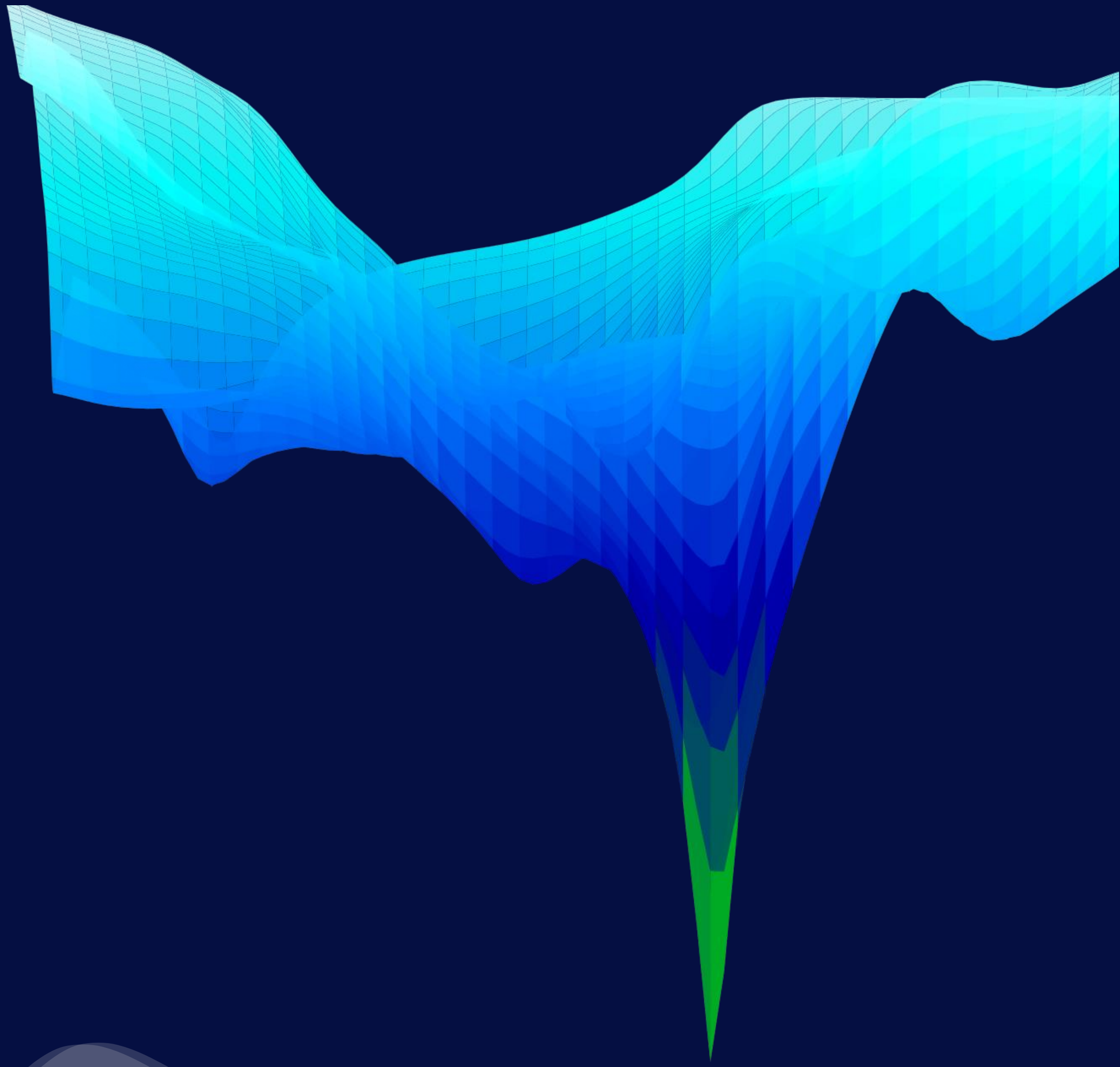
Scientific Machine Learning (SciML) for Solving PDEs

(Case Studies: two-phase flow in porous media)



University of Campinas (Nov. 2024)

A part of INTPART project, supported by NCS2030 - National Centre for Sustainable Subsurface Utilization of the Norwegian Continental Shelf.

