Applications V: Neural Operators

Jonathan Bedford,
Tectonic Geodesy working group,
Ruhr University Bochum,
Germany

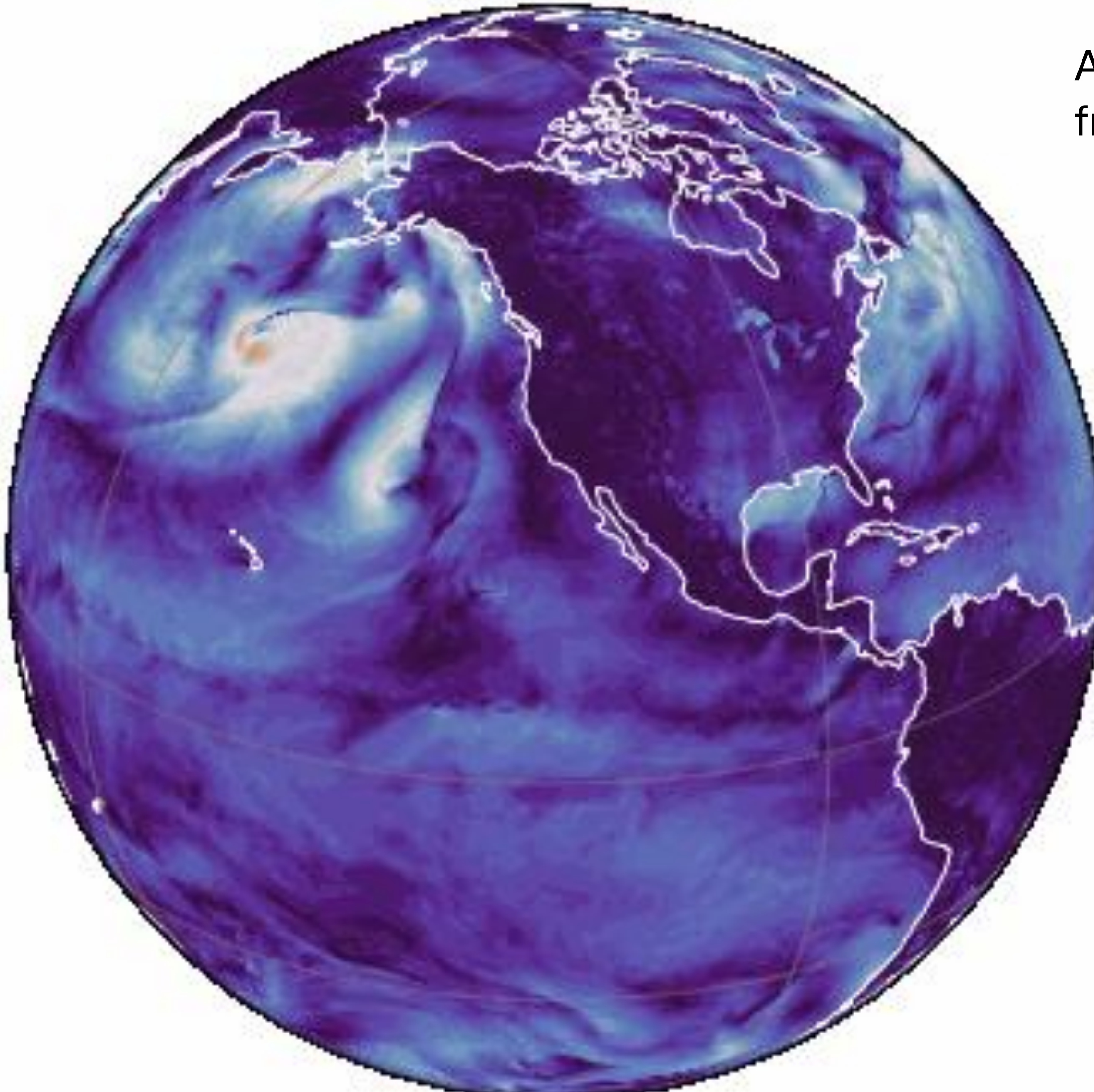
Presentation for International Training School

AI 4 Seismology, 07.05.2025

GIF from https://github.com/NVIDIA/torch-harmonics; Spherical Fourier Neural Operator

(fairly) new type of architecture

Autoregression framing



Relevant to situations involving physics

Past atmosphere params. -> future atmosphere params.

Neural operators

- What is a neural operator?
- Neural operator architectures
- Literature examples of NO applications
- Applications in seismology and related research
- Discussion

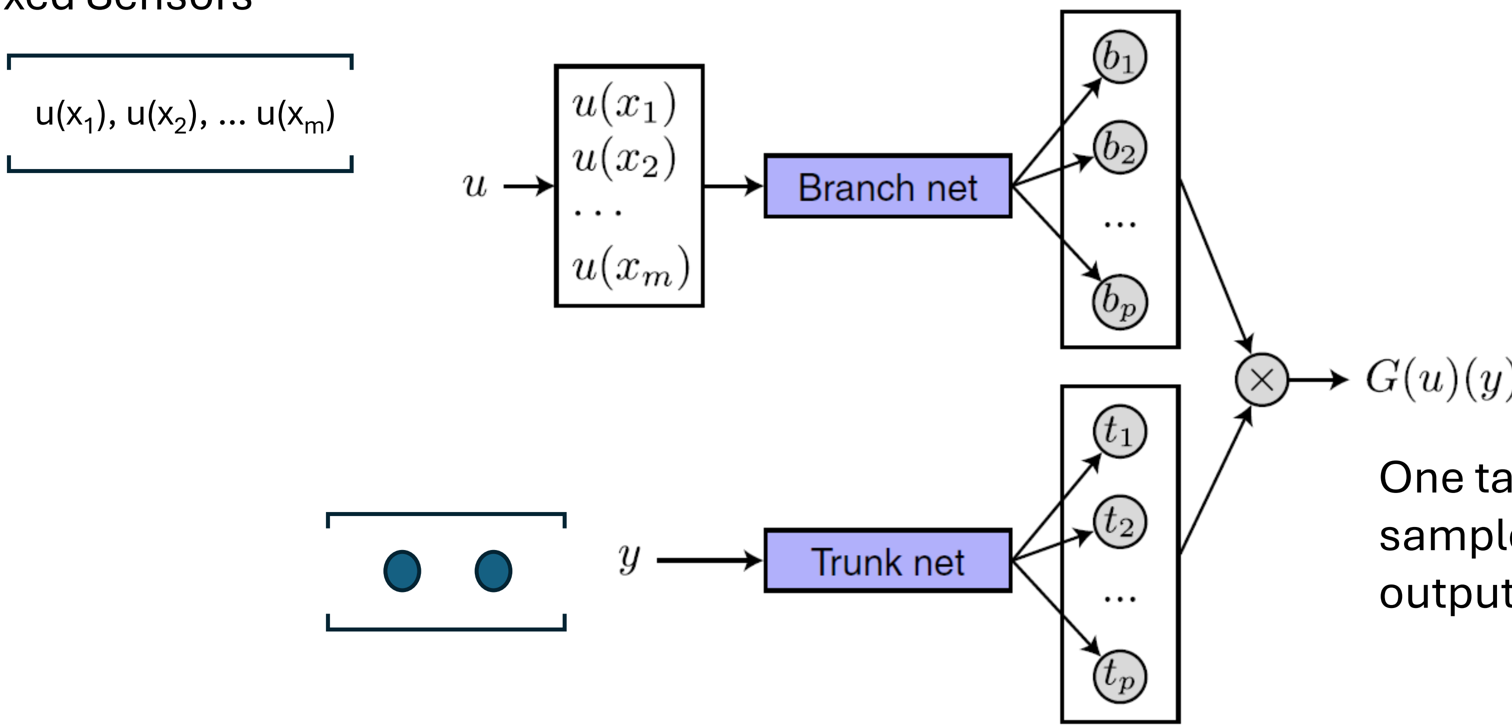
Neural operators

- What is a neural operator?
- Neural operator architectures
- Literature examples of NO applications
- Applications in seismology and related research
- Discussion

What is a Neural Operator?

“Deep-O-Net”

“Fixed Sensors”



One target sample as output

Single point in the domain where your output function exists (e.g. a location in 2D space)

[Lu et al. 2021, Nature Machine Intelligence]

What is a Neural Operator?

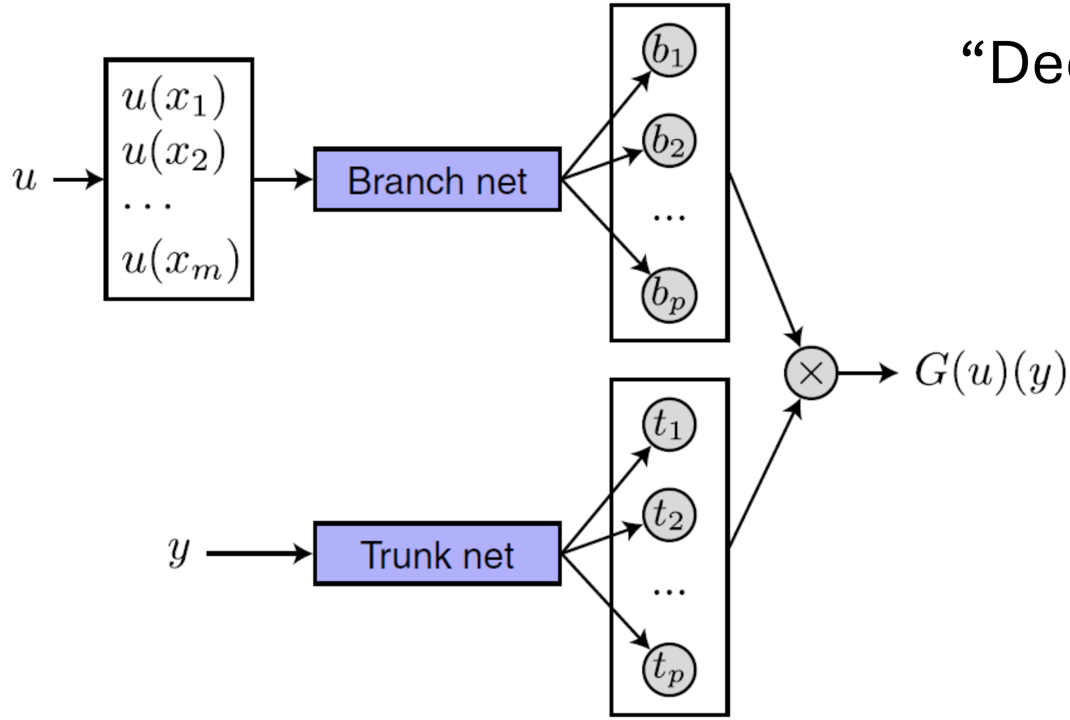
Universal Approximation Theorem
for Operator:

$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{\text{branch}} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{\text{trunk}} \right| < \epsilon$$

∈ can be minimized so that the neural operator approximates the true operator

[Lu et al. 2021, Nature Machine Intelligence]

What is a Neural Operator?



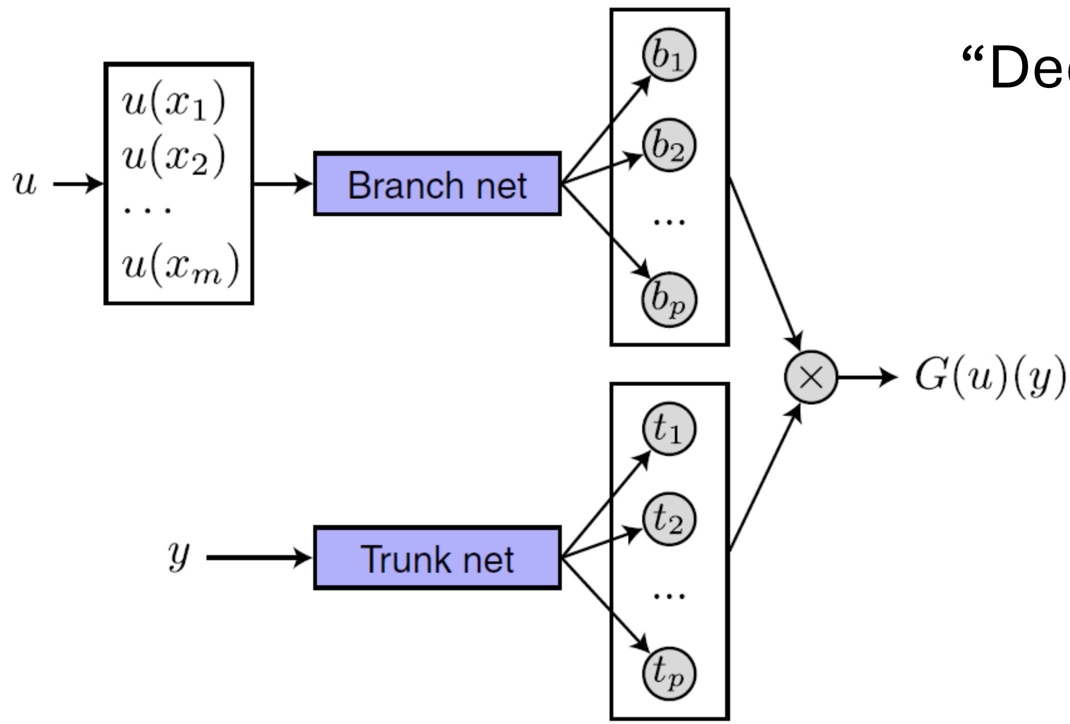
"Deep-O-Net"

$$G(u)(y) \approx \sum_{k=1}^{p} \underbrace{b_k(u(x_1), u(x_2), \ldots, u(x_m))}_{\text{branch}} \underbrace{t_k(y)}_{\text{trunk}}$$

[Lu et al. 2021, Nature Machine Intelligence]

What is a Neural Operator?

