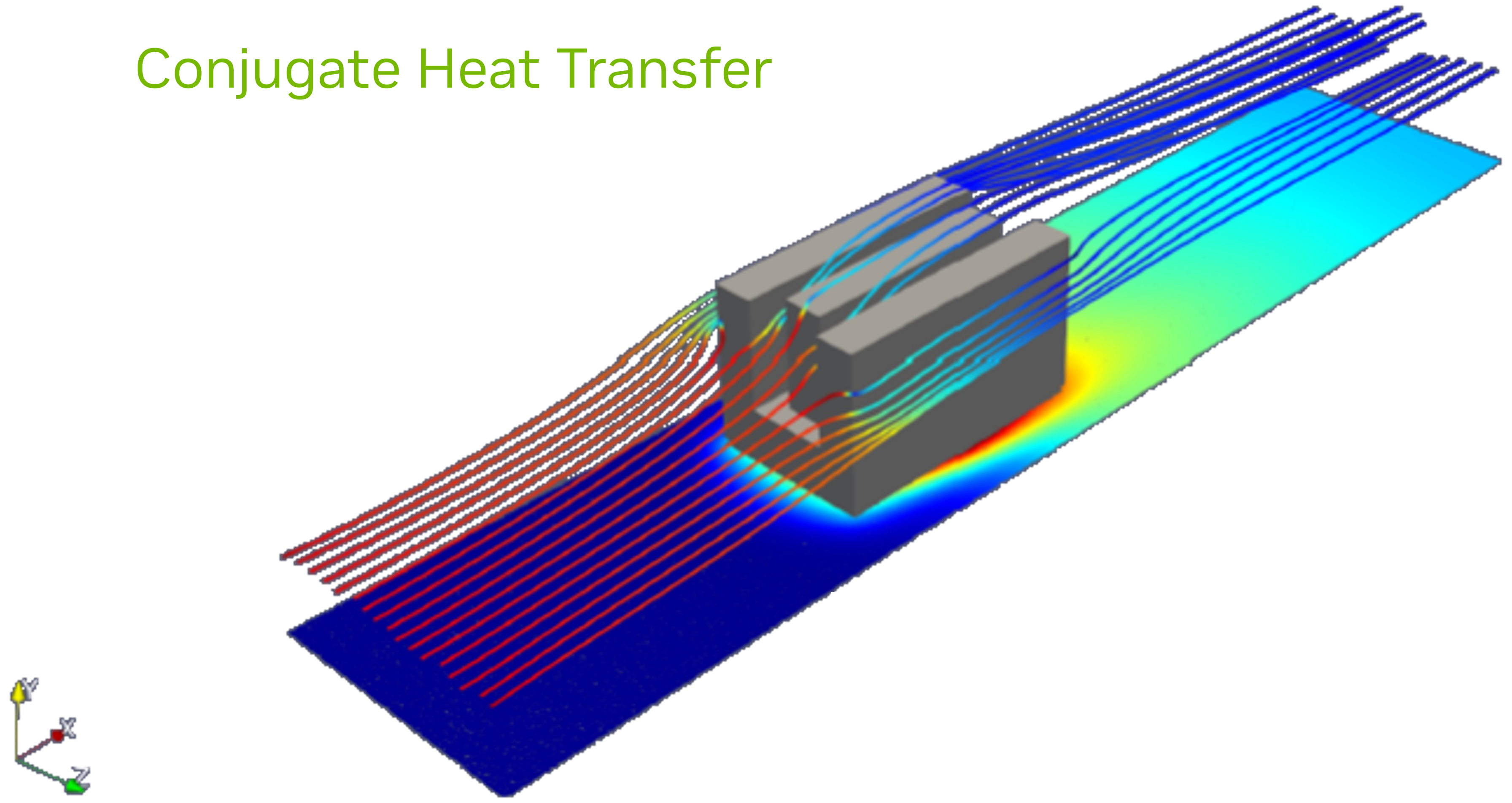


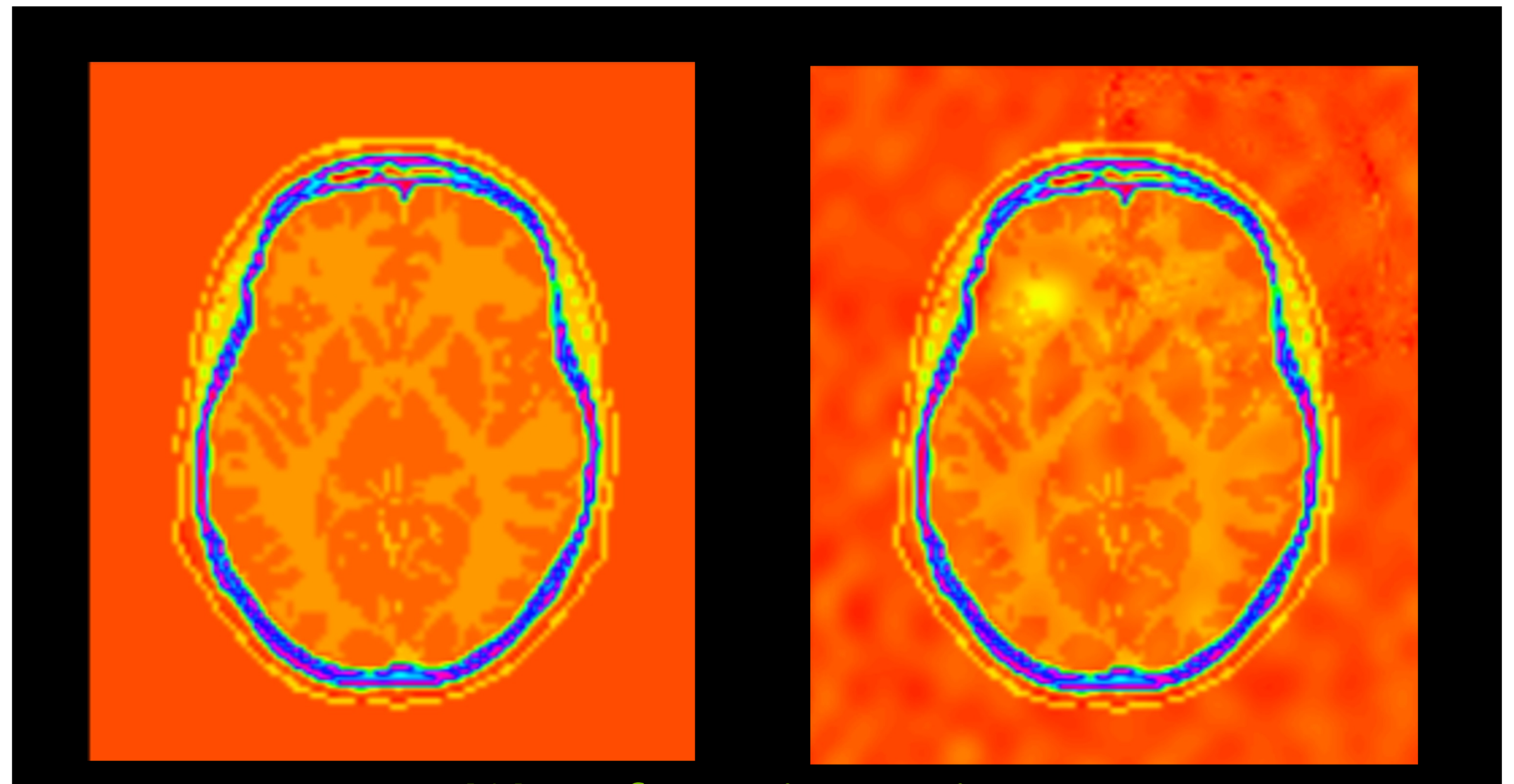
Electromagnetics



Turbulence Super Resolution

Conjugate Heat Transfer



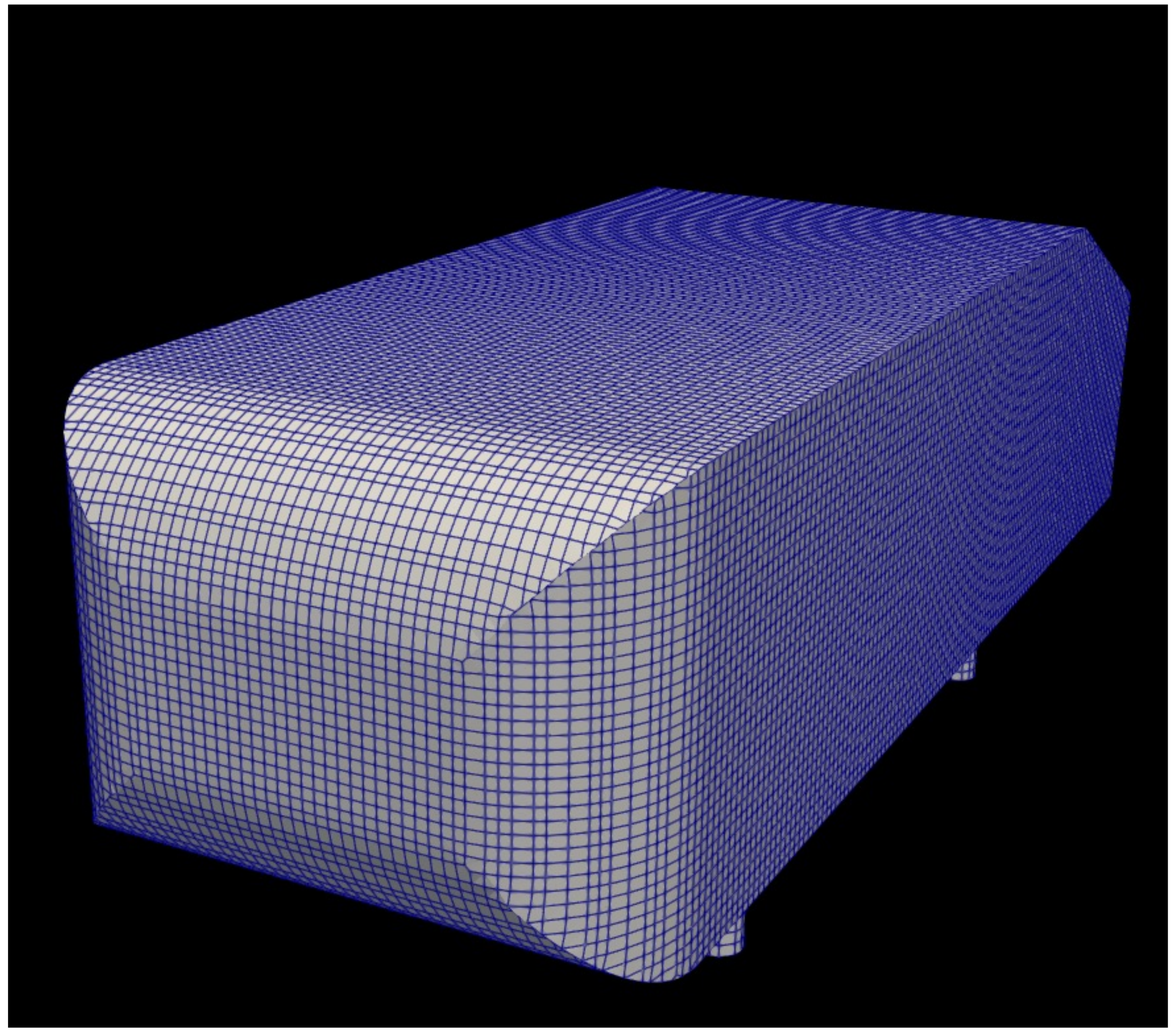


Waveform inversion

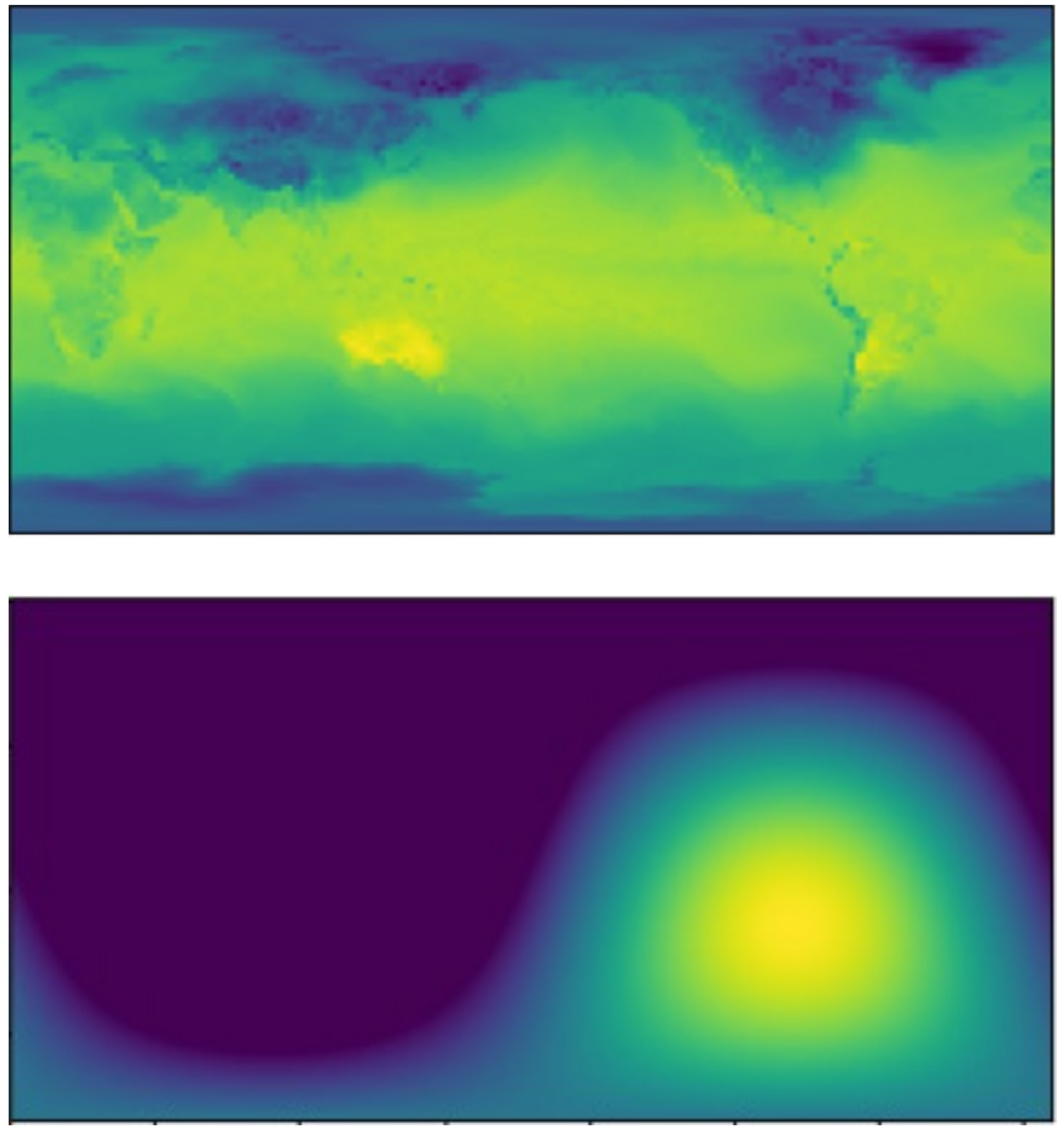