NCS2030 National Centre for Sustainable Subsurface Utilization of the Norwegian Continental Shelf
University of Stavanger

University of Stavanger

JASSEM ABBASI

https://jcabbasi.github.io/

Scientific Machine Learning & Reservoir Engineering

ABOUT ME

Dedicated Research Scientist specializing in computational modeling and data analysis with over five years of experience in both academic and industry research. Expertise in Scientific Machine Learning, numerical simulations, and fluid dynamics, with a focus on subsurface flow.

QUALIFICATIONS

Programming
Scientific Software Development
Machine Learning | Deep Learning
Optimization | Inverse Calculations
Applied Mathematics | Statistics
Data Science | Data Analytics
Reservoir Engineering/Simulation
Computational Fluid Dynamics (CFD)
Flow in Porous Media
Thermodynamics

CURRENT ACTIVITY

Application of Physics-informed Machine Learning for Modelling of Multiphase Flow Processes in Porous Media

EXPERIENCE

ETH Zürich (2024)
ETH AI Center – Visiting Researcher

EQUINOR ASA, Norway (2022) [intern]
Subsurface Geoscience/Reservoir Simulation Engineer

ZODAN SOLUTIONS LTD., UK (2019-2020)
Scientific Software Developer

SHIRAZ UNIVERSITY/PETROAZMA COMPANY (2016-2018)
Reservoir [Simulation] Engineer/Researcher

PETROTIRAZIS OIL COMPANY PTED. (2016) [intern]
Scientific Software Developer (Intern)

EDUCATION

UNIVERSITY OF STAVANGER (2021- Dec. 2024)
Scientific Machine Learning (PhD)

SHIRAZ UNIVERSITY (2014-2016)
Reservoir Engineering (M.Sc.)

PETROLEUM UNIVERSITY OF TECHNOLOGY (2010-2014)
Reservoir Engineering (B.Sc.)

PUBLICATIONS (selected)

ML4PS @ NeurIPS (2024): History-Matching of Imbibition Flow in Multiscale Fractured Porous Media Using Physics-Informed Neural Networks (PINNs)

SPE Journal (2024): Application of Physics-Informed Neural Networks for Estimation of Saturation Functions from Counter current Spontaneous Imbibition Tests →

Neurocomputing (2024): Physical Activation Functions (PAFs): An Approach for More Efficient Induction of Physics into Physics-Informed Neural Networks (PINNs) →

Energy and Fuels (2023): Simulation and Prediction of Spontaneous Imbibition at Early and Late Times Using Physics-Informed Neural Networks →

...