“Deep-O-Net”



U(x)

$\downarrow$

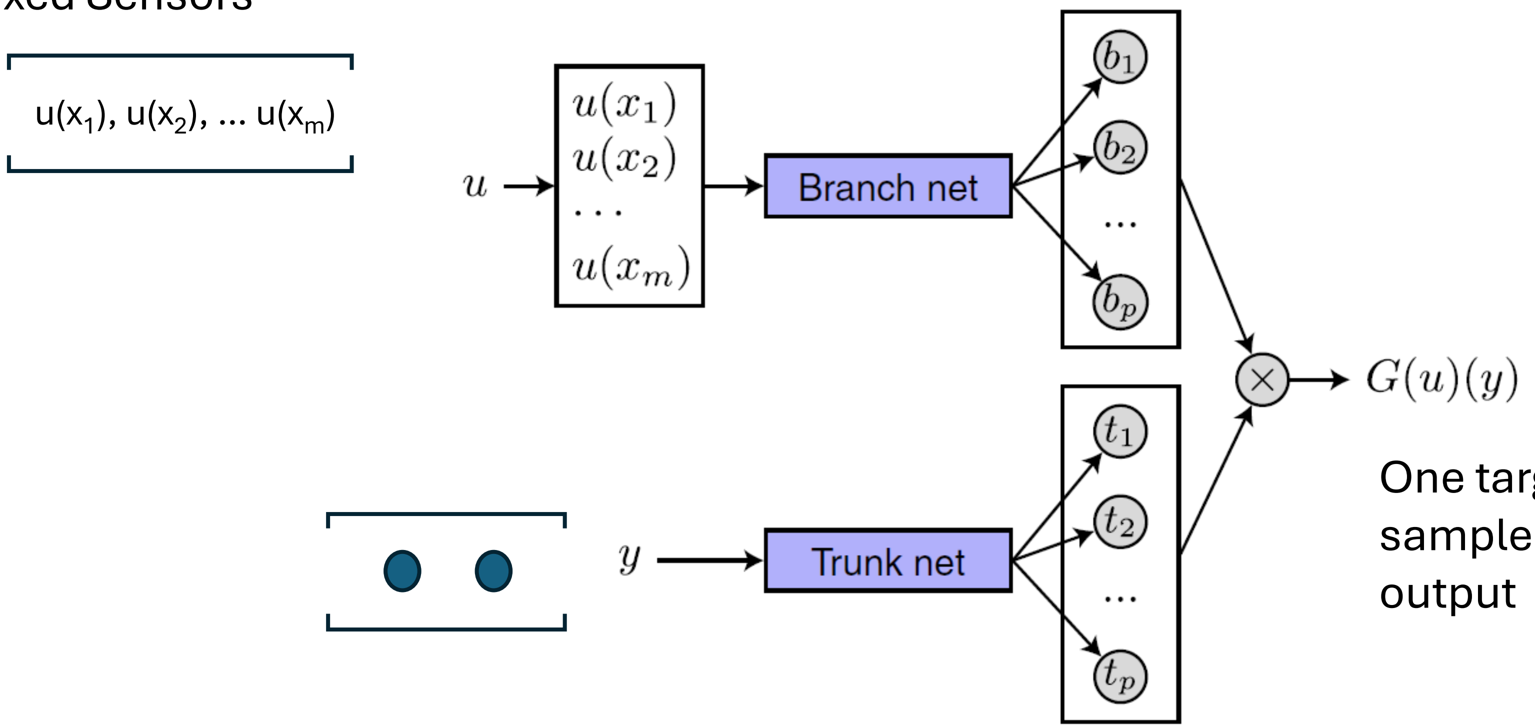
V(y)

$$V(y) = G(u)(y) \approx \sum_{k=1}^{p} \underbrace{b_k(u(x_1), u(x_2), \dots, u(x_m))}_{\text{branch}} \underbrace{t_k(y)}_{\text{trunk}}$$

[Lu et al. 2021, Nature Machine Intelligence]

Training a deep-o-net

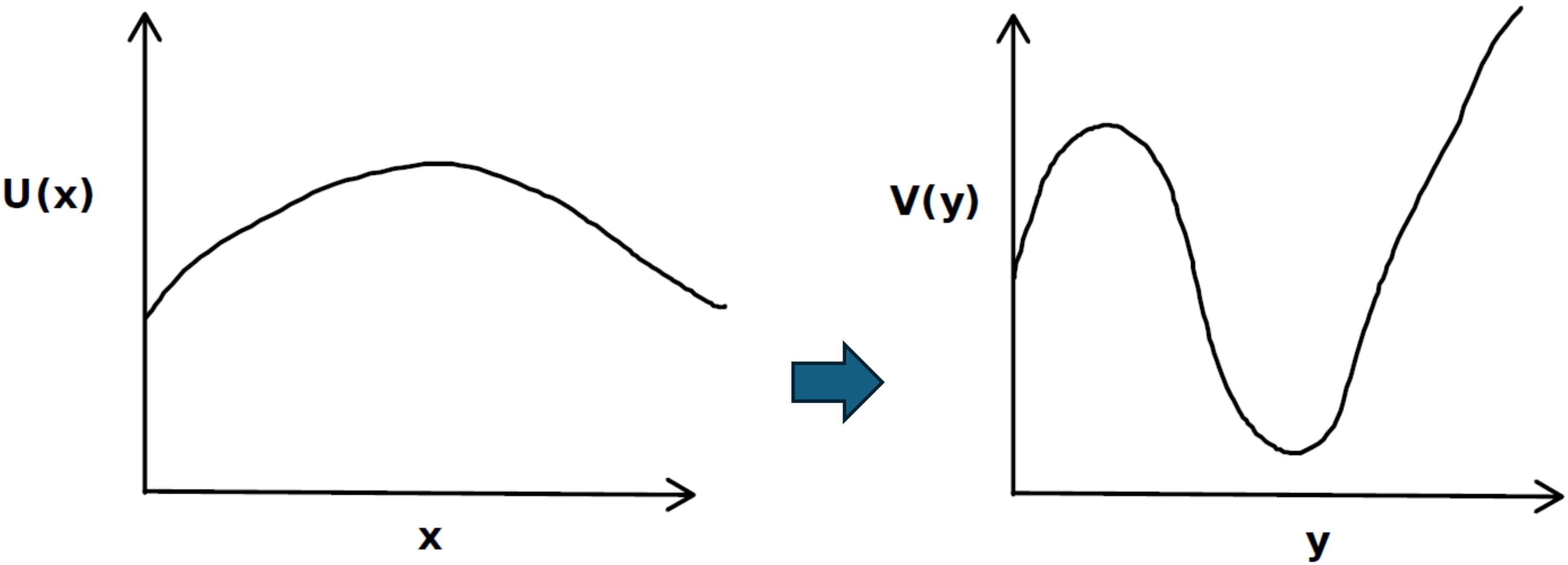
"Fixed Sensors"



One target sample as output

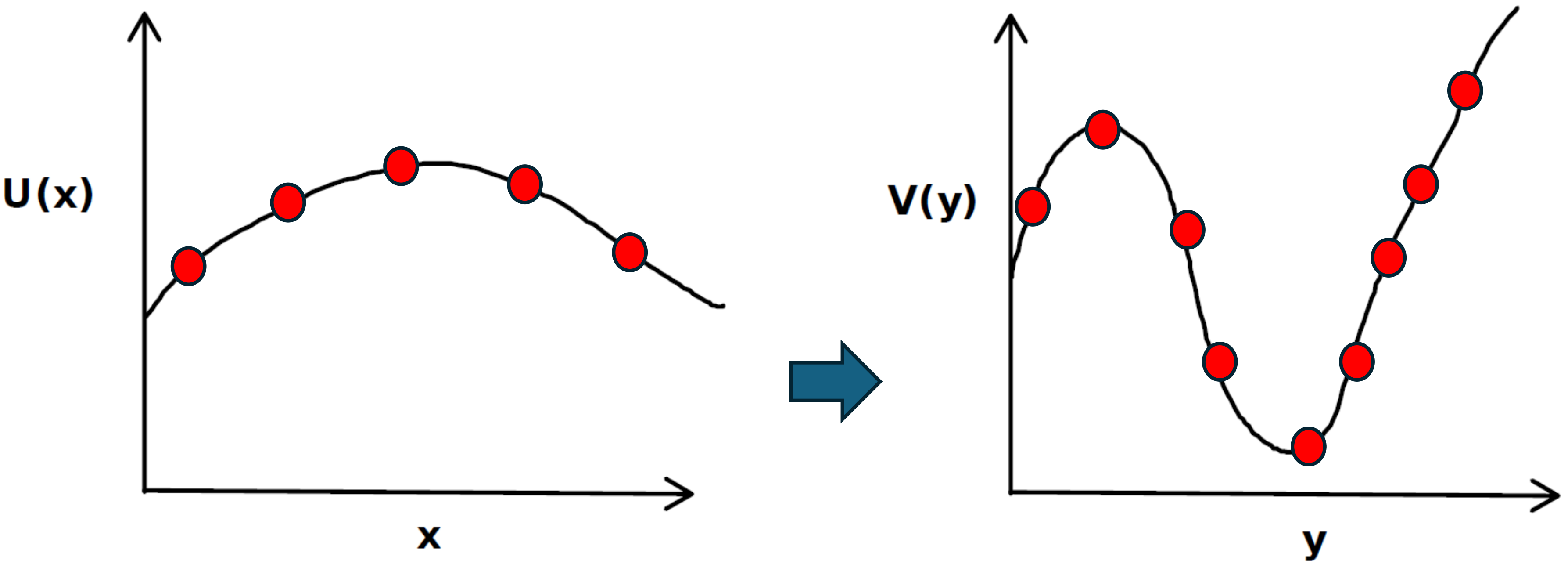
Single point in the domain where your output function exists (e.g. a location in 2D space)

What are they good for?



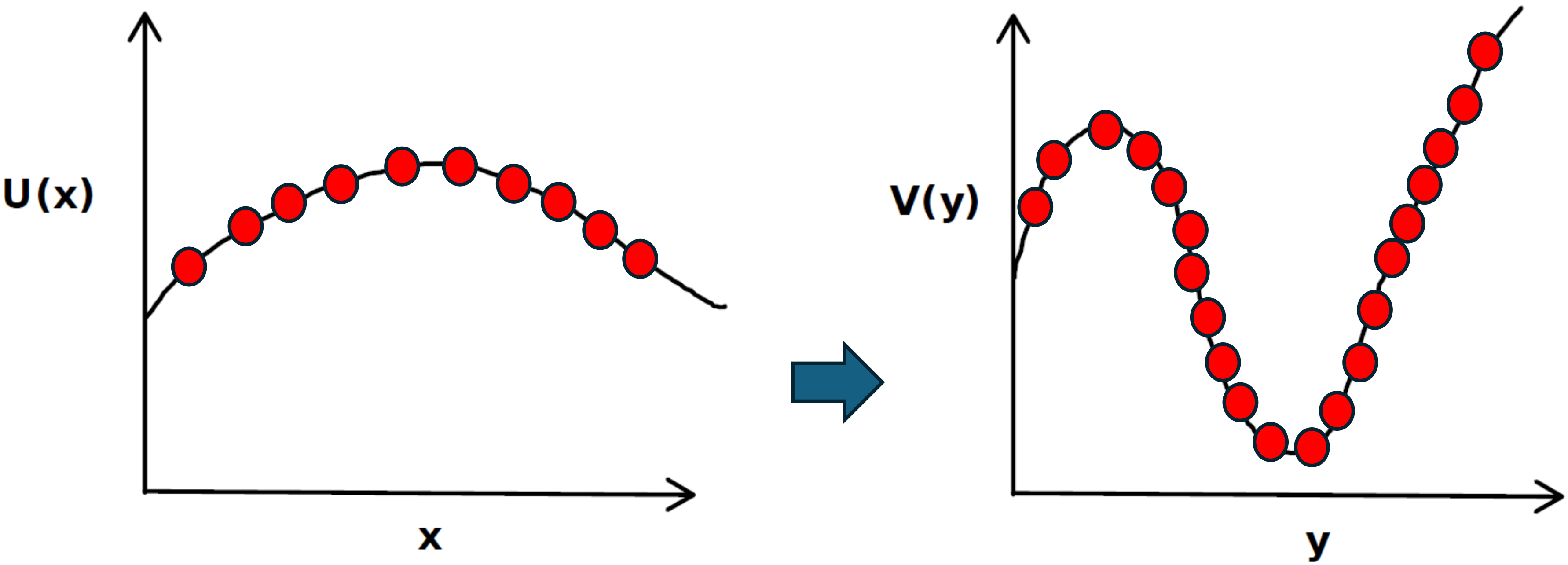
Mapping from continuous function space to continuous function space (...supposedly)

What are they good for?