Summary: Modulus Framework

Variety of data handling

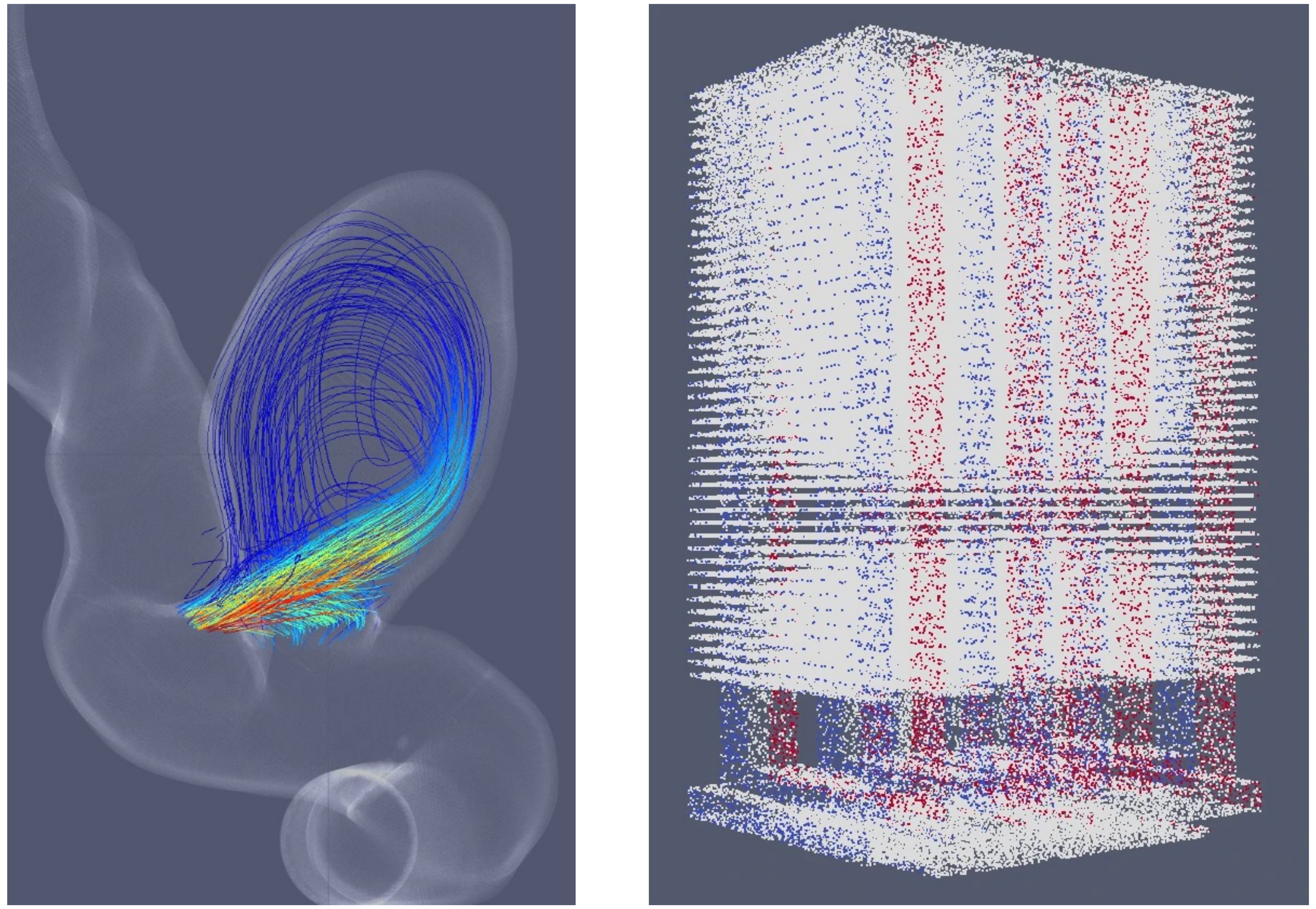
- Optimized datasets and data loaders to work with:
  - Point Clouds
    - Constructive Solid Geometry
    - STL
  - Grid Based
  - Graph/Mesh Based