Contents

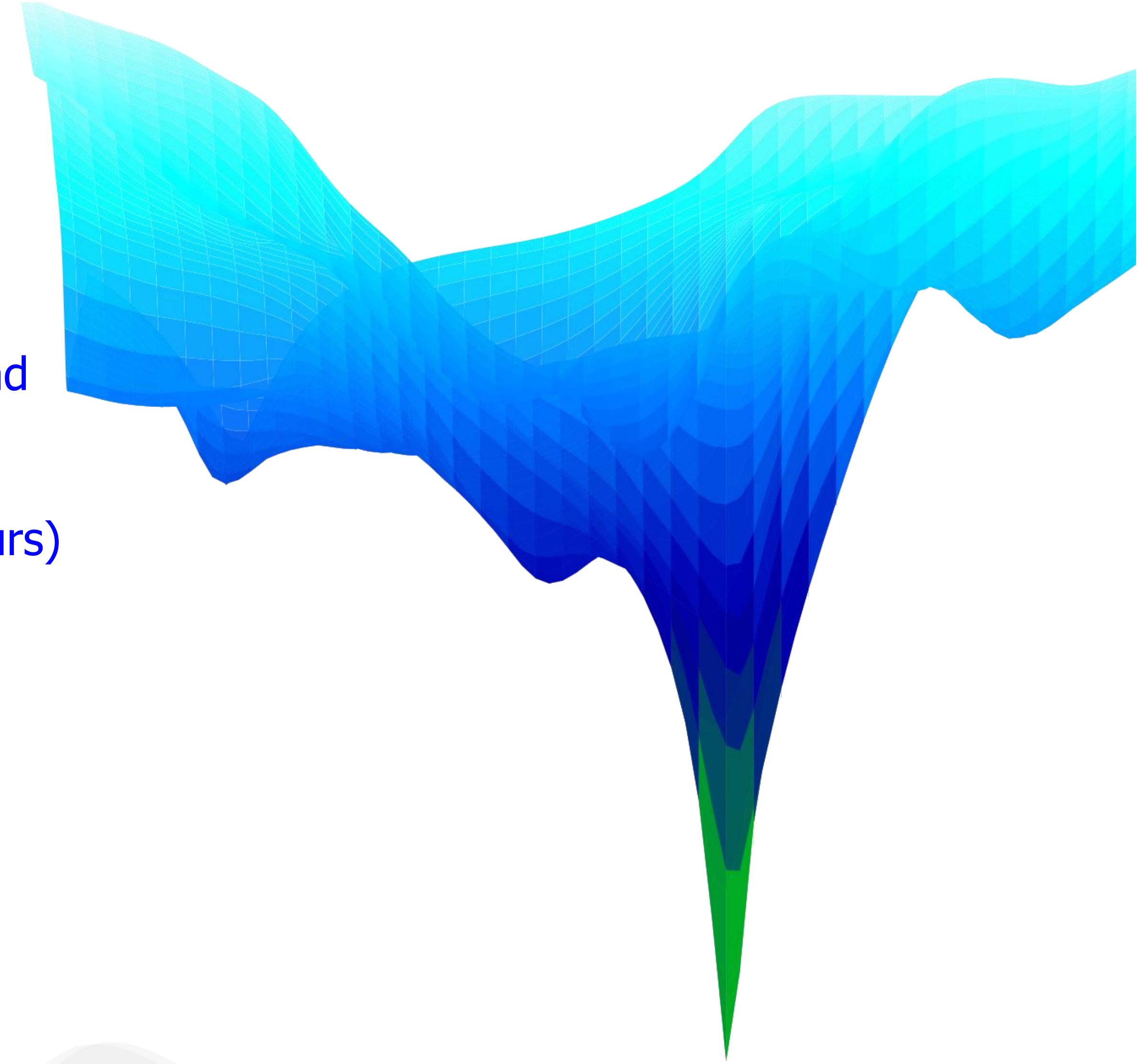
Day 1

Part 1: Scientific Machine Learning (SciML): Introduction and Recent Advances (1 hour)

Part 2: Physics-Informed Neural Networks (PINNs) (1.5 hours)

Day 2

Part 3: Hands-on Experience with PINNs (2.5 hours)



Key Take-aways

- Understand what SciML is?
- Recognize the limitations of traditional numerical methods for PDEs, and why SciML can help.
- Identify the unique advantages SciML offers in handling complex PDE challenges.
- Limitations and future directions of SciML in PDEs
- See real-world examples where SciML improved PDE-solving outcomes