Reality requires some discretization

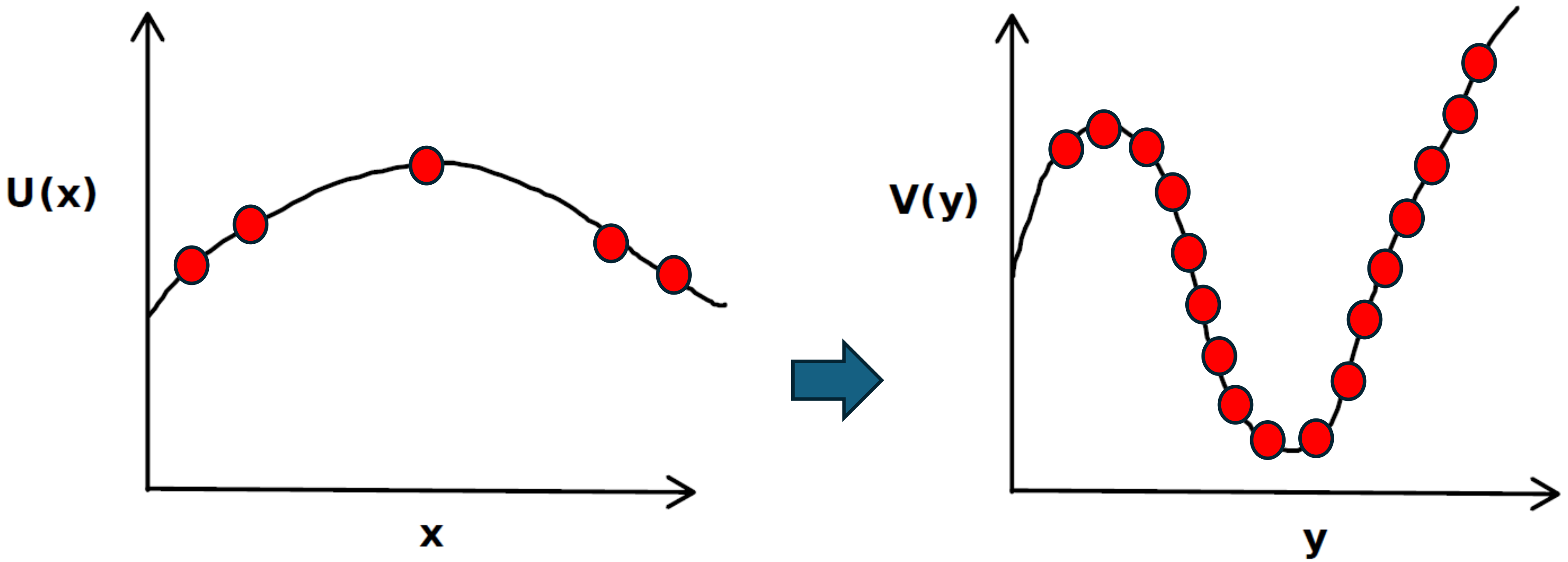
What are they good for?



Neural operators trained at lower resolution can often “perform well at higher resolution”

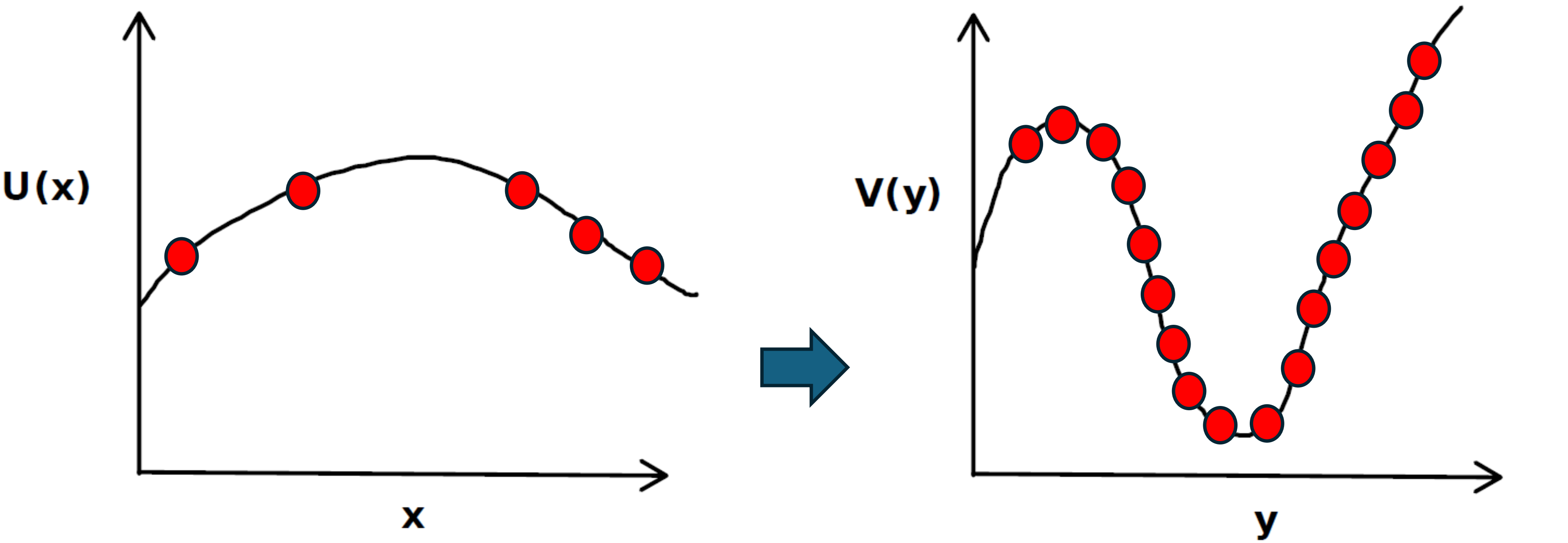
Property of “Super-resolution”

What are they good for?



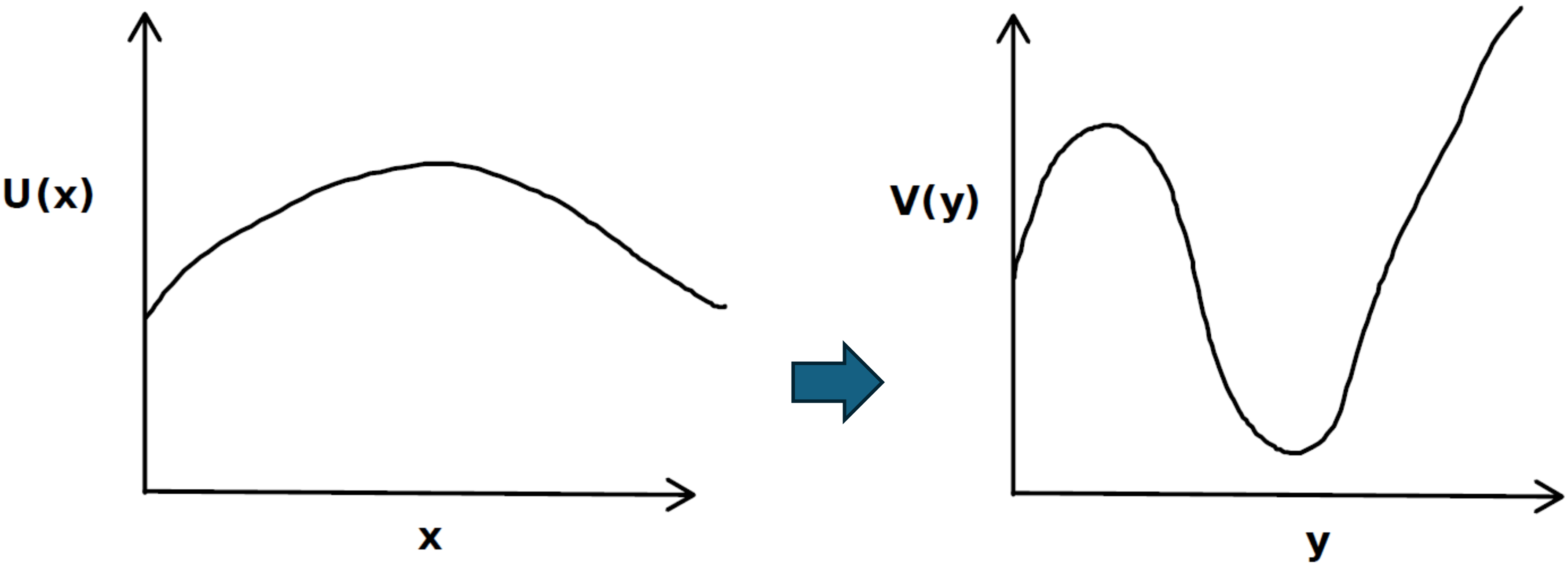
Some neural operators can handle different sampling of input domain

What are they good for?



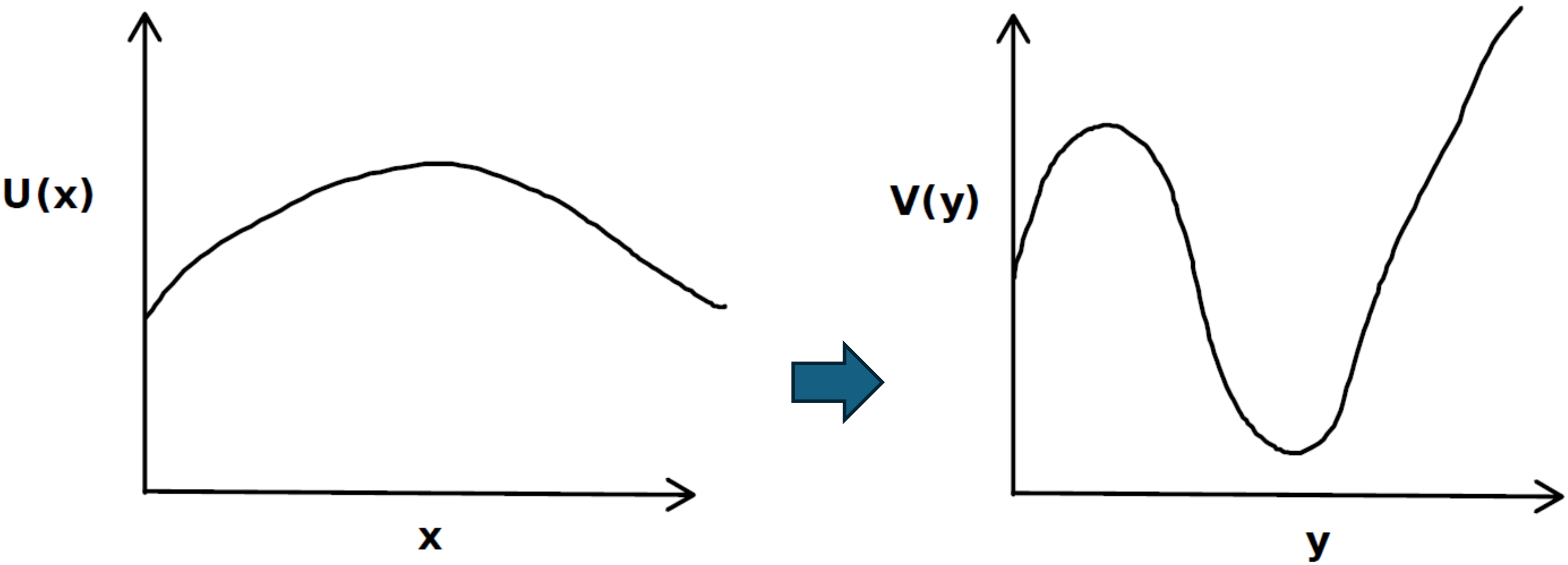
Some neural operators can handle different sampling of input domain

What are they good for?



Mapping between function spaces makes them ideal for applications where PDEs would typically need to be solved with numerical methods. Surrogates for numerical models. Types of problems that are pervasive in physics and engineering.

What are they good for?

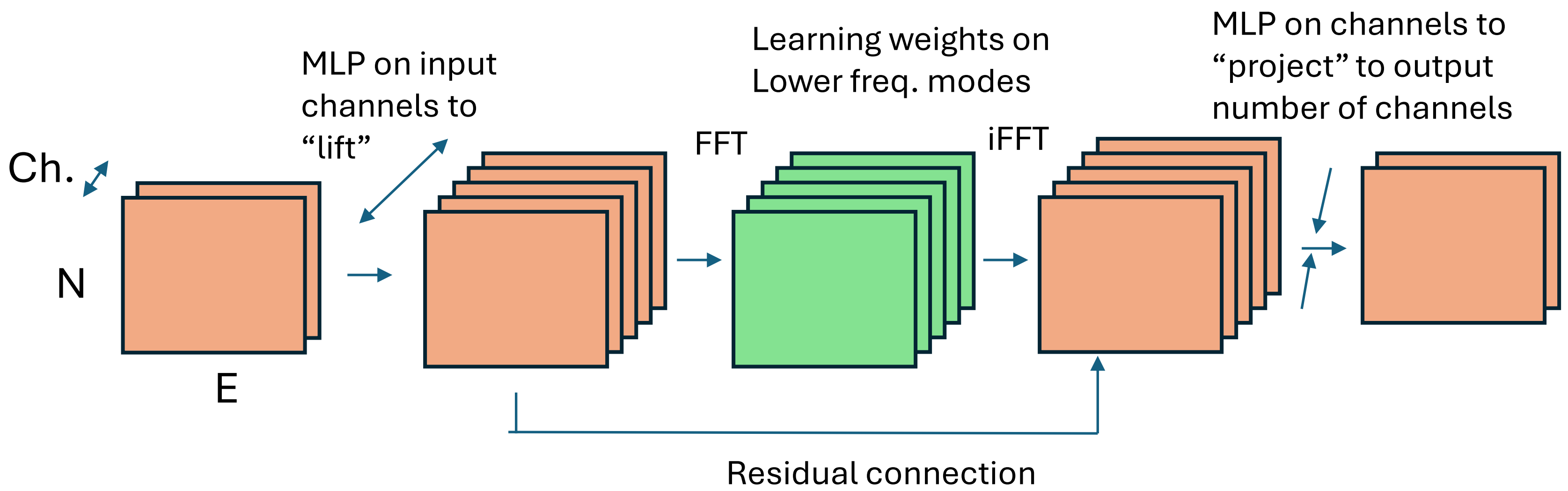


Mapping between function spaces makes them ideal for applications where PDEs would typically need to be solved with numerical methods. Surrogates for numerical models. Types of problems that are pervasive in physics and engineering.