Ahmed body dataset and data loader (graph)



Weather/climate data loaders (grid)

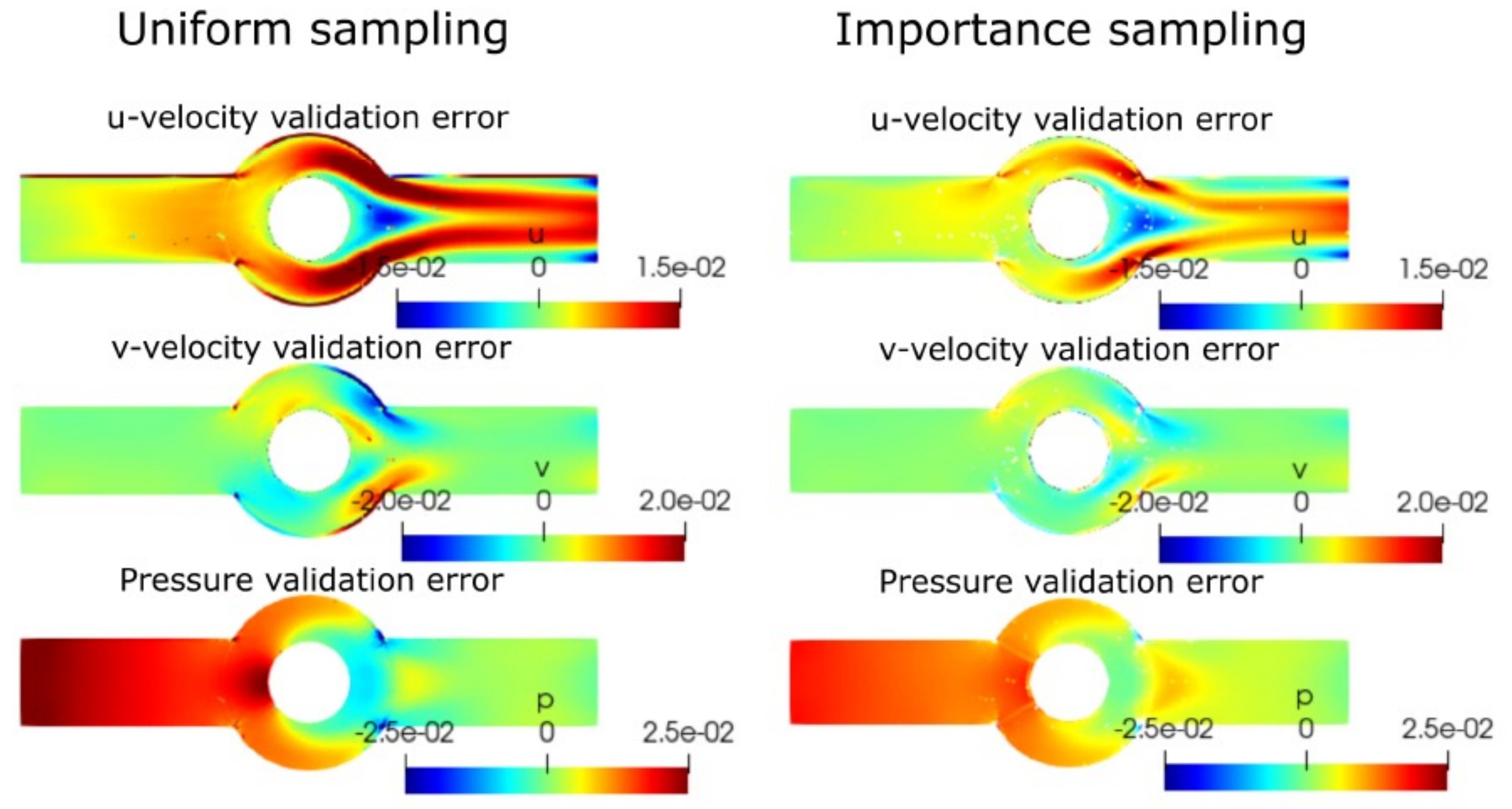


STL and Constructive Solid Geometry (point cloud)