References

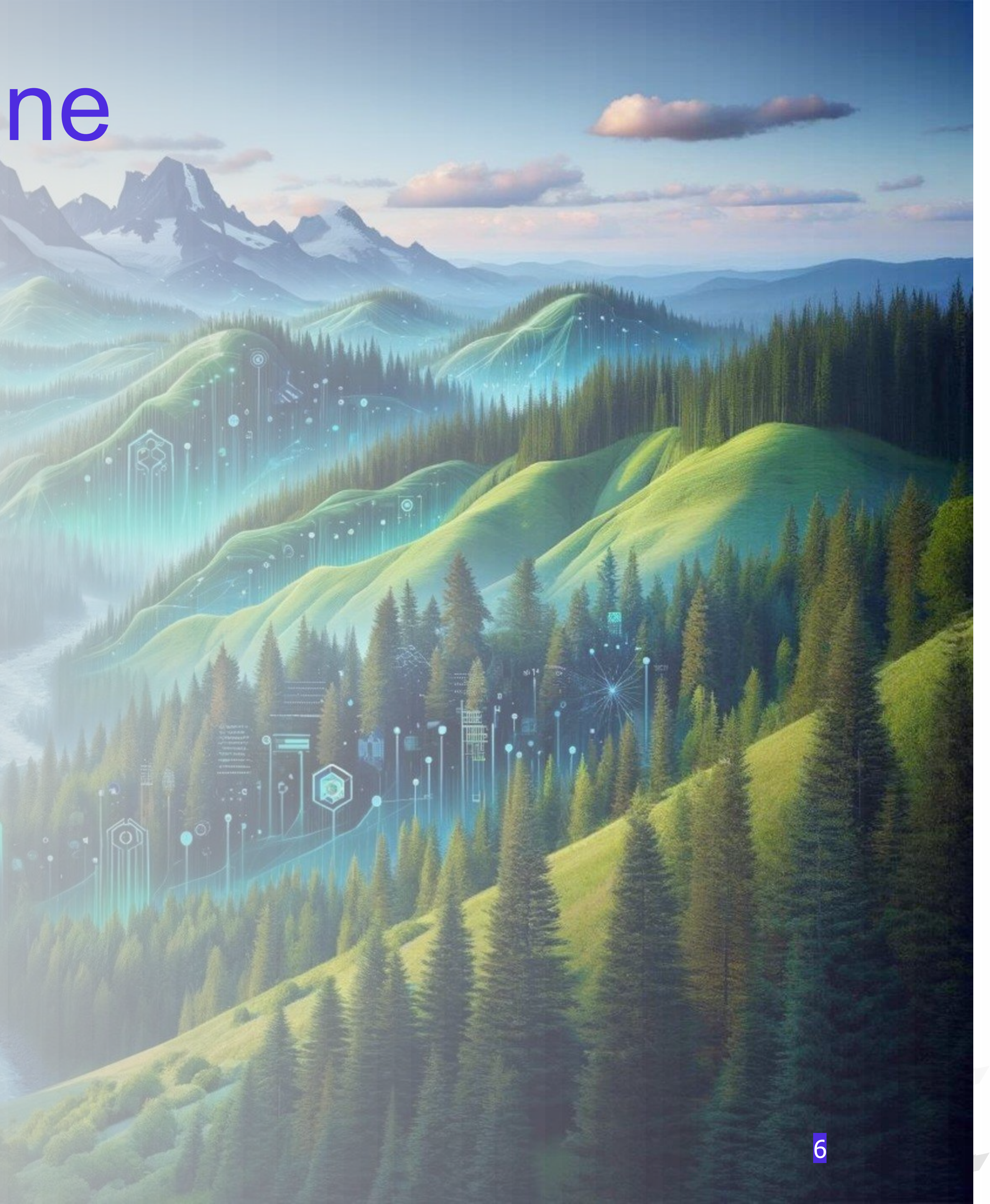
- Deep Learning in Scientific Computing, ETH Zurich, CAMLab

https://www.youtube.com/playlist?list=PLJkYEExhe7rYY5HjpIJbgo-tDZ3bIAqAm

- Deep Learning for Scientists and Engineers, by George Em Karniadakis, Khemraj Shukla, Summer 2023 at KTH Stockholm
- Steve Brunton YouTube channel. https://www.youtube.com/@Eigensteve
- https://sciml.tamids.tamu.edu/ecen-689-scientific-machine-learning-spring-2024/
- …

What is Scientific Machine Learning (SciML)?