Neural operators

- What is a neural operator?
- Neural operator architectures
- Literature examples of NO applications
- Applications in seismology and related research
- Discussion

Fourier Neural Operator



Comments:

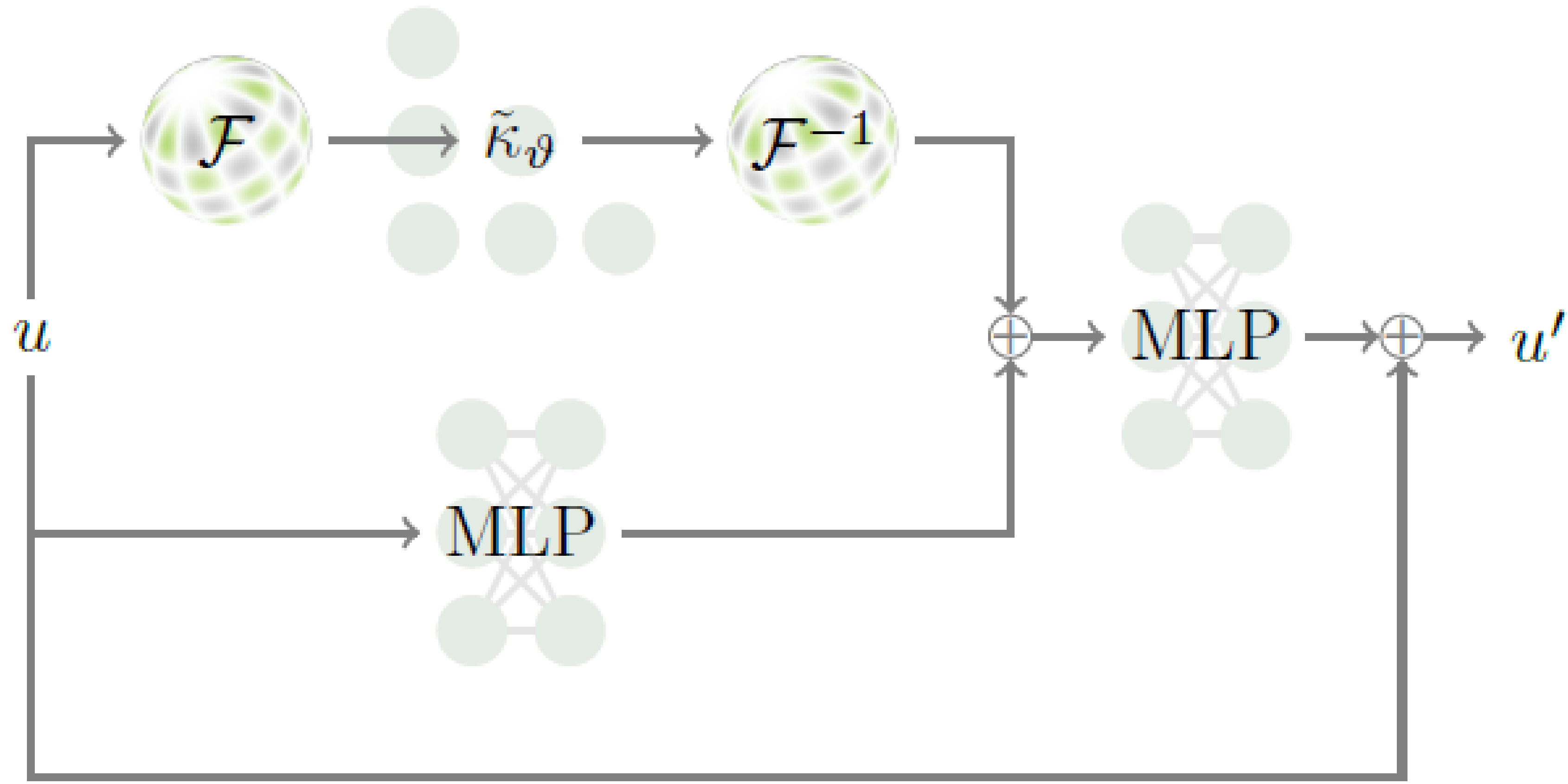
- Agnostic to the input dimension; train on lo-res, inference on hi-res (“zero-shot super-resolution”)

- No branch, no trunk… but still an operator network…

- ..the necessary convolution in the original domain is replaced by multiplication in frequency domain which leads to much faster application of the operator across all samples

[Fig. from J.Bedford, description of FNO model of Li et al. 2021 ICLR]

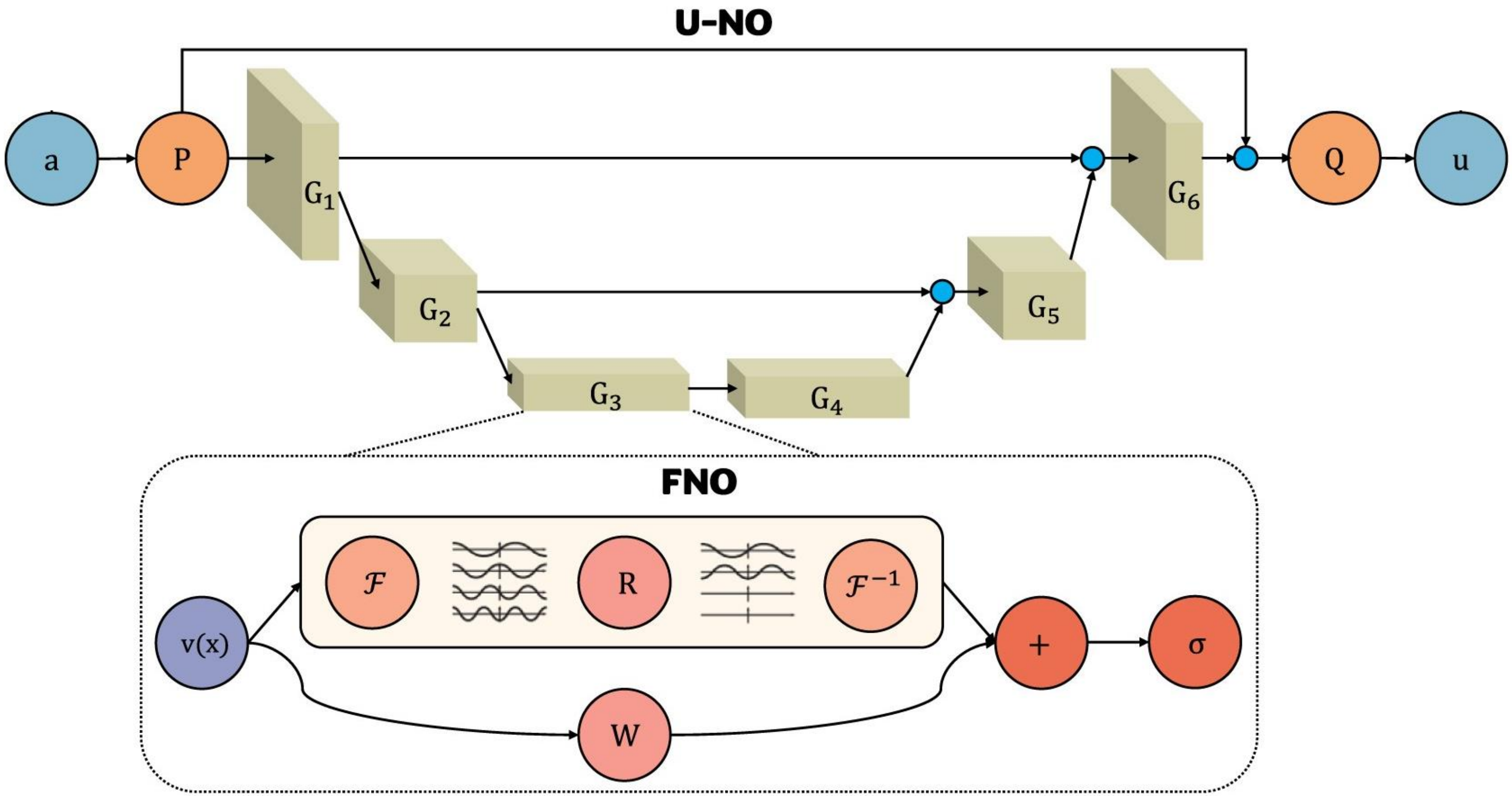
Fourier Neural Operator variants



SFNO (**S**pherical)

[Fig. from Bonev et al. 2023, PMLR]

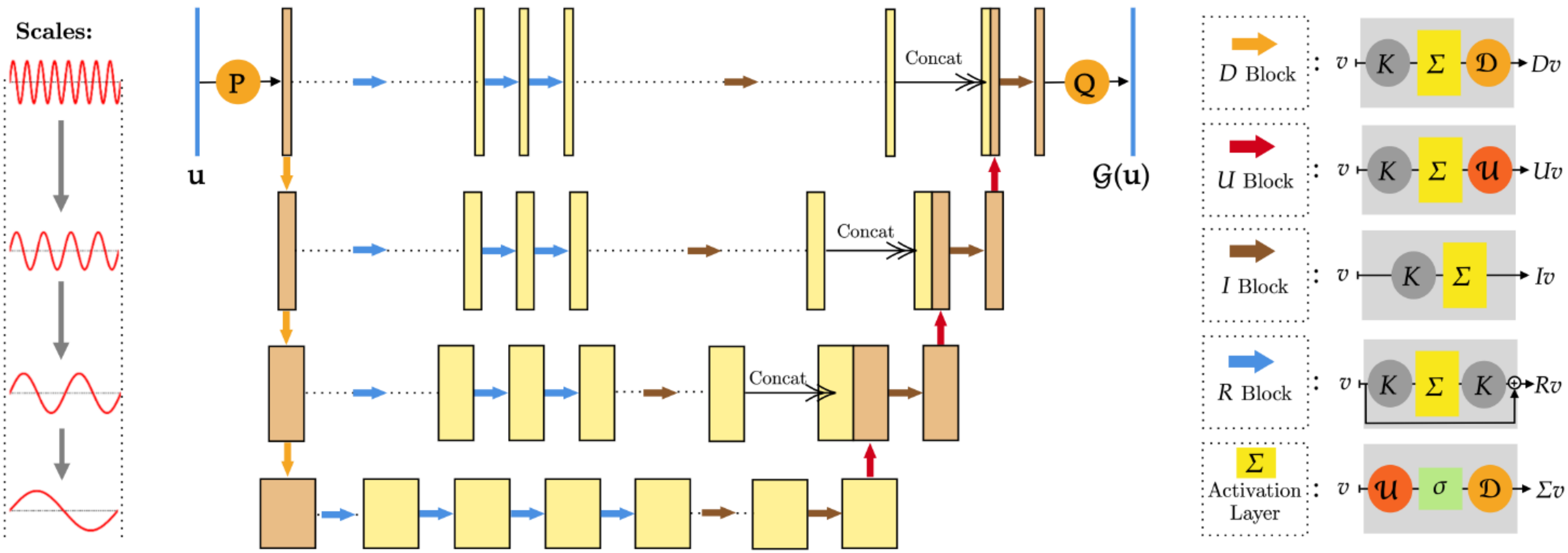
Fourier Neural Operator variants



UFNO (**U** shaped)

[Fig. from Zou et al. 2024 Geophysical Journal International]

Convolutional Neural Operator



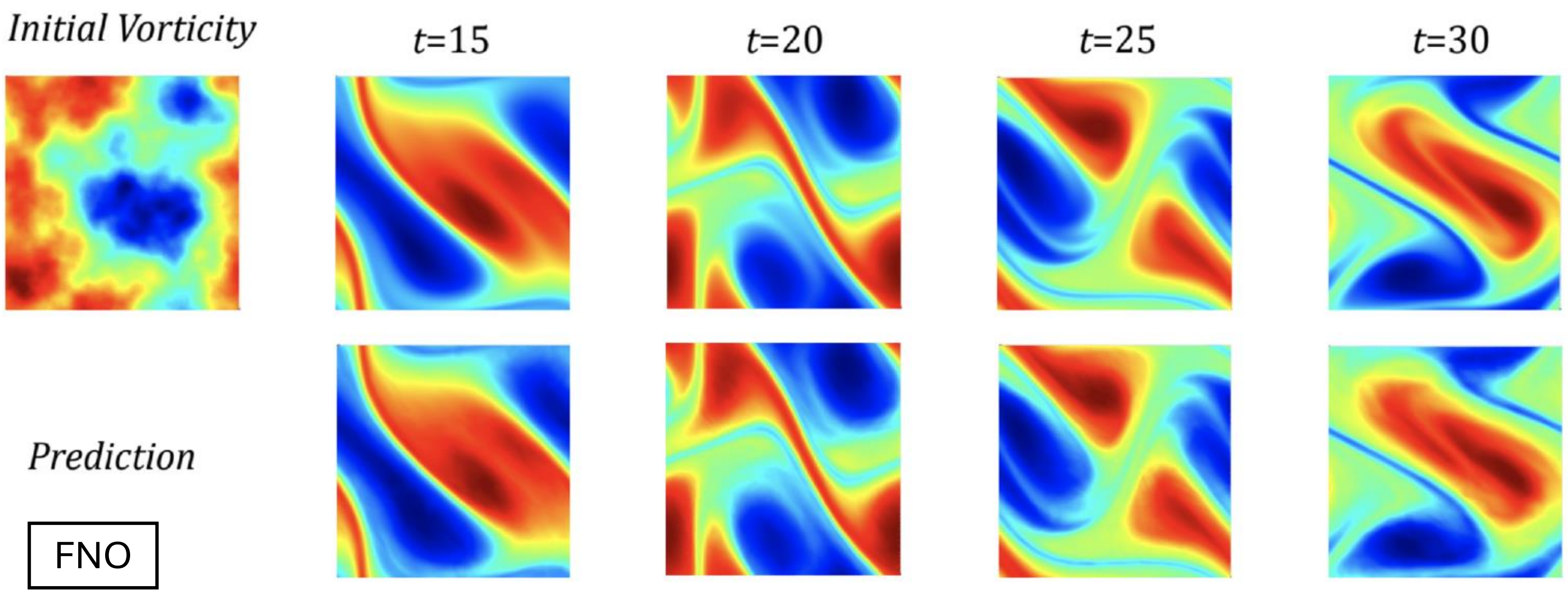
CNO (**C**onvolutional)