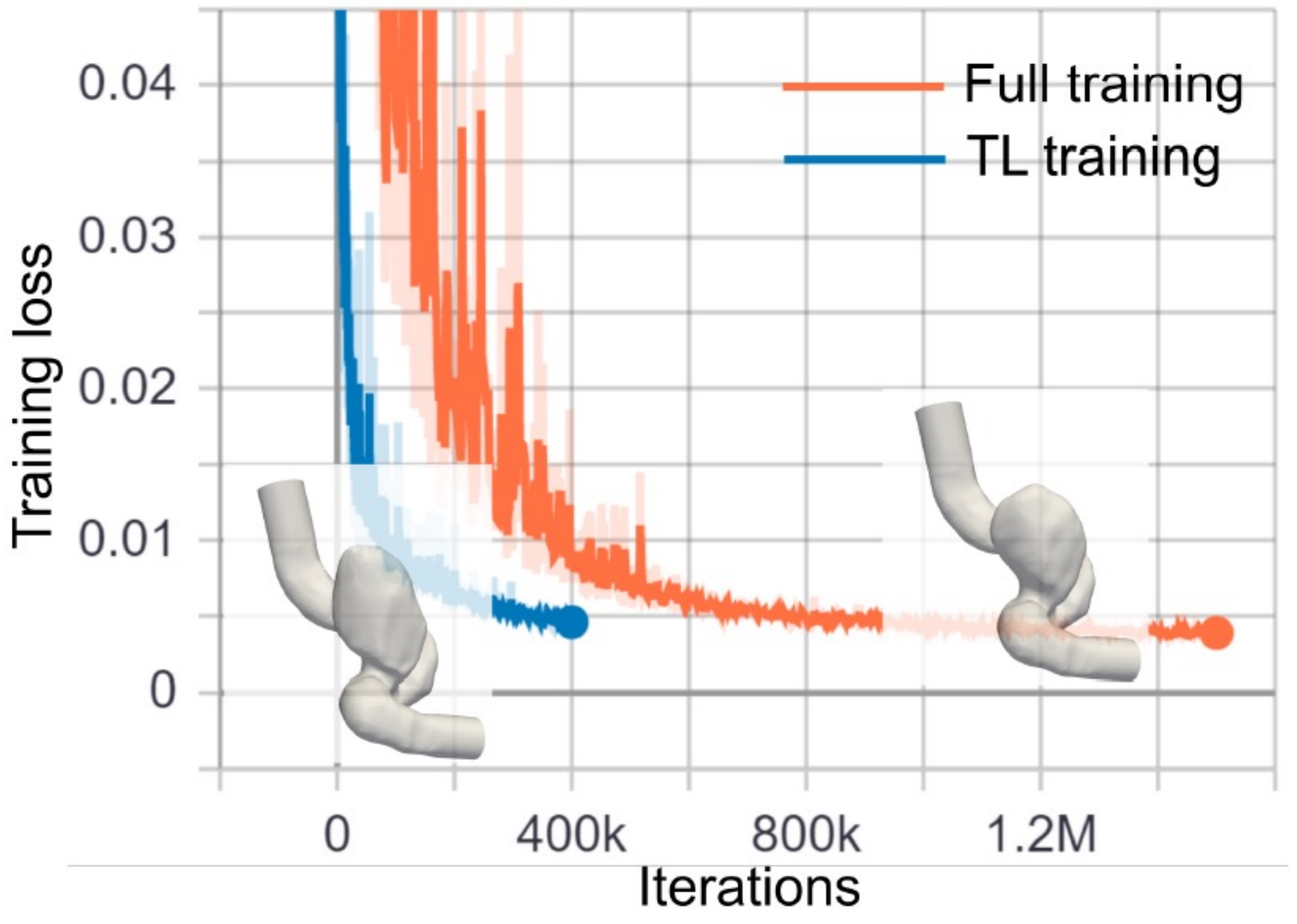
Summary: Modulus Framework

Other miscellaneous features

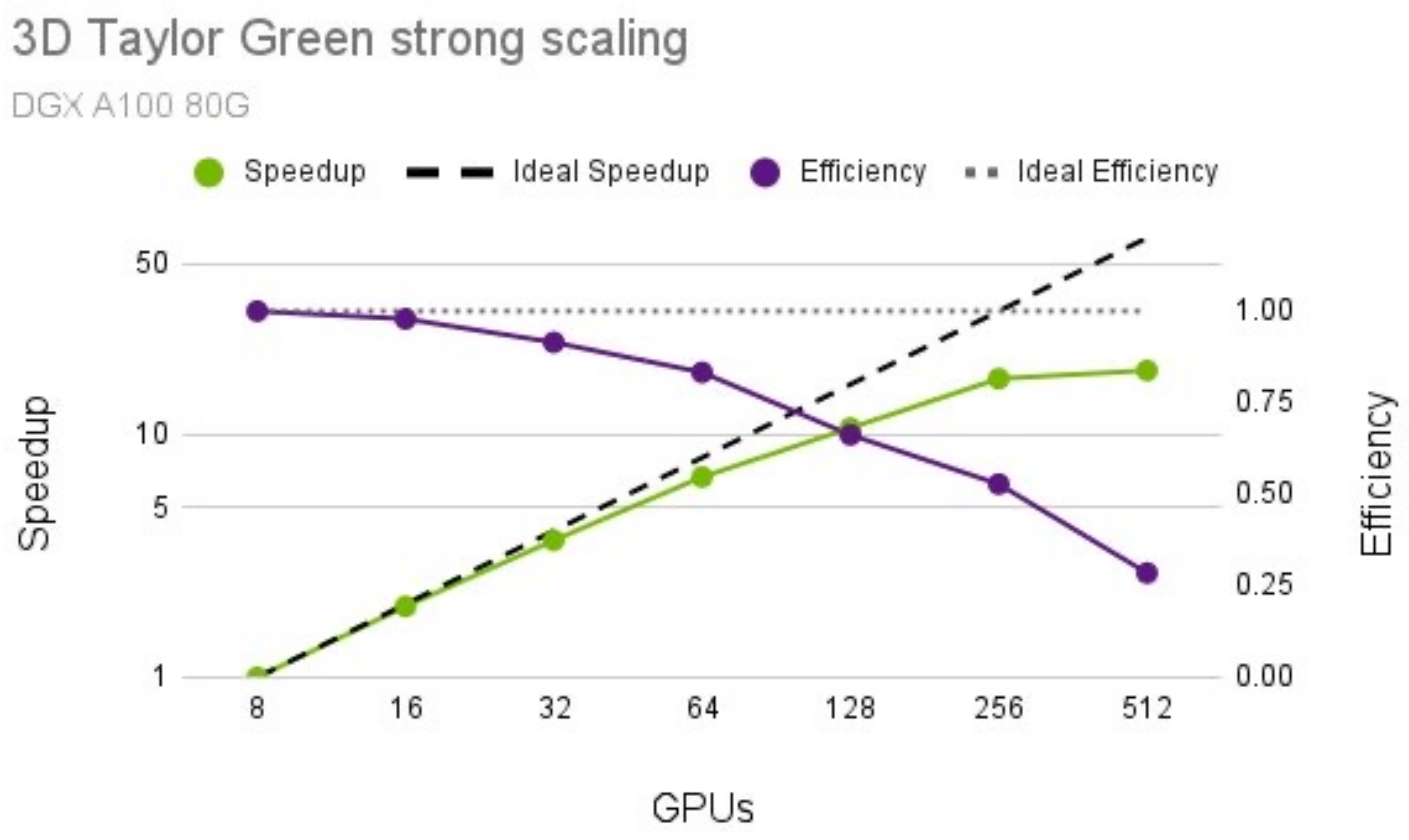
- Distributed utils
- Performance and scaling utils:
  - Abstractions for AMP, CUDA Graphs, JIT
  - Multi-GPU/Multi-Node scaling
  - Gradient aggregation
- Training features:
  - Support for several optimizers
  - Etc.



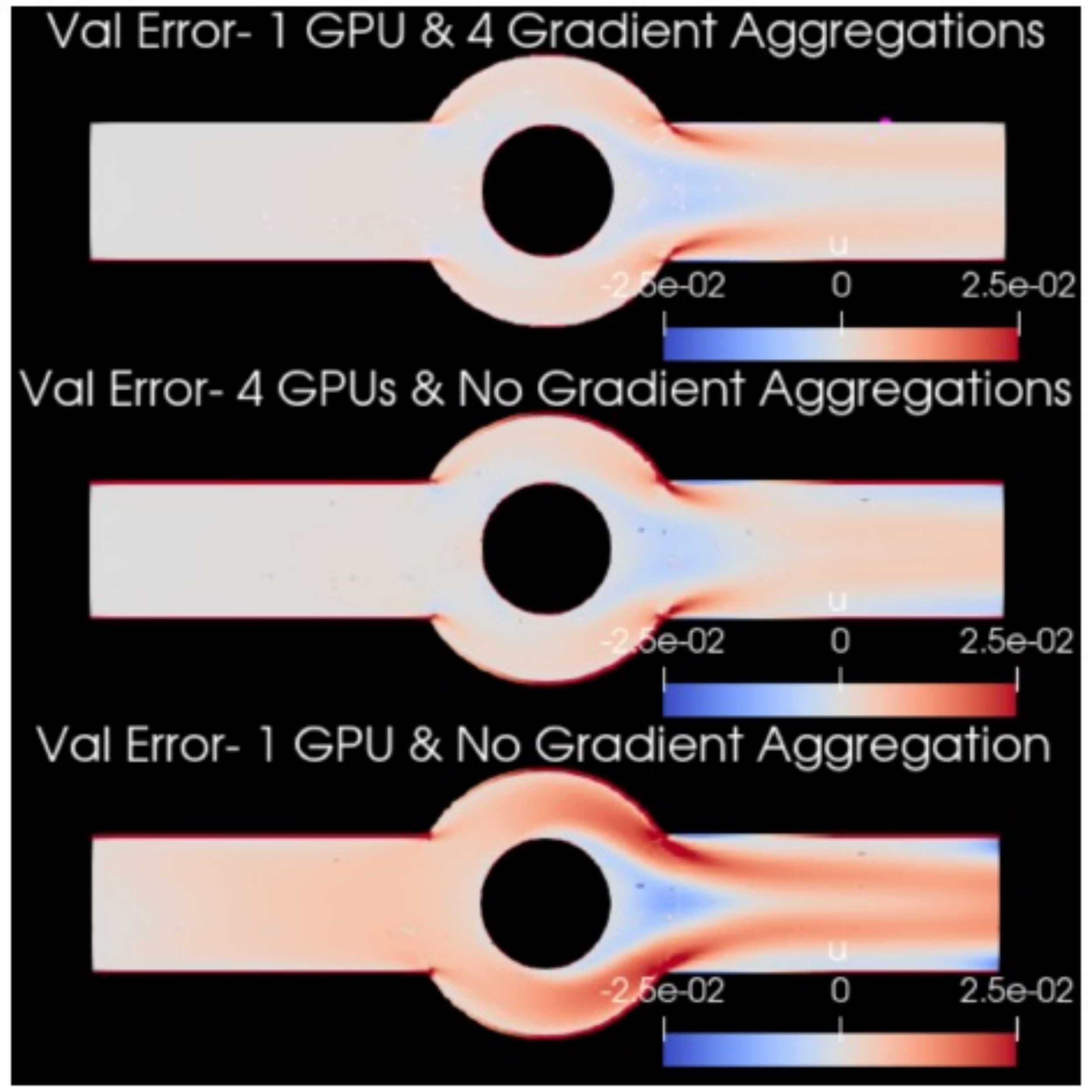
Importance sampling



STL geometry and Transfer Learning



Mulit-GPU, Multi-Node scaling



Gradient Aggregation

Modulus access

Availability via PyPi, NGC Container registry and GitHub

- Download Modulus Docker Container

```
docker run --gpus all -it nvidia/modulus/modulus:24.01 bash
```

- Download Modulus via PyPi

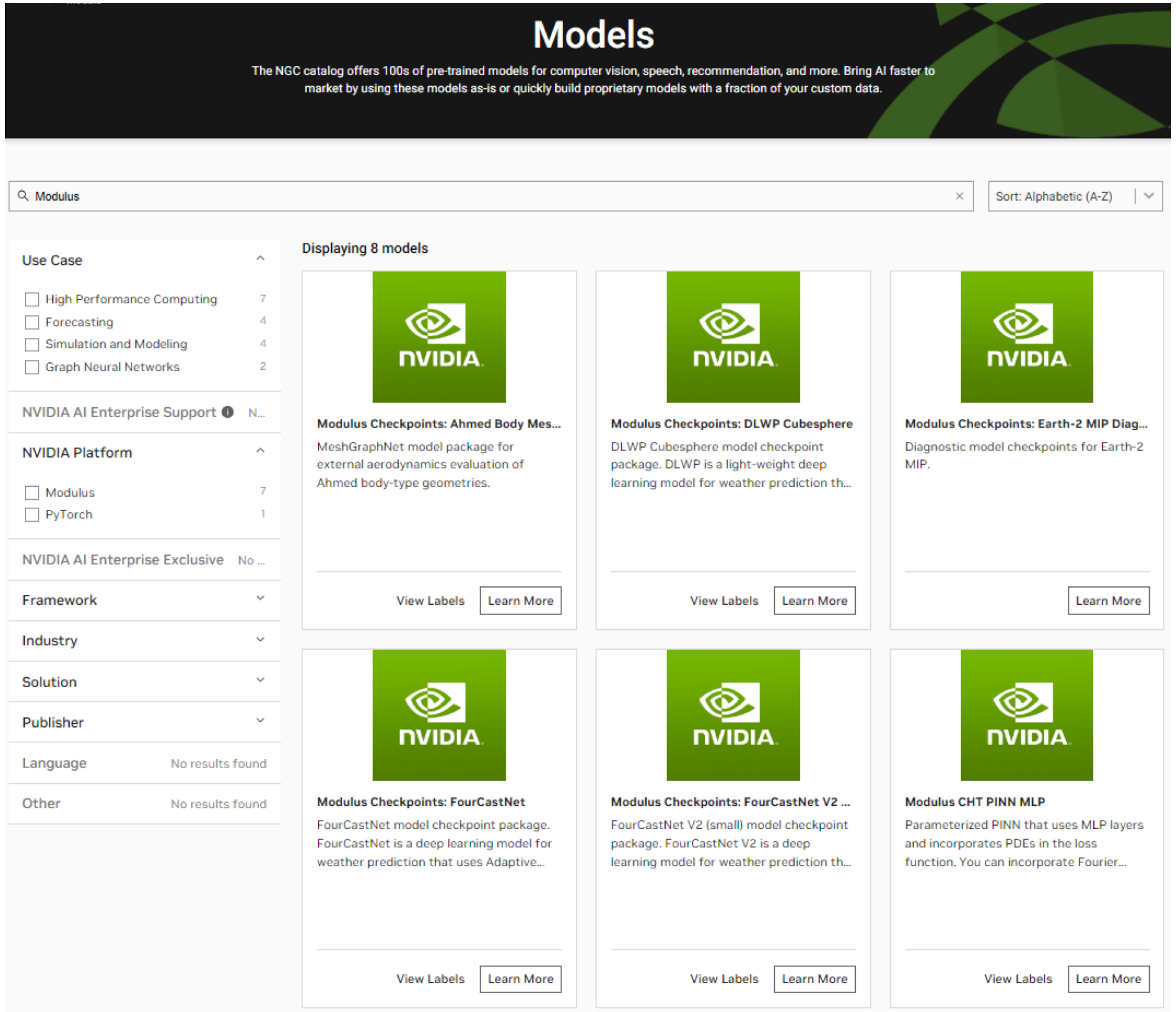
```
pip install nvidia-modulus nvidia-modulus.sym
```