- Scientific Machine Learning (SciML) is an interdisciplinary field that combines principles of scientific computing, applied mathematics, data science, and machine learning to model, predict, and analyze complex systems governed by the laws of physics, chemistry, biology, or engineering.
- Main Characteristics: Use Domain Knowledge



#AI4Science

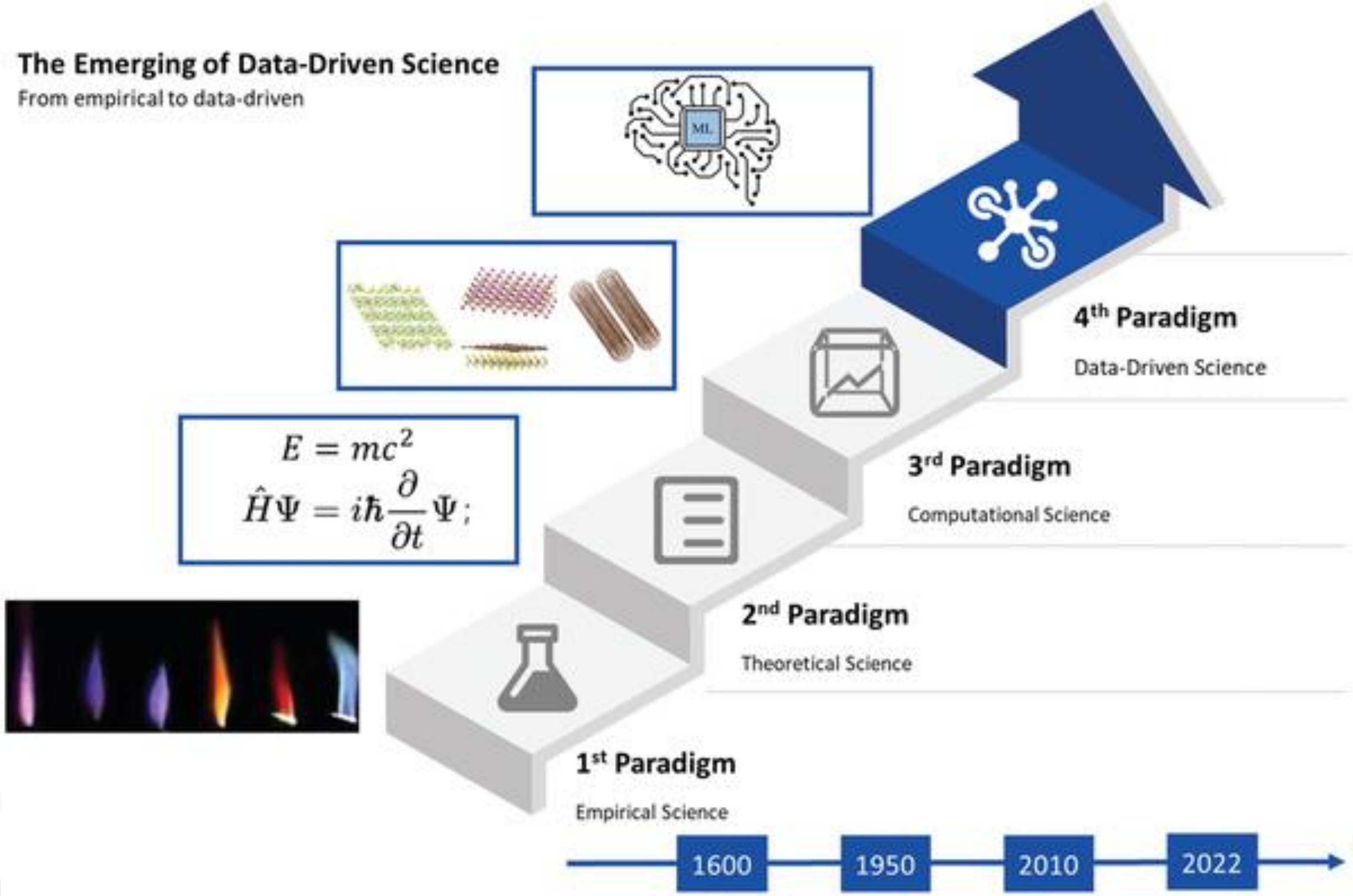
The Nobel Prize in Physics & Chemistry

Trained neural networks using physics:
- John Hopfield (Princeton University)
- Geoffrey Hinton (University of Toronto)