[Fig. from Raonic et al. 2023]

Neural operators

- What is a neural operator?
- Neural operator architectures
- Literature examples of NO applications
- Applications in seismology and related research
- Discussion

Surrogate numerical simulations

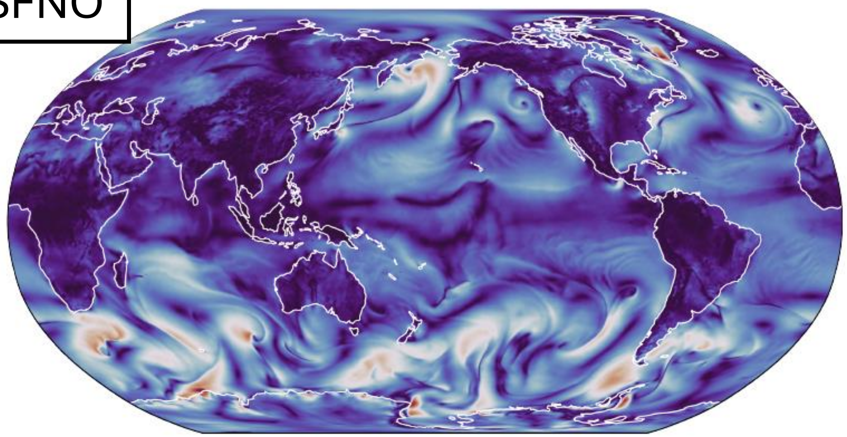


- Navier-Stokes fluid vorticity dataset created with pseudospectral numerical method
- FNO 3-D Trained on (64,64,20); evaluated on (256,256,80) -> Super resolution
- Over 200 times faster than numerical model simulation

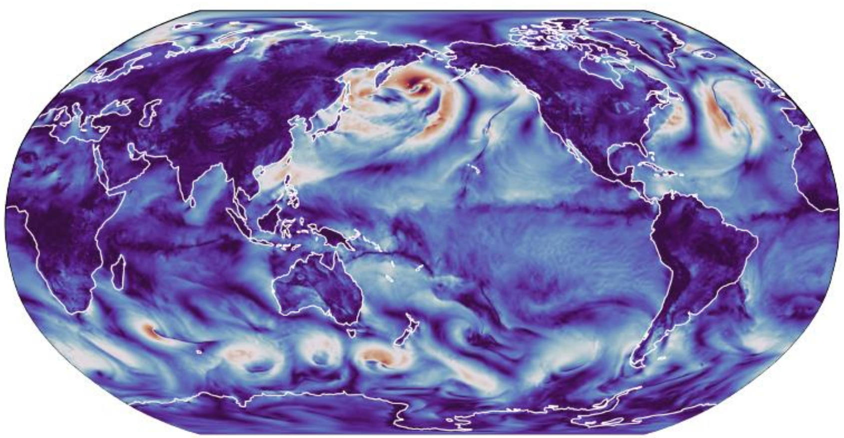
[Fig. from Li et al. 2021, ICLR]

Surrogate numerical simulations

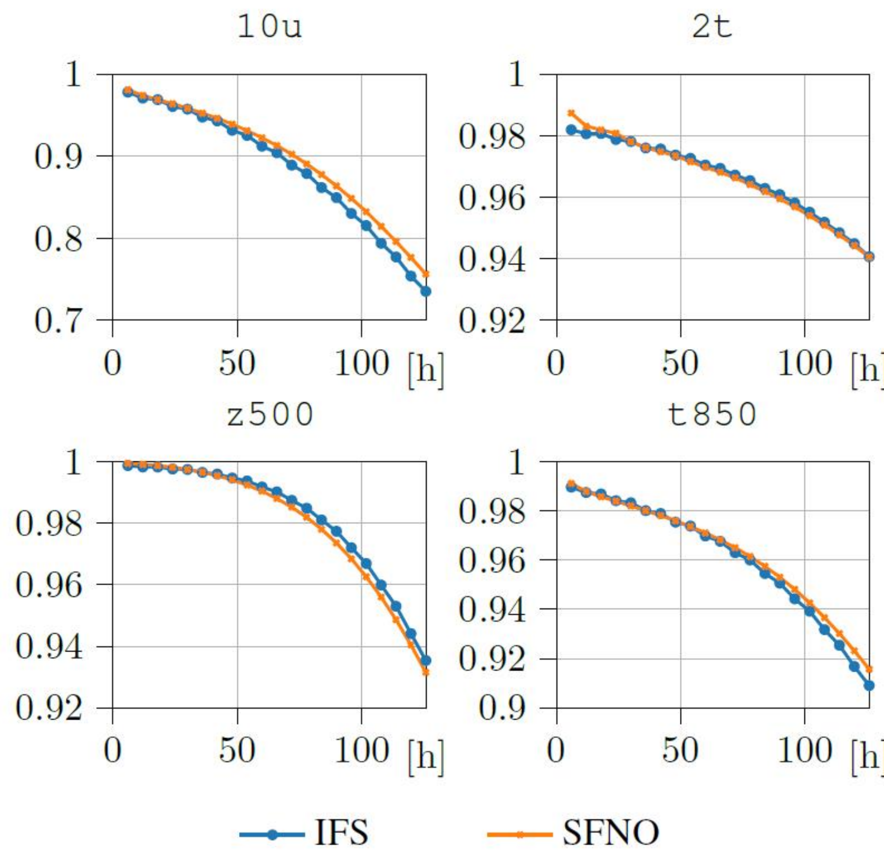
SFNO



(a) Spherical FNO using the SHT



(b) Ground Truth



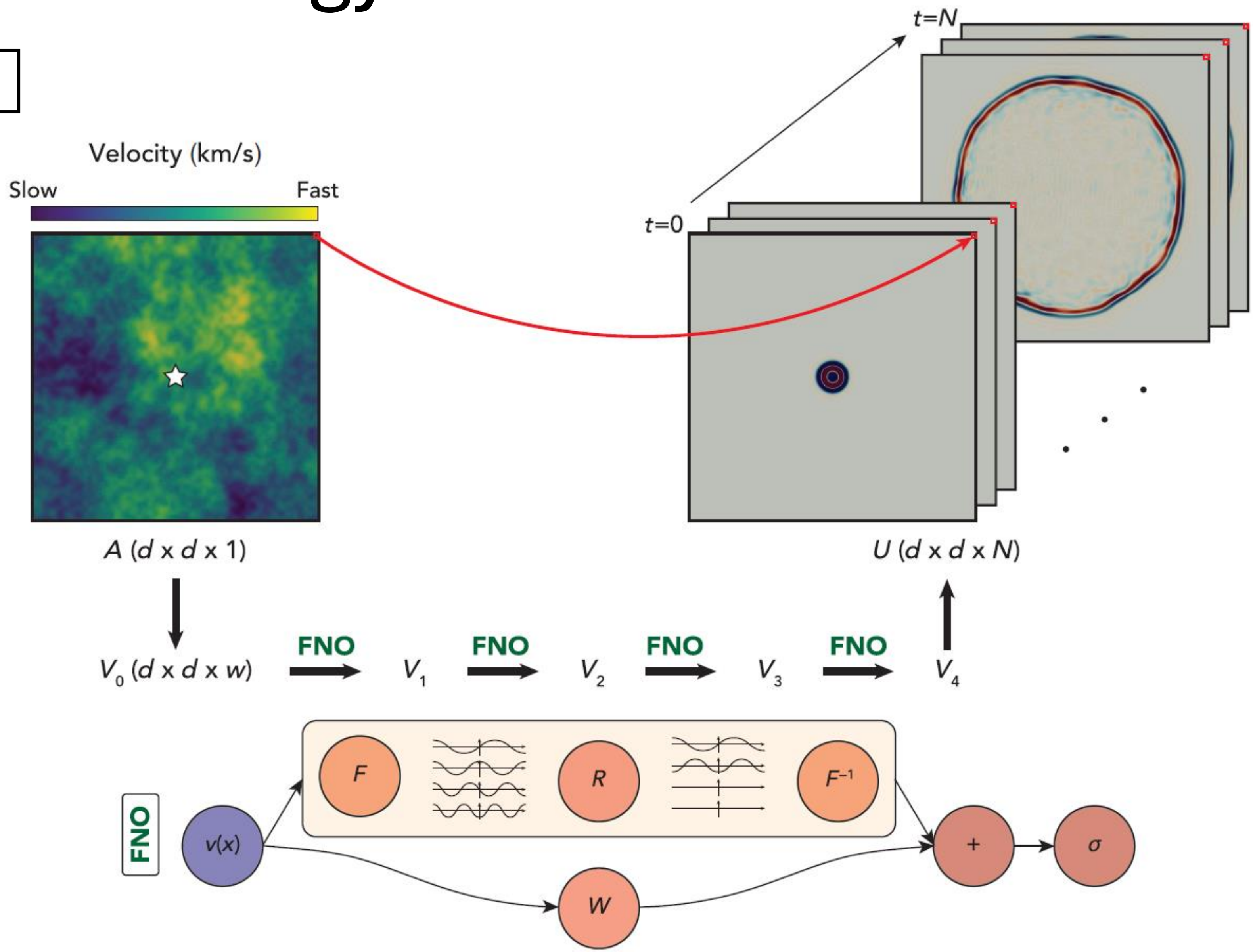
Forecasting skill of SFNO compared to numerical weather model (IFS).

5 year long rollout (1450 autoregressive steps)
Stability.

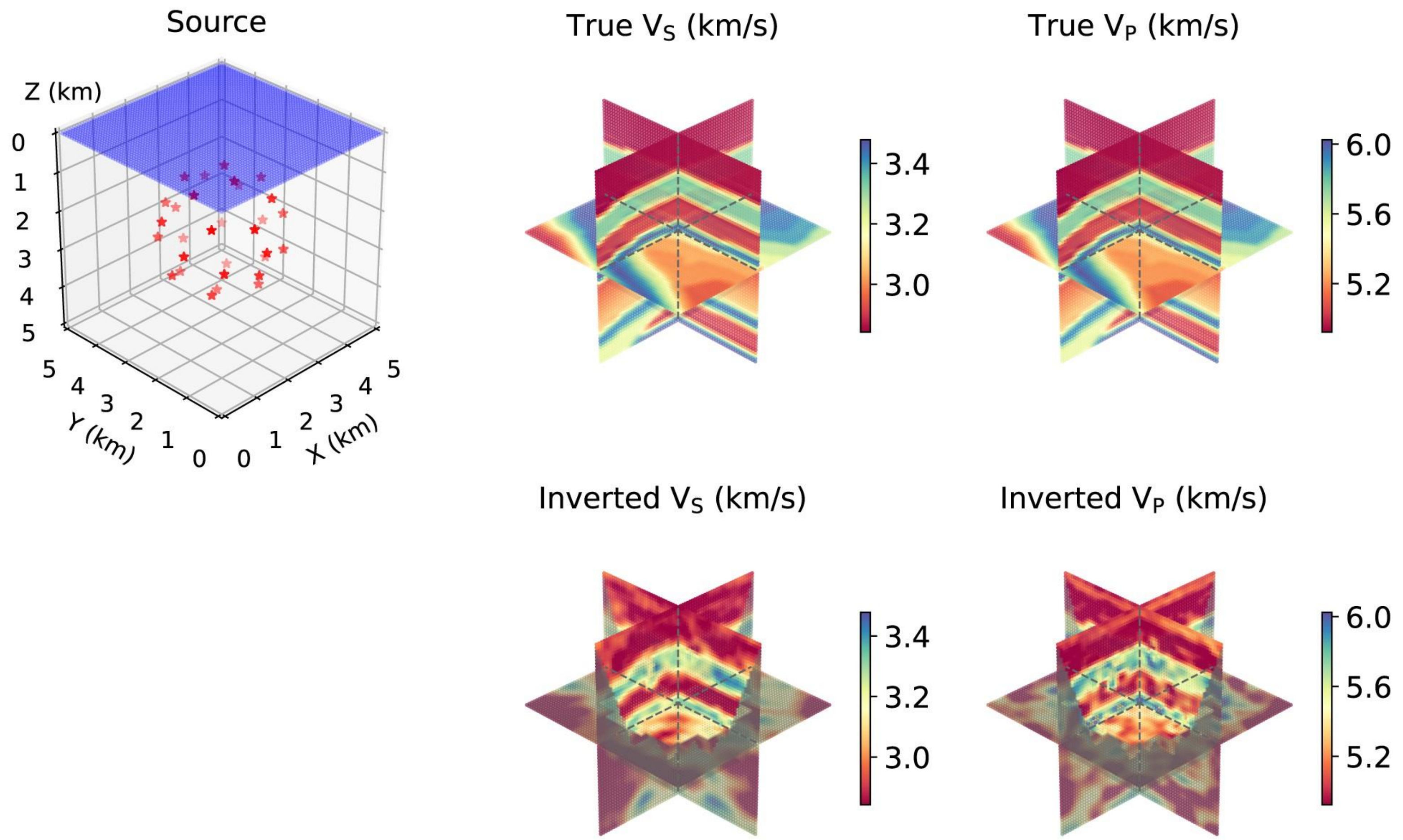
[Fig. from Bonev et al. 2023, PMLR]