- Develop using Modulus (OSS on GitHub)

```
git clone https://github.com/NVIDIA/modulus.git
```

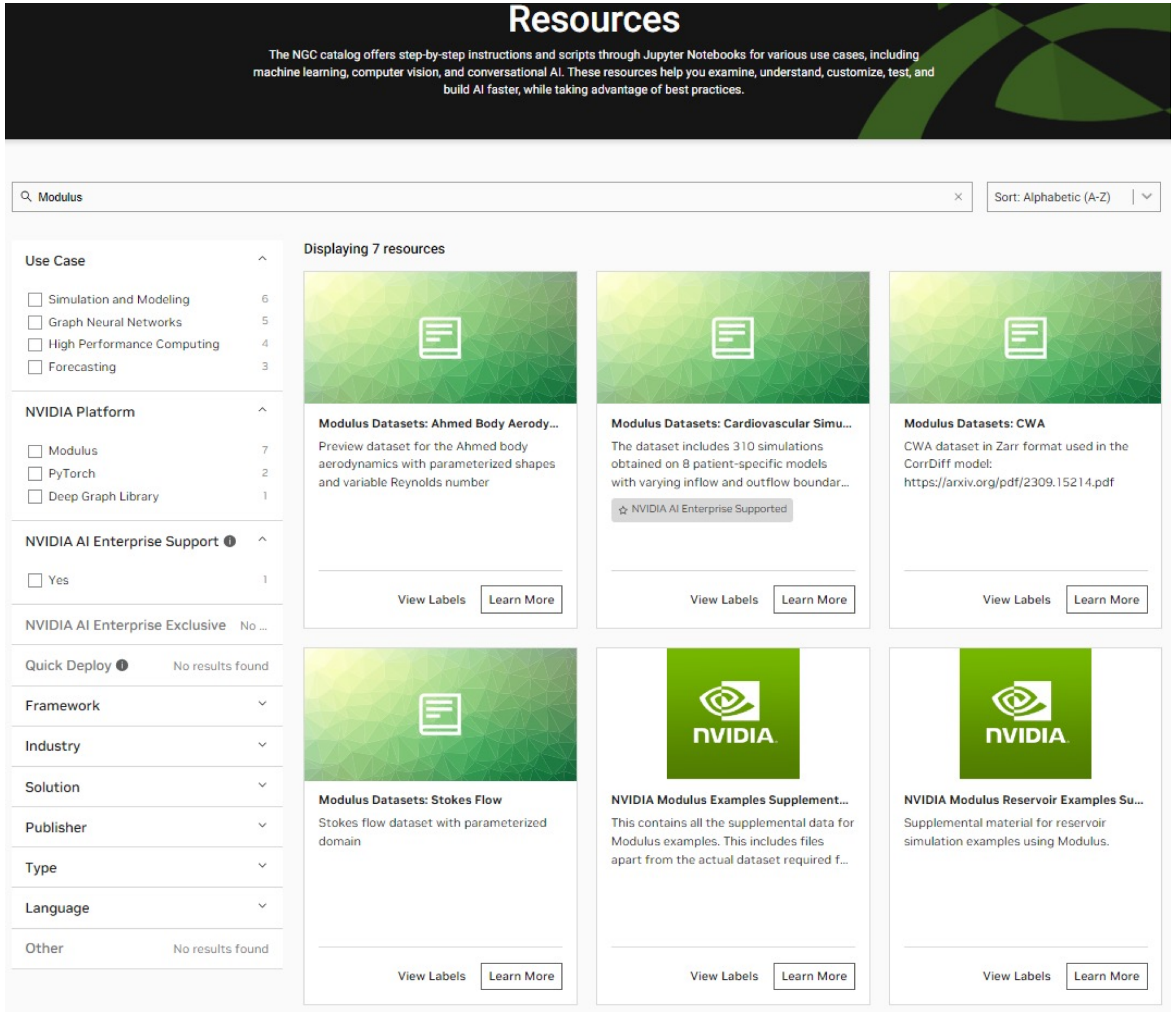
Modulus NGC resources

Pre-trained checkpoints, datasets and resources available via NGC to enable scientific exploration



[Modulus Models on NGC](#)

Pre-trained model packages



[Modulus Resources on NGC](#)

Datasets and supplemental materials

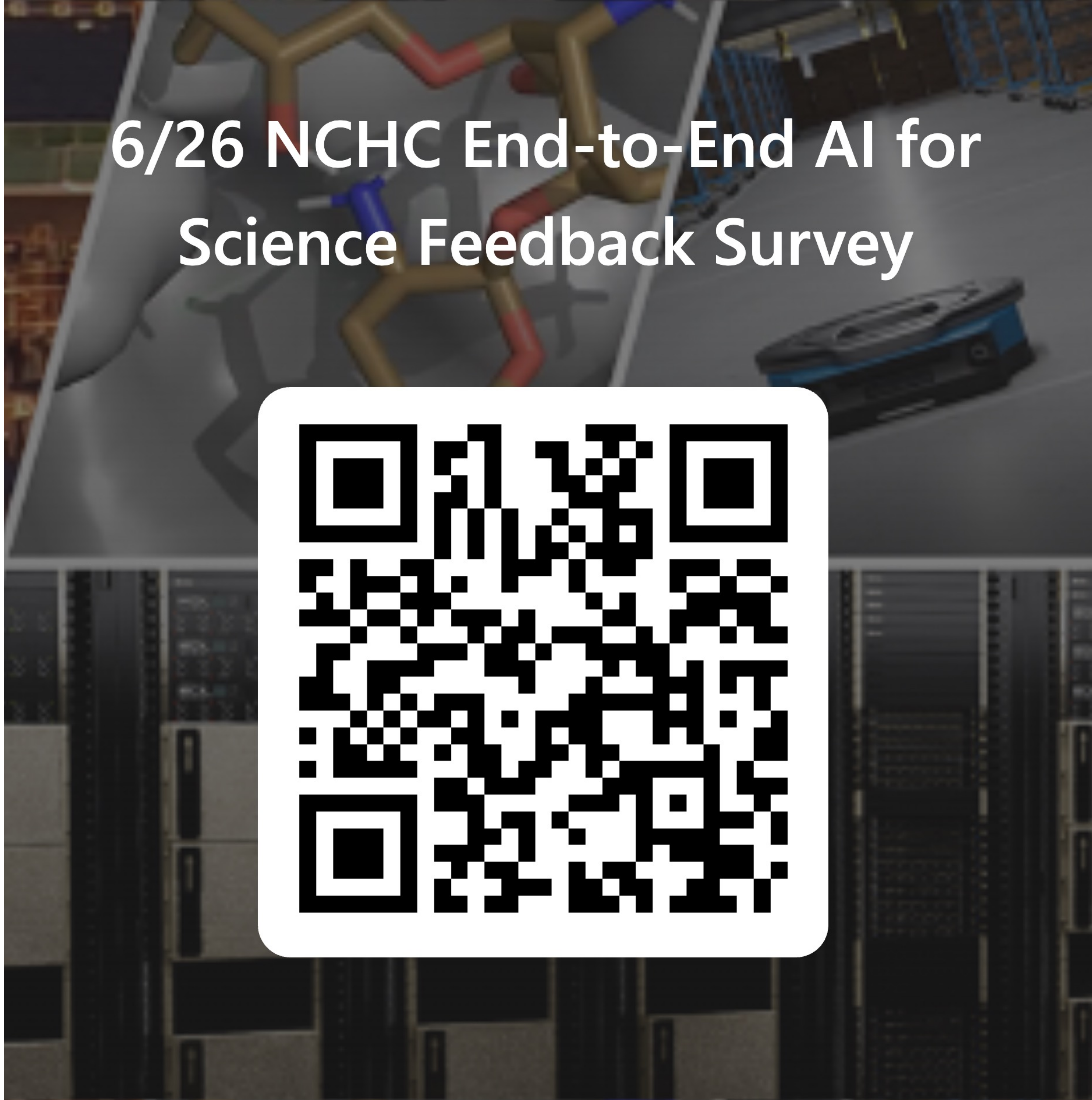
(FREE DLI SELF-PACED COURSE)