For computational protein design
- David Baker (University of Washington)
- Demis Hassabis (Google DeepMind)
- John M. Jumper (Google DeepMind)

Paradigms of Science

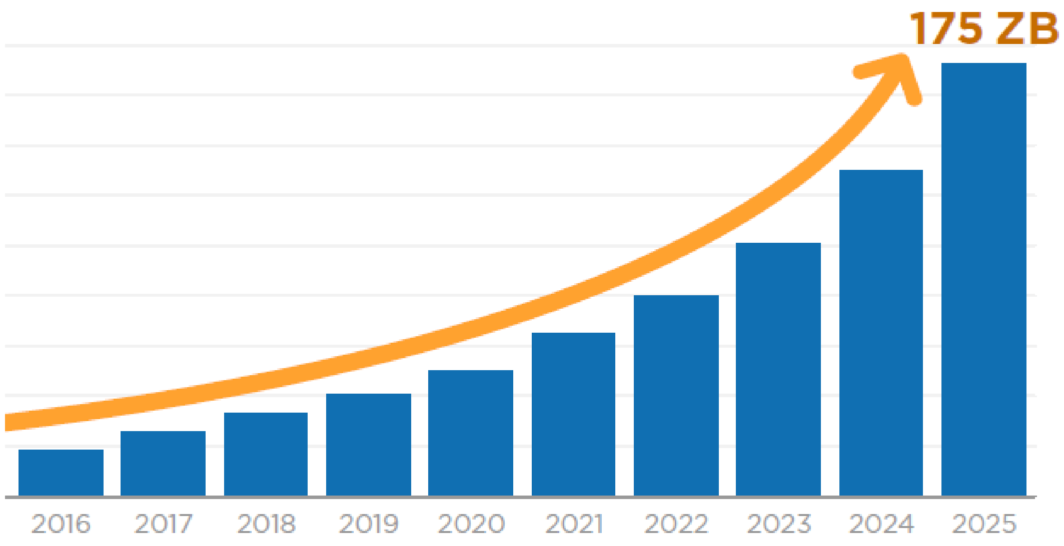
What is happening?



Data-Driven Materials Innovation and Applications, 2022

Why now?