In seismology

FNO



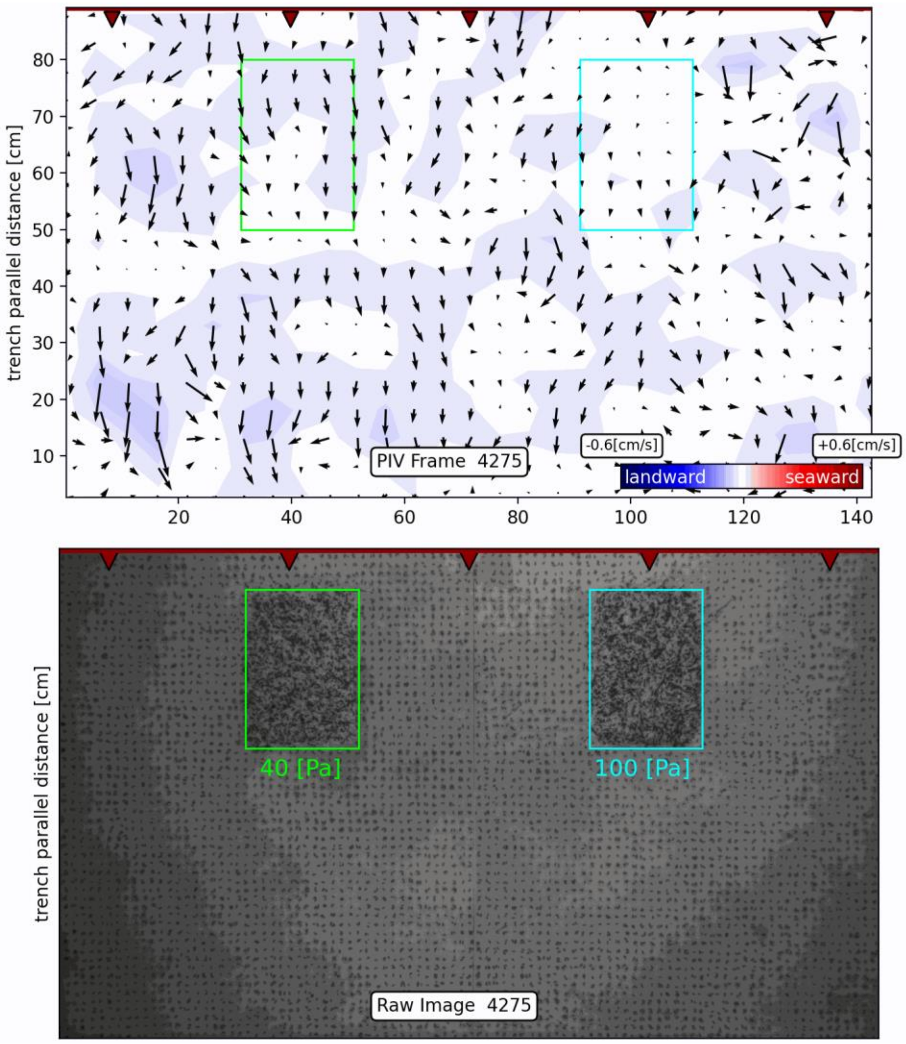
[Yang et al. The Seismic Record, 2021]



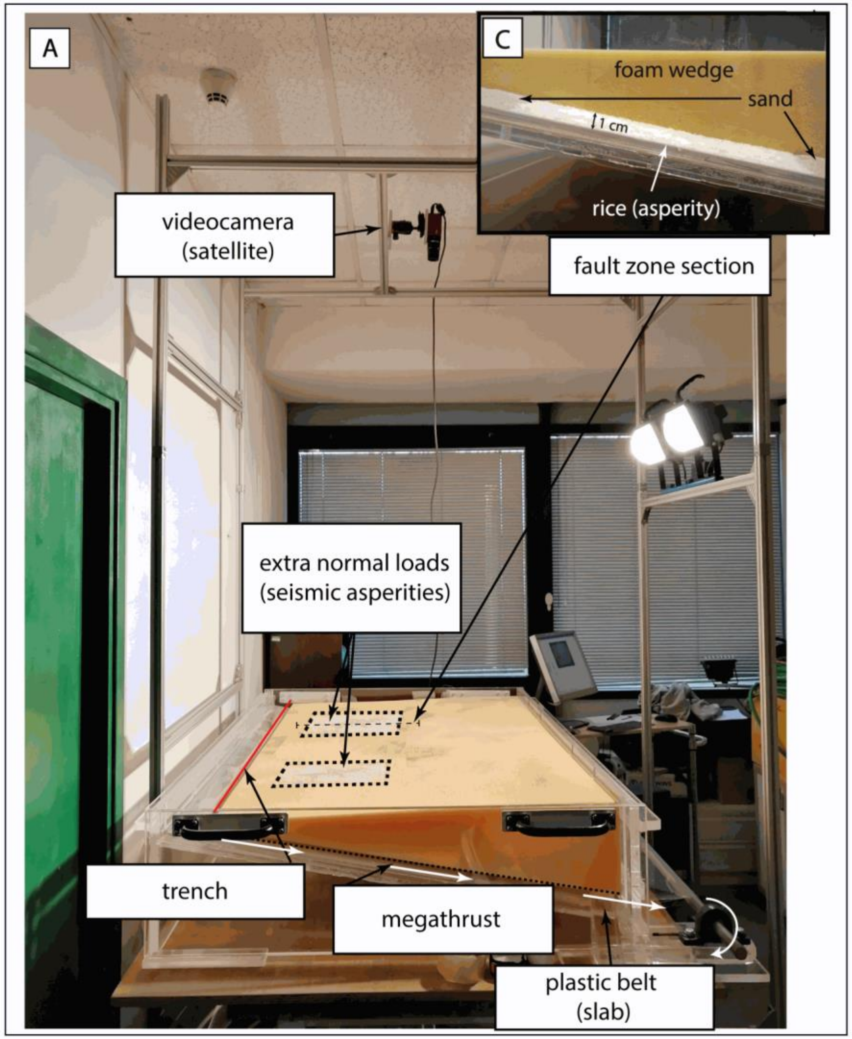
Models also being used in inversions (here the UFNO)

[Zou et al. 2024; Geophysical Journal International]

Autoregression for Laboratory earthquakes

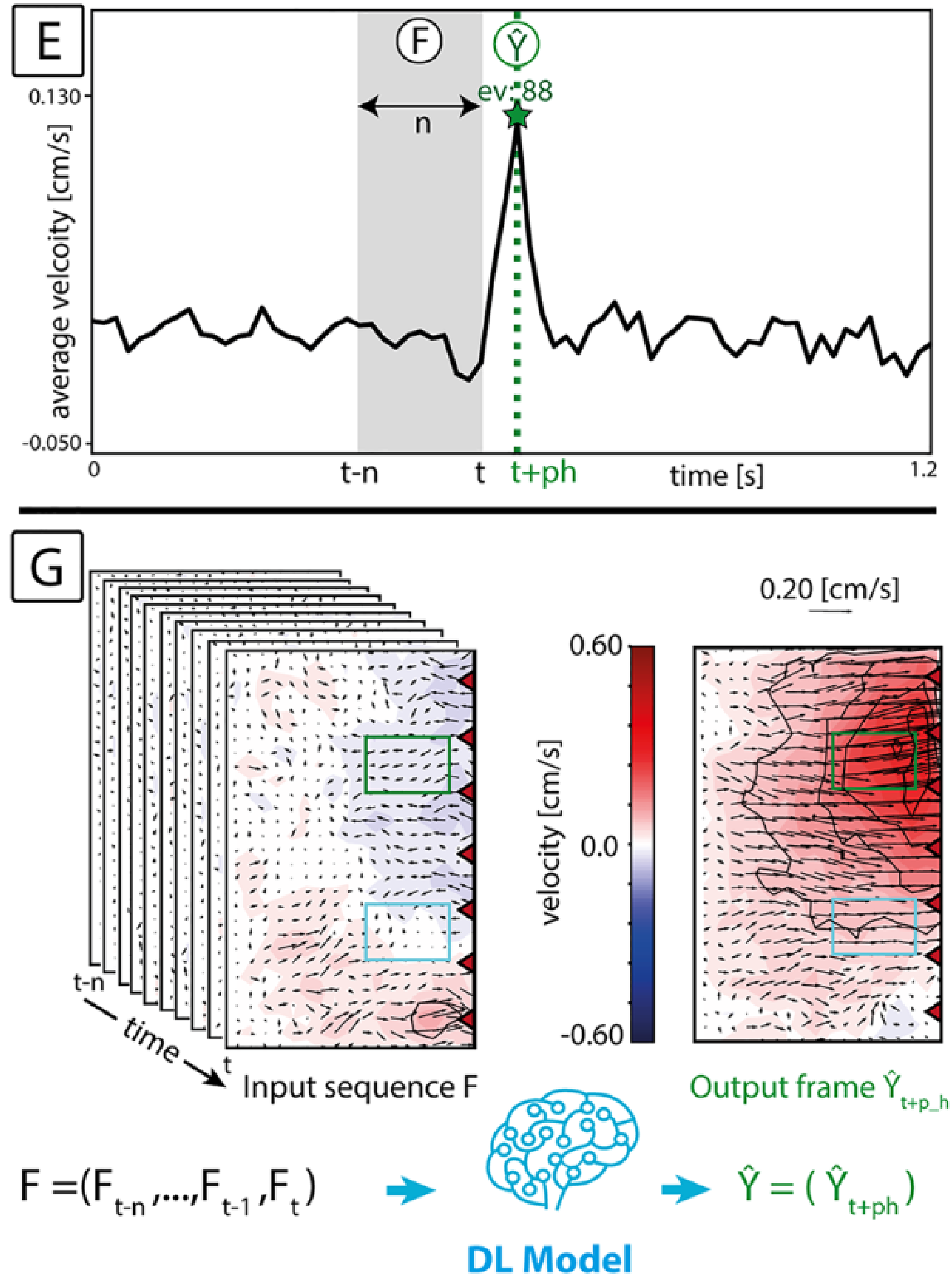


[Video courtesy of Giacomo Mastella]

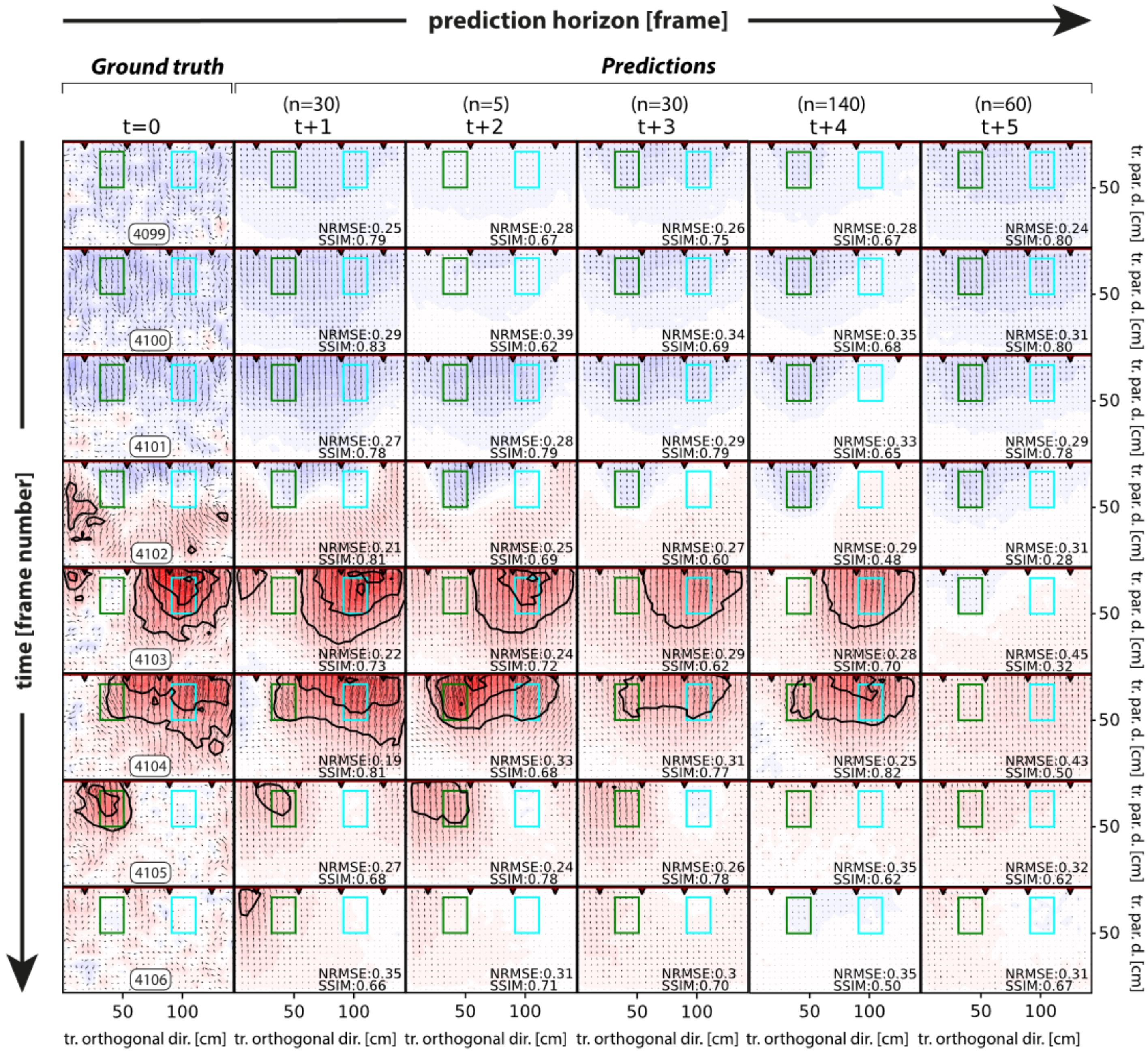


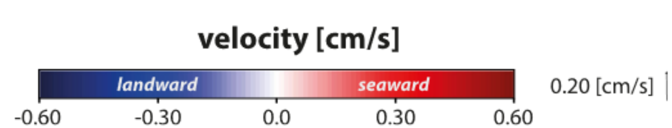
[Foamquake; Mastella et al. JGR 2022]