INTRODUCTION TO PHYSICS-INFORMED MACHINE LEARNING WITH MODULUS

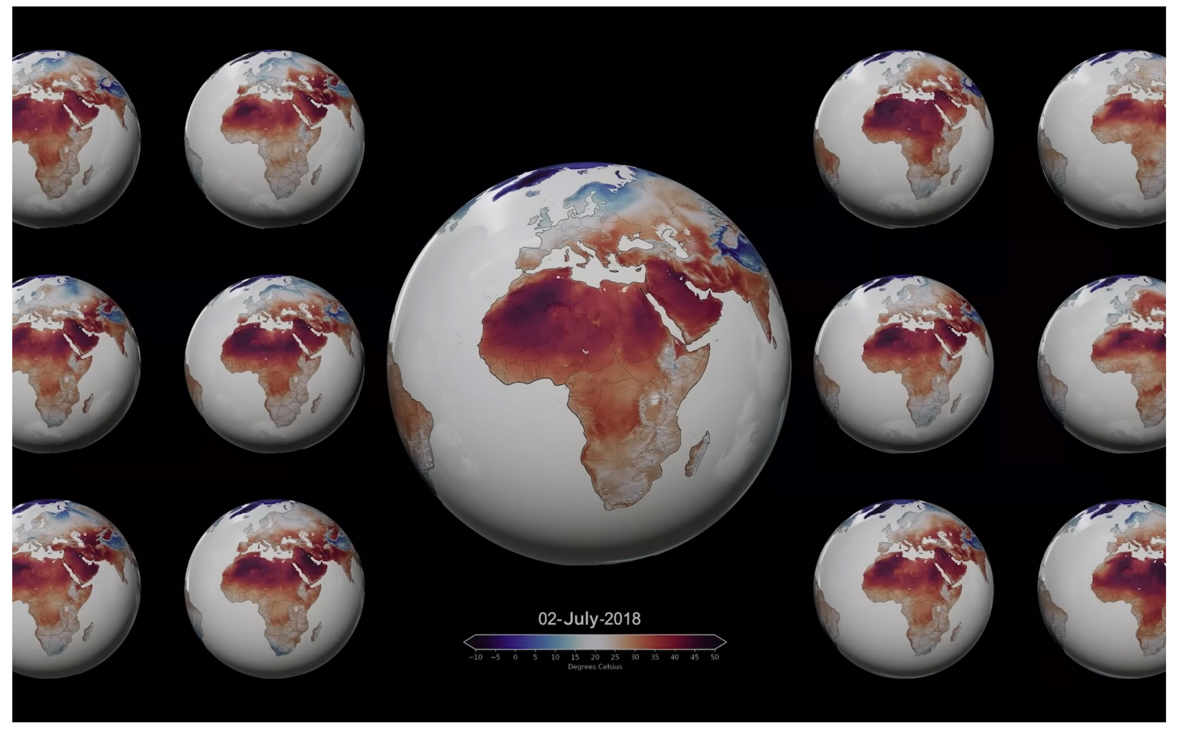
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-OV-04+V1