Enhanced forecasting



[Mastella et al., Geophysical Research Letters, 2022]



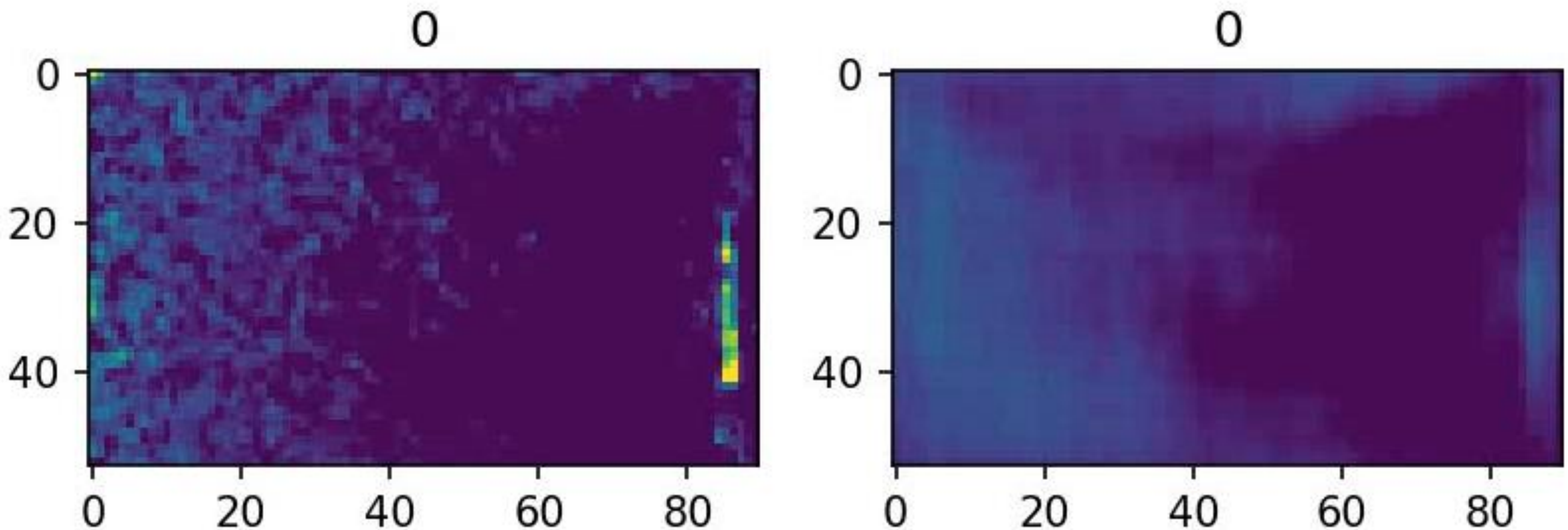


(convLSTM)

[Mastella et al., Geophysical Research Letters, 2022]

Data

Forecast (with prediction horizon of 10)



A “toy” FNO-2D model trained on some data of Van Rijsingen et al. [GRL 2019]

Plan to scale this up to the experiments of Elvira Latypova (PhD student at RUB).

Neural operators

- What is a neural operator?
- Neural operator architectures
- Literature examples of NO applications
- Applications in seismology and related research
- Discussion

Do we need neural operators?

	In/Out	FFNN	GT	UNet	ResNet	DON	FNO	CNO
Poisson Equation	In	5.74%	2.77%	0.71%	0.43%	12.92%	4.98%	**0.21%**
	Out	5.35%	2.84%	1.27%	1.10%	9.15%	7.05%	**0.27%**
Wave Equation	In	2.51%	1,44%	1.51%	0.79%	2.26%	1.02%	**0.63%**
	Out	3.01%	1.79%	2.03%	1.36%	2.83%	1.77%	**1.17%**
Smooth Transport	In	7.09%	0.98%	0.49%	0.39%	1.14%	0.28%	**0.24%**
	Out	650.6%	875.4%	1.28%	0.96%	157.2%	3.90%	**0.46%**
Discontinuous Transport	In	13.0%	1.55%	1.31%	**1.01%**	5.78%	1.15%	**1.01%**
	Out	257.3%	22691.1%	1.35%	1.16%	117.1%	2.89%	**1.09%**
Allen-Cahn Equation	In	18.27%	0.77%	0.82%	1.40%	13.63%	**0.28%**	0.54%
	Out	46.93%	2.90%	2.18%	3.74%	19.86%	**1.10%**	2.23%
Navier-Stokes Equations	In	8.05%	4.14%	3.54%	3.69%	11.64%	3.57%	**2.76%**
	Out	16.12%	11.09%	10.93%	9.68%	15.05%	9.58%	**7.04%**
Darcy Flow	In	2.14%	0.86%	0.54%	0.42%	1.13%	0.80%	**0.38%**
	Out	2.23%	1.17%	0.64%	0.60%	1.61%	1.11%	**0.50%**
Compressible Euler	In	0.78%	2.09%	0.38%	1.70%	1.93%	0.44%	**0.35%**
	Out	1.34%	2.94%	0.76%	2.06%	2.88%	0.69%	**0.59%**

[Table. from CNO paper of Raonic et al. 2023]

“Representation equivalency”

Are Neural Operators Really Neural Operators? Frame Theory Meets Operator Learning

Francesca Bartolucci[1], Emmanuel de Bézenac[2], Bogdan Raonić[2,3], Roberto Molinaro[2], Siddhartha Mishra[2,3], and Rima Alaifari[2,3]

[1]Delft University of Technology
[2]Seminar for Applied Mathematics, ETH Zürich
[3]ETH AI Center

Is this really happening?

Continuous function space → another Continuous function space

Taking care about aliasing -> CNO has some solutions to this.

Getting started with operator learning

aai-institute / continuiti Public

Notifications | Fork 3 | Star 30

Code | Issues 8 | Pull requests 5 | Discussions | Actions | Security | Insights

main | 10 Branches | 2 Tags | Go to file | Code

Samuel Burbulla Merge pull request #156 from aai-institute/feature/support-ty... 74e1687 · 9 months ago 561 Commits

.github	Update PyPI publish workflow.	9 months ago
benchmarks	Remove files from lfs.	9 months ago
build_scripts	Merge branch 'main' into cleanup/restructure-docs	9 months ago
data	Remove files from lfs.	9 months ago
docs	Update README.	9 months ago
src/continuiti	add package typing marker	9 months ago
tests	add isolate deepxde during testing	9 months ago
.bumpversion.toml	Bump version: 0.2.0 → 0.2.1	9 months ago
.flake8	mv flake8 configuration to individual file	last year
.gitattributes	Add gitattributes.	9 months ago
.gitignore	Update gitignore.	last year

About

Learning function operators with neural networks.

physics-informed-ml transferlab neural-operators

- Readme
- LGPL-3.0 license
- Activity
- Custom properties
- 30 stars
- 2 watching
- 3 forks

Report repository

Releases 2

0.2.0 Latest
on Aug 20, 2024

+ 1 release

continuiti

Learning function operators with neural networks.

continuiti is a Python package for deep learning on function operators with a focus on elegance and generality. It provides a *unified interface* for neural operators (such as DeepONet or FNO) to be used in a plug and play fashion. As operator learning is particularly useful in scientific machine learning, **continuiti** also includes physics-informed loss functions and a collection of relevant benchmarks.

Tutorials

Getting started with continuiti

How-to Guides

Solve your problem

Background

More details on operator learning

Reference

API documentation

How-to Guides

Neural operators extend the concept of neural networks to function mappings, which enables discretization-invariant and mesh-free mappings of data with applications to physics-informed training, super-resolution, and more.

This is a collection of notebooks that showcase some applications of **continuiti** and serve as a guide to solve specific problems.

Time Series

Operator learning for non-uniform time series

Super-resolution

Neural operators for super-resolution

Physics-informed

Training physics-informed neural operators

Meshes

Reading meshes for operator learning

Self-supervised

Training self-supervised neural operators

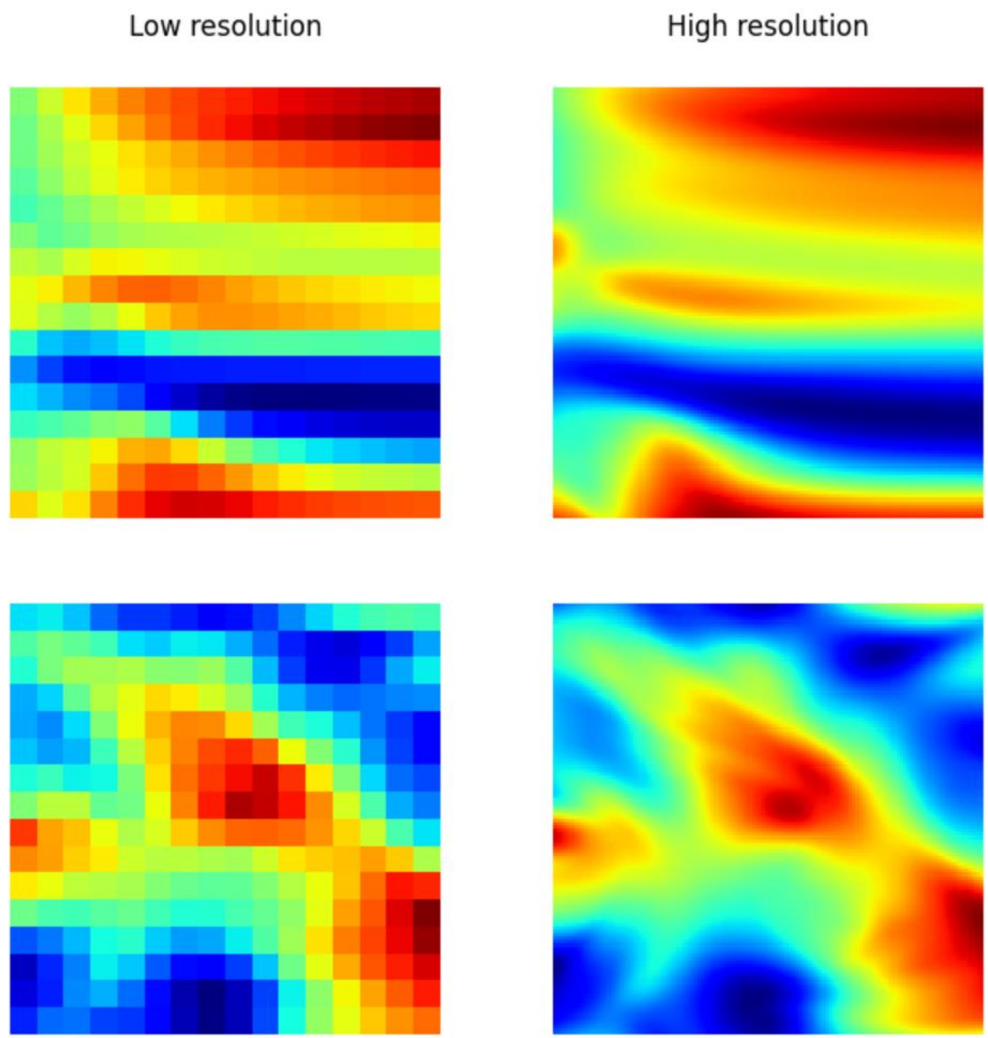


Table of contents

Neural Operators in PyTorch

NeuralOperator
Learning in infinite dimensions

`neuraloperator` is a comprehensive library for learning neural operators in PyTorch. It is the official implementation for Fourier Neural Operators and Tensorized Neural Operators.

Unlike regular neural networks, neural operators enable learning mapping between function spaces, and this library provides all of the tools to do so on your own data.

NeuralOperators are also resolution invariant, so your trained operator can be applied on data of any resolution.

Quickstart

This guide will walk you through the standard ML workflow of loading data, creating a neural operator, training it on the data and saving the trained model for later use. (Check out Examples for more info)

First install the library `pip install neuraloperator` (see Installing NeuralOperator for more options).

To create a Fourier Neural Operator model:

```
from neuralop.models import FNO
```

Computational and Applied Mathematics Laboratory @ ETH Zurich

We are part of the Seminar for Applied Mathematics at the Department of Mathematics.