FEEDBACK SURVEY

https://forms.office.com/r/ZJ6a4SjdWT



Physics-ML Success stories - Modulus Case-Studies

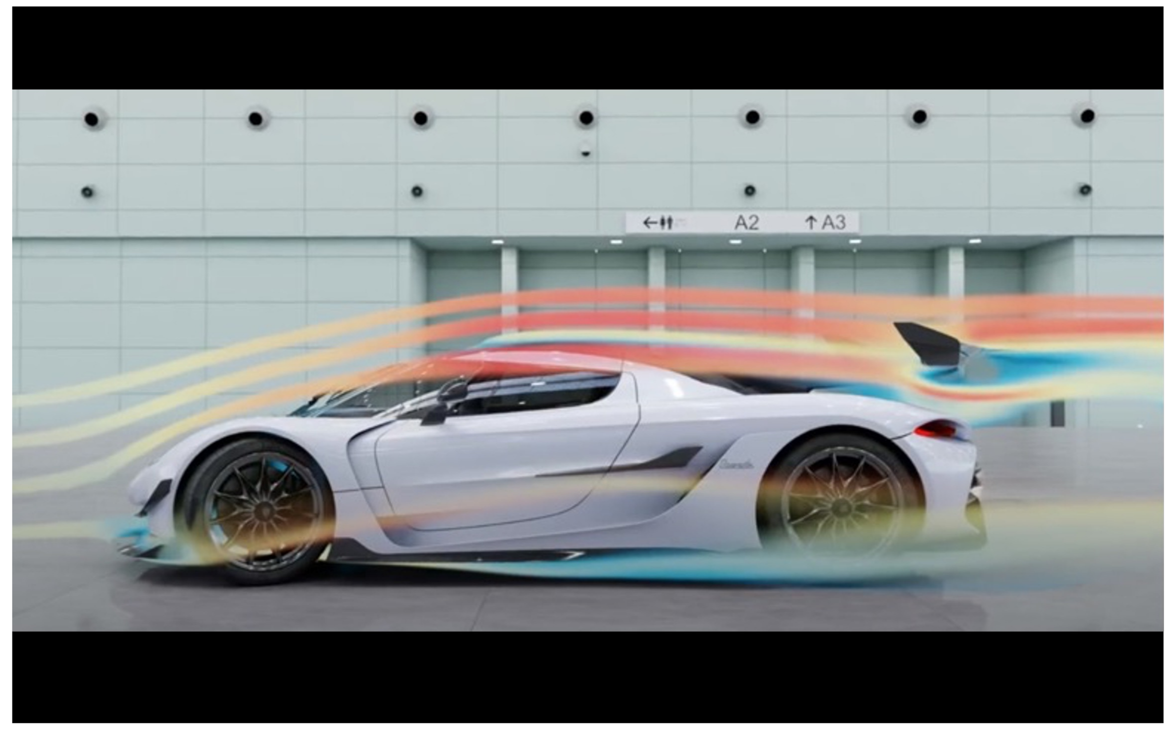


Weather modeling

Demo: Link 1, Link 2, Link 3

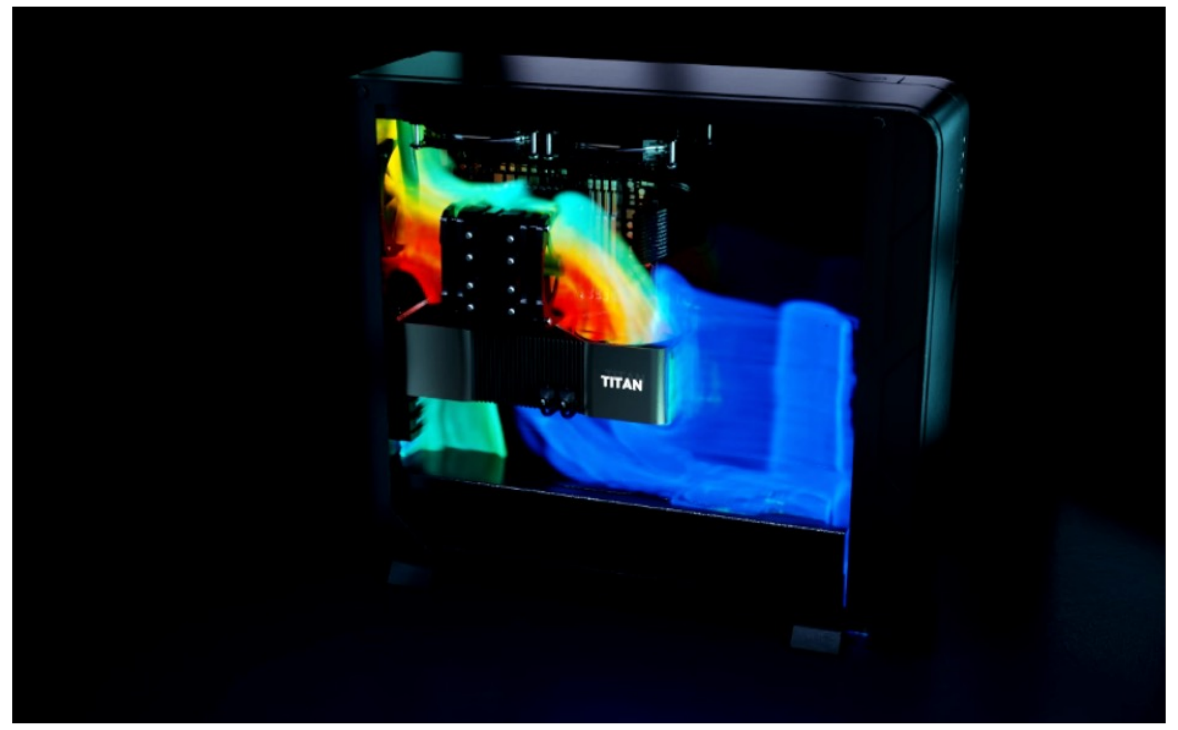
Wind farm Super Resolution

Demo: Link, Blog: Link



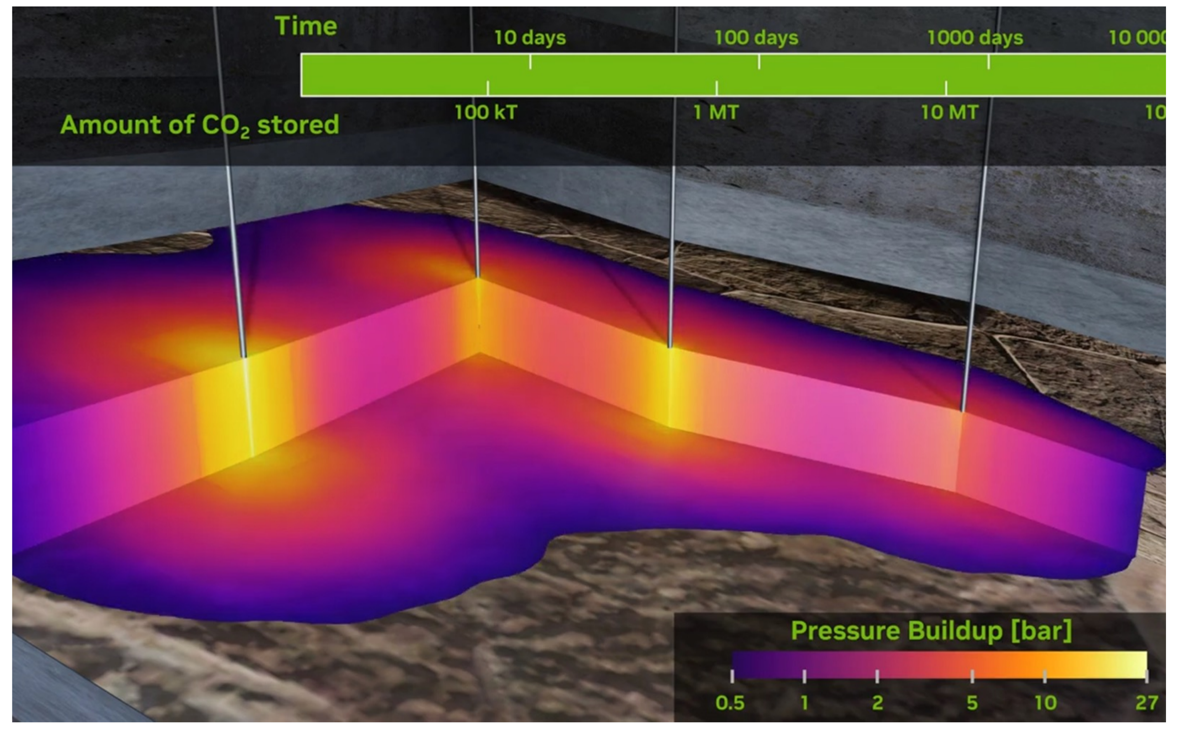
Automotive CFD

GTC 2024 Keynote



RTX 4090 heat sink design

Demo: Link

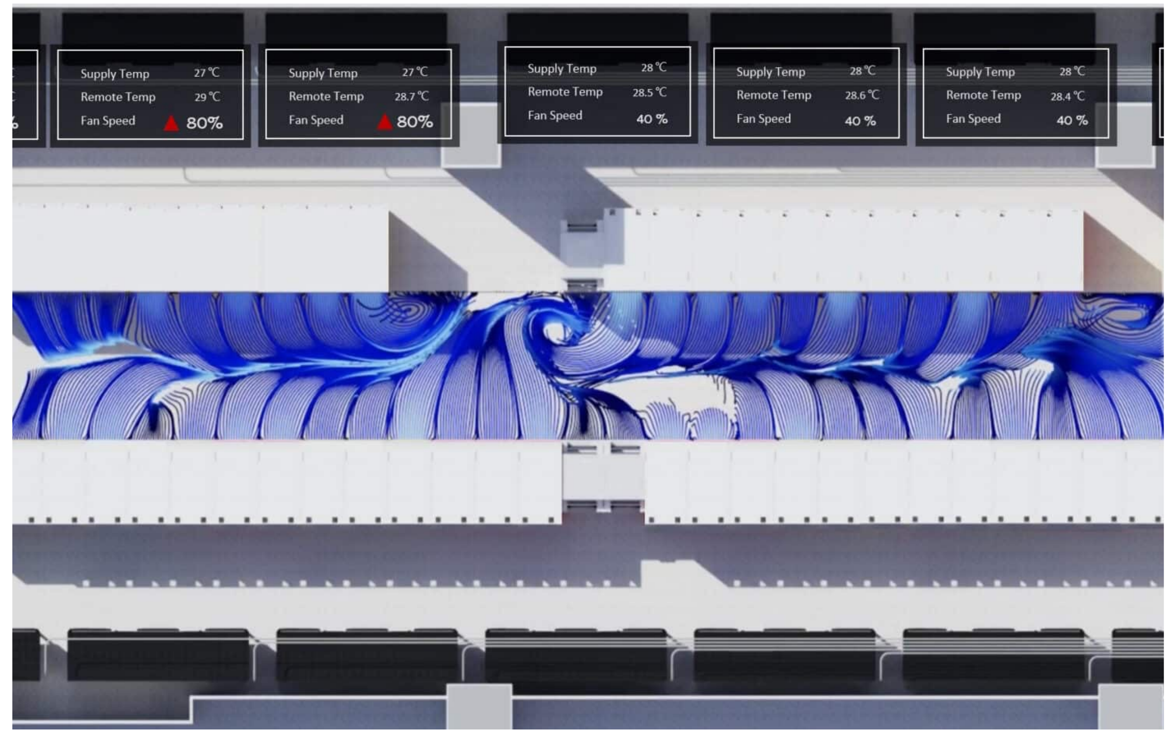


Carbon capture and storage

Demo: Link, Blog: Link

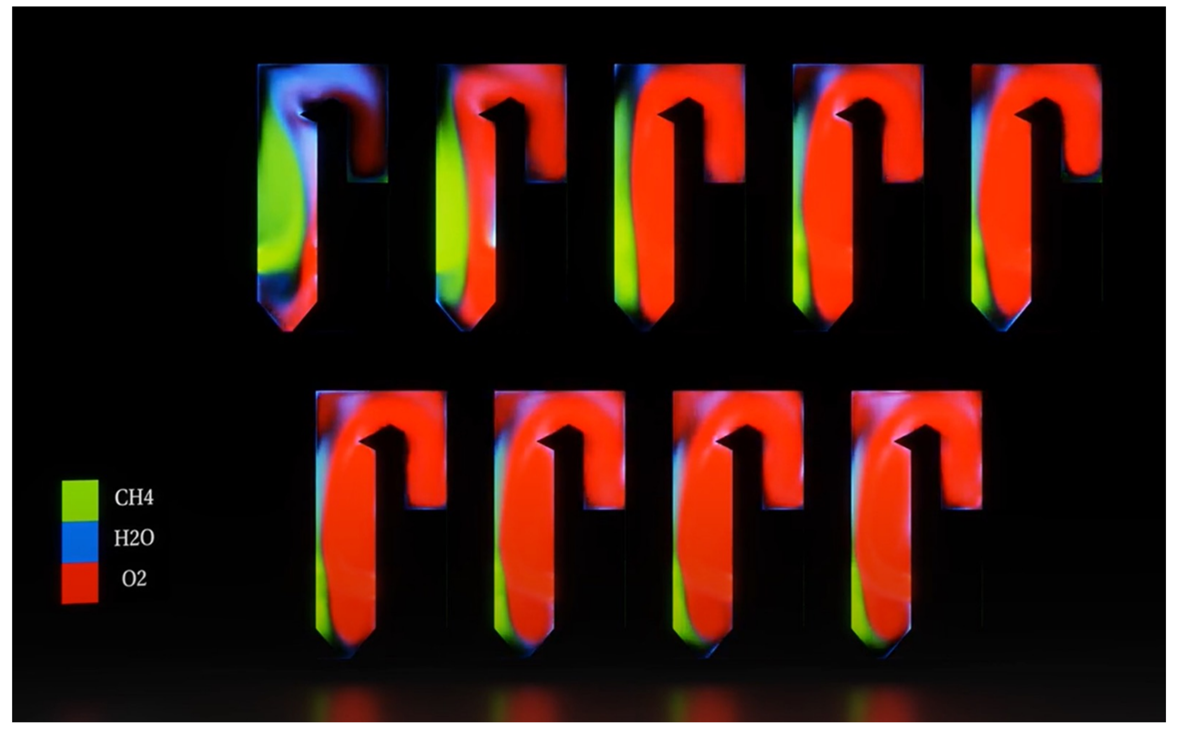