113 followers · Switzerland · https://camlab.ethz.ch

Overview · Repositories 10 · Projects · Packages · People 3

Popular repositories

AI_Science_Engineering Public
This repository is the official project page of the course AI in the Sciences and Engineering, ETH Zurich.
Jupyter Notebook · 205 · 44

ConvolutionalNeuralOperator Public
This repository is the official implementation of the paper Convolutional Neural Operators for robust and accurate learning of PDEs
Python · 168 · 19

poseidon Public
Code for the paper "Poseidon: Efficient Foundation Models for PDEs"
Python · 138 · 24

GenCFD Public
Code for the paper "Generative AI for fast and accurate statistical computation of fluids"
Python · 27 · 6

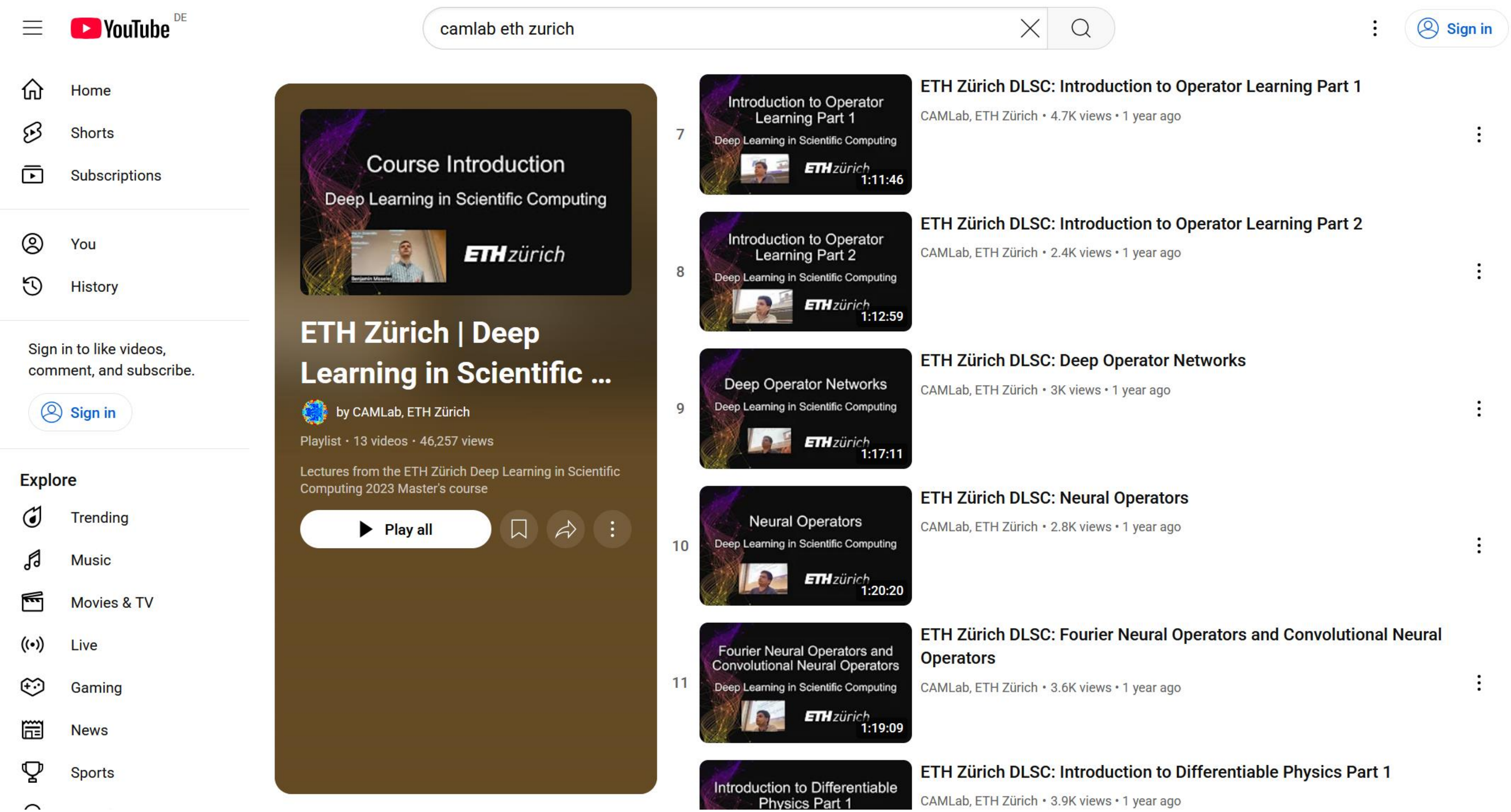
GEMS Public
Protein-Ligand Binding Affinity Prediction with GNN and Transfer Learning From Protein Language Models
Python · 25 · 8

rigno Public
Graph-based operator learning in arbitrary geometries
Python · 23 · 3

People

Top languages

Python · Jupyter Notebook


YouTube DE
camlab eth zurich
Sign in
Home
Shorts
Subscriptions
You
History
Sign in to like videos, comment, and subscribe.
Sign in
Explore
Trending
Music
Movies & TV
Live
Gaming
News
Sports
Course Introduction
Deep Learning in Scientific Computing
ETH zürich
ETH Zürich | Deep Learning in Scientific ...
by CAMLab, ETH Zürich
Playlist • 13 videos • 46,257 views
Lectures from the ETH Zürich Deep Learning in Scientific Computing 2023 Master's course
Play all
7
Introduction to Operator Learning Part 1
Deep Learning in Scientific Computing
1:11:46
ETH Zürich DLSC: Introduction to Operator Learning Part 1
CAMLab, ETH Zürich • 4.7K views • 1 year ago
8
Introduction to Operator Learning Part 2
Deep Learning in Scientific Computing
1:12:59
ETH Zürich DLSC: Introduction to Operator Learning Part 2
CAMLab, ETH Zürich • 2.4K views • 1 year ago
9
Deep Operator Networks
Deep Learning in Scientific Computing
1:17:11
ETH Zürich DLSC: Deep Operator Networks
CAMLab, ETH Zürich • 3K views • 1 year ago
10
Neural Operators
Deep Learning in Scientific Computing
1:20:20
ETH Zürich DLSC: Neural Operators
CAMLab, ETH Zürich • 2.8K views • 1 year ago
11
Fourier Neural Operators and Convolutional Neural Operators
Deep Learning in Scientific Computing
1:19:09
ETH Zürich DLSC: Fourier Neural Operators and Convolutional Neural Operators
CAMLab, ETH Zürich • 3.6K views • 1 year ago
Introduction to Differentiable Physics Part 1
ETH Zürich DLSC: Introduction to Differentiable Physics Part 1
CAMLab, ETH Zürich • 3.9K views • 1 year ago

README Apache-2.0 license

📝 NVIDIA Modulus has been renamed to NVIDIA PhysicsNeMo

repo status Active | license Apache-2.0 | code style black

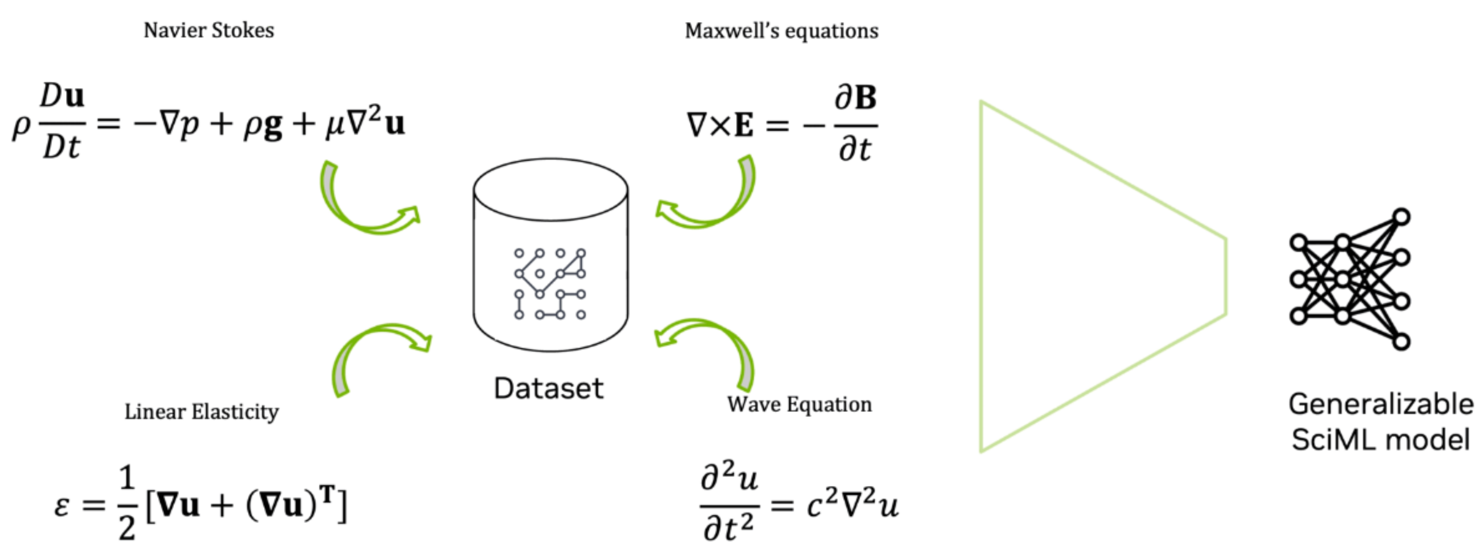
Getting Started | Install guide | Contributing Guidelines | Resources | PhysicsNeMo Migration Guide | Communication | License

What is PhysicsNeMo?

NVIDIA PhysicsNeMo is an open-source deep-learning framework for building, training, and fine-tuning deep learning models using state-of-the-art SciML methods for AI4science and engineering.

PhysicsNeMo provides utilities and optimized pipelines to develop AI models that combine physics knowledge with data, enabling real-time predictions.

Whether you are exploring the use of Neural operators, GNNs, or transformers or are interested in Physics-informed Neural Networks or a hybrid approach in between, PhysicsNeMo provides you with an optimized stack that will enable you to train your models at scale.



NeuralOperators

`NeuralOperators.jl` is a package written in Julia to provide the architectures for learning mapping between function spaces, and learning grid invariant solution of PDEs.

Installation

On Julia 1.10+, you can install `NeuralOperators.jl` by running

```
import Pkg
Pkg.add("NeuralOperators")
```

Currently provided operator architectures are :

- Fourier Neural Operators (FNOs)
- DeepONets
- Nonlinear Manifold Decoders for Operator Learning (NOMADs)

Reproducibility

▸ The documentation of this SciML package was built using these direct dependencies,

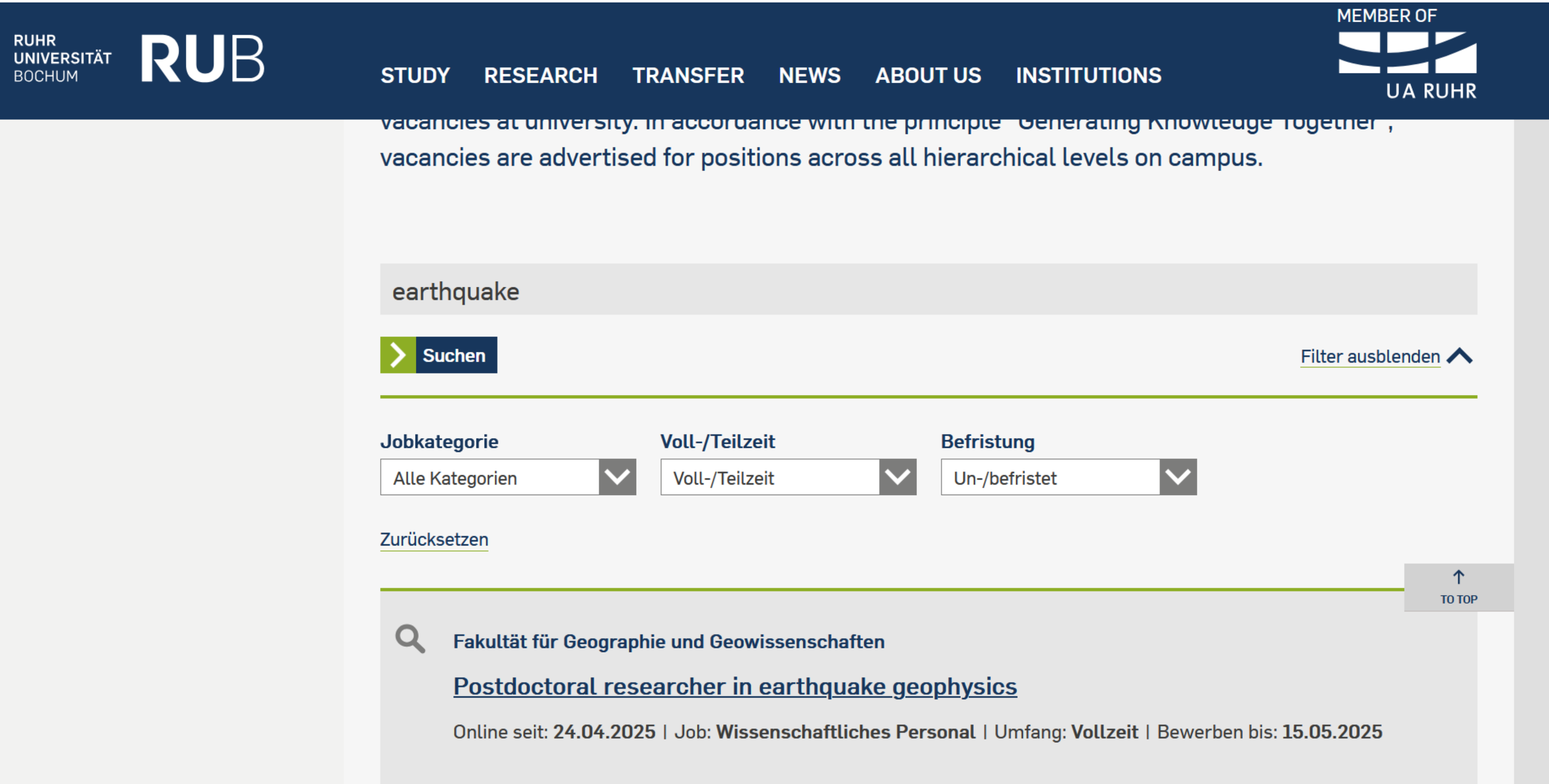
▸ and using this machine and Julia version.

▸ A more complete overview of all dependencies and their versions is also provided.

You can also download the manifest file and the project file.

JOB ADVERTISEMENT

https://uni.ruhr-uni-bochum.de/en/vacancies



Applying SciML; PINNs; Neural Operators, on problems related to earthquake faulting and the seismic cycle

https://jobs.ruhr-uni-bochum.de/jobposting/02d6f3fe552e0347881bb33e1377e7c4fa433ce30

Questions?