HRSG Digital Twin

Demo: Link, GTC Session: Link



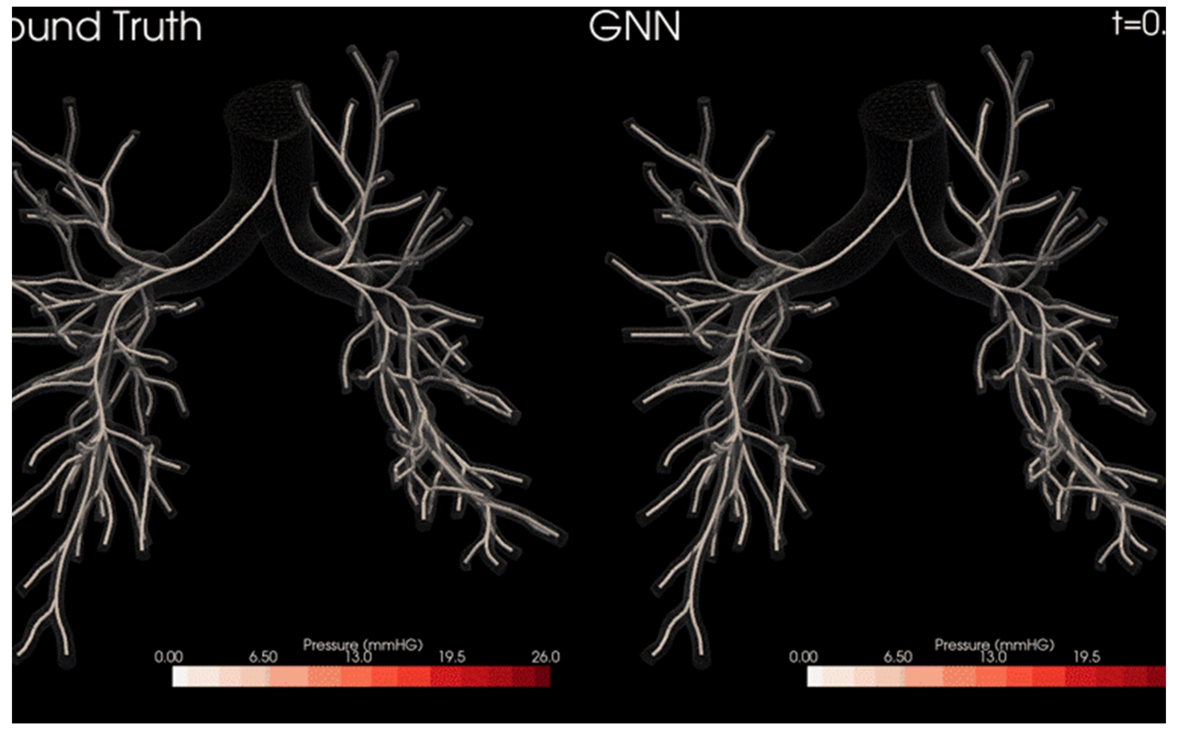
Data Center Digital Twin

Blog: Link, GTC Session: Link



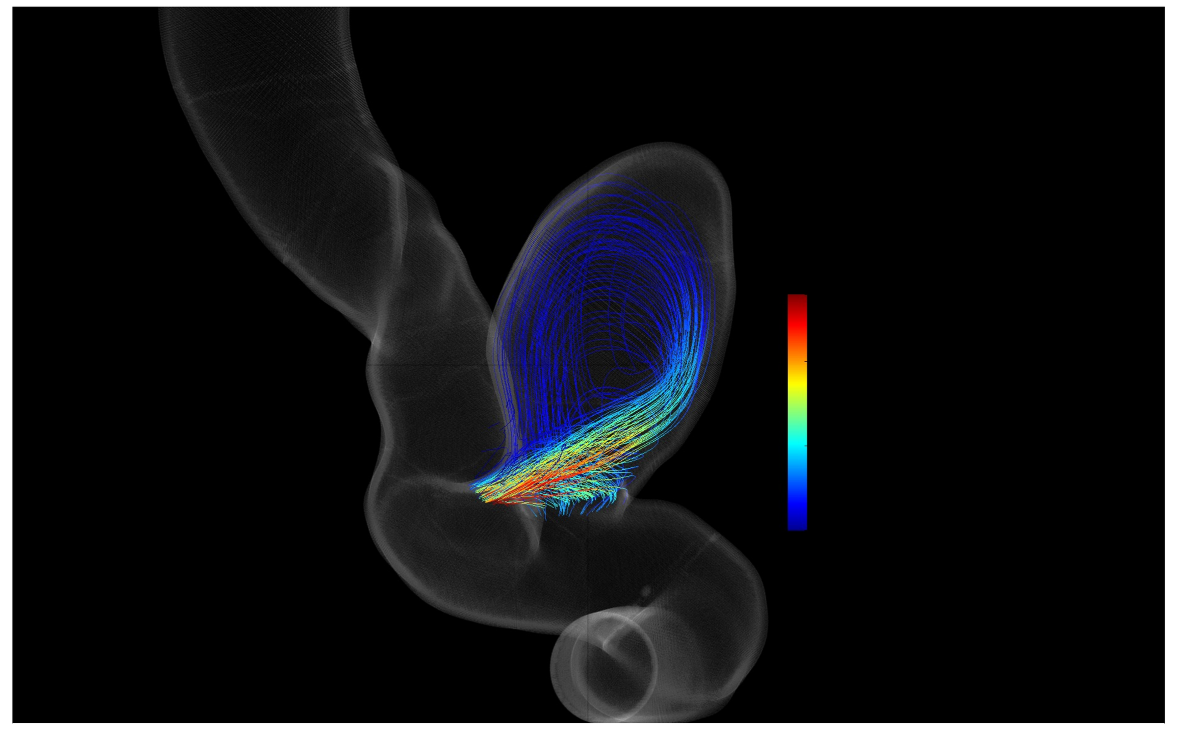
Thermal Boiler Digital Twin

Blog: Link, GTC Session: Link



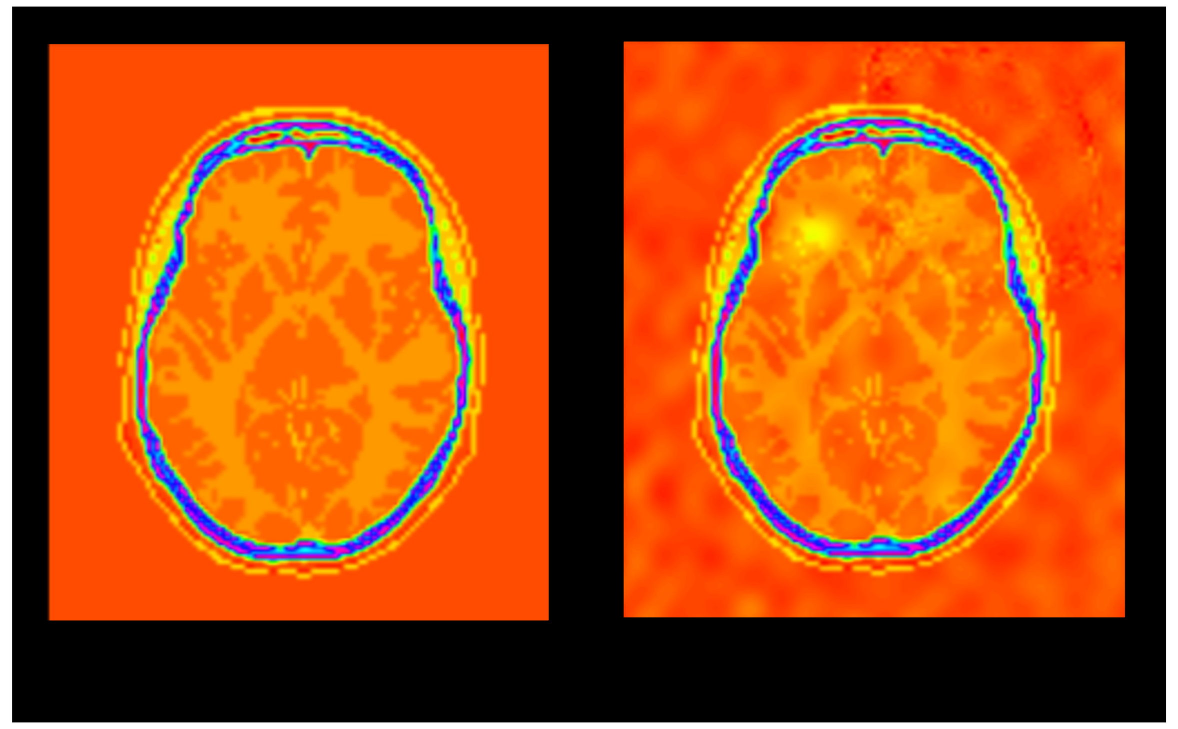
Cardiovascular Simulation

Blog: Link



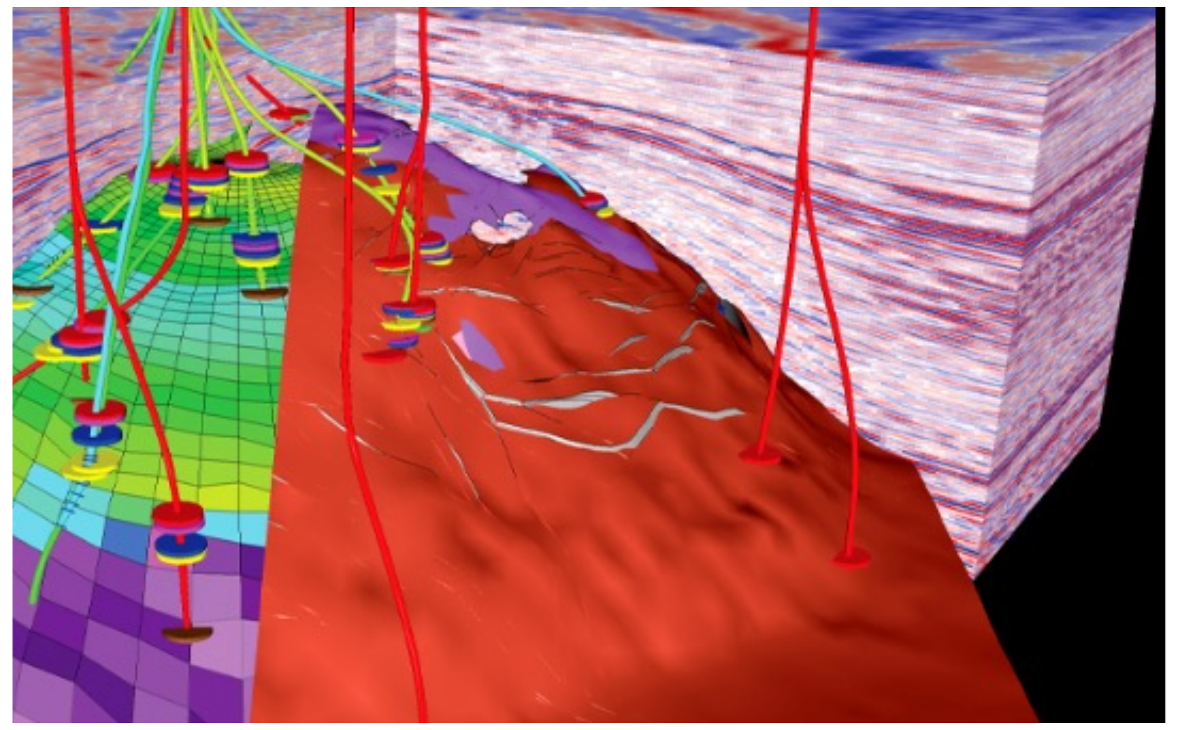
Brain Aneurysm Simulation

Demo: Link



Brain Anomaly Detection

Resource: Link



Sub surface simulations

Resource: Link

NVIDIA

THANK YOU