Volume of Data Generated Annually



Training Compute

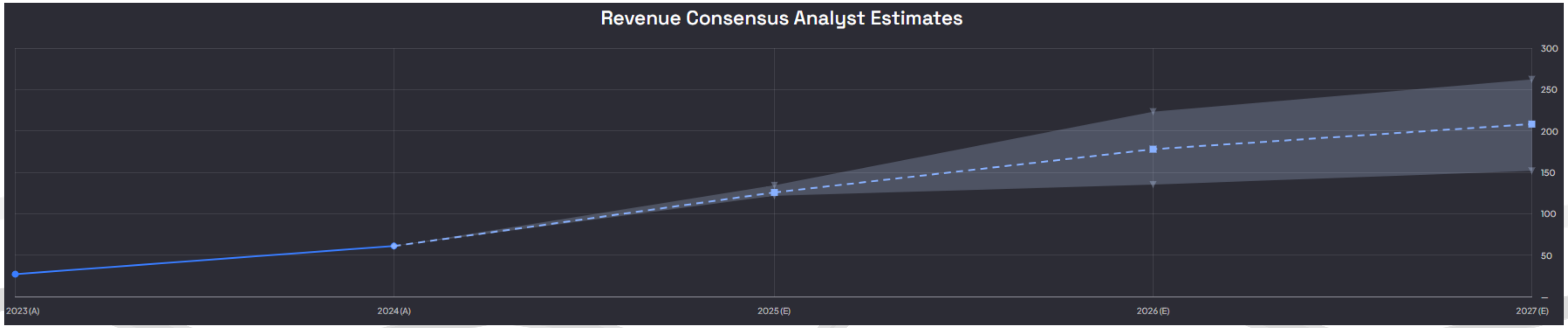
Larger Datasets

Better Hardware

Smarter Algorithms

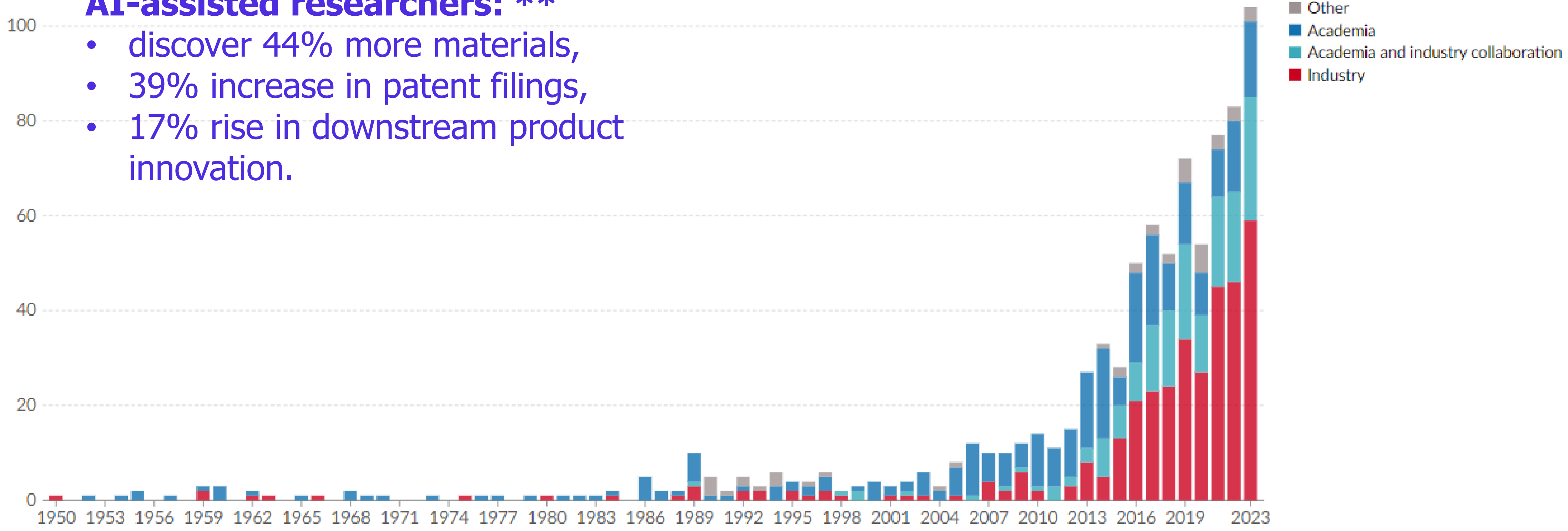
NVIDIA, a new giant!

	Rank	Name	Market Cap	Price	Today	Price (30 days)	Country
☆	1	Apple AAPL	$ 3.518 T	$231.41	▲ 0.36%		USA
☆	2	NVIDIA NVDA	$ 3.471 T	$141.54	▲ 0.80%		USA
☆	3	Microsoft MSFT	$ 3.183 T	$428.15	▲ 0.81%		USA
☆	4	Alphabet (Google) GOOG	$ 2.013 T	$166.99	▲ 1.50%		USA
☆	5	Amazon AMZN	$ 1.971 T	$187.83	▲ 0.78%		USA



Affiliation of teams building notable AI systems
Growing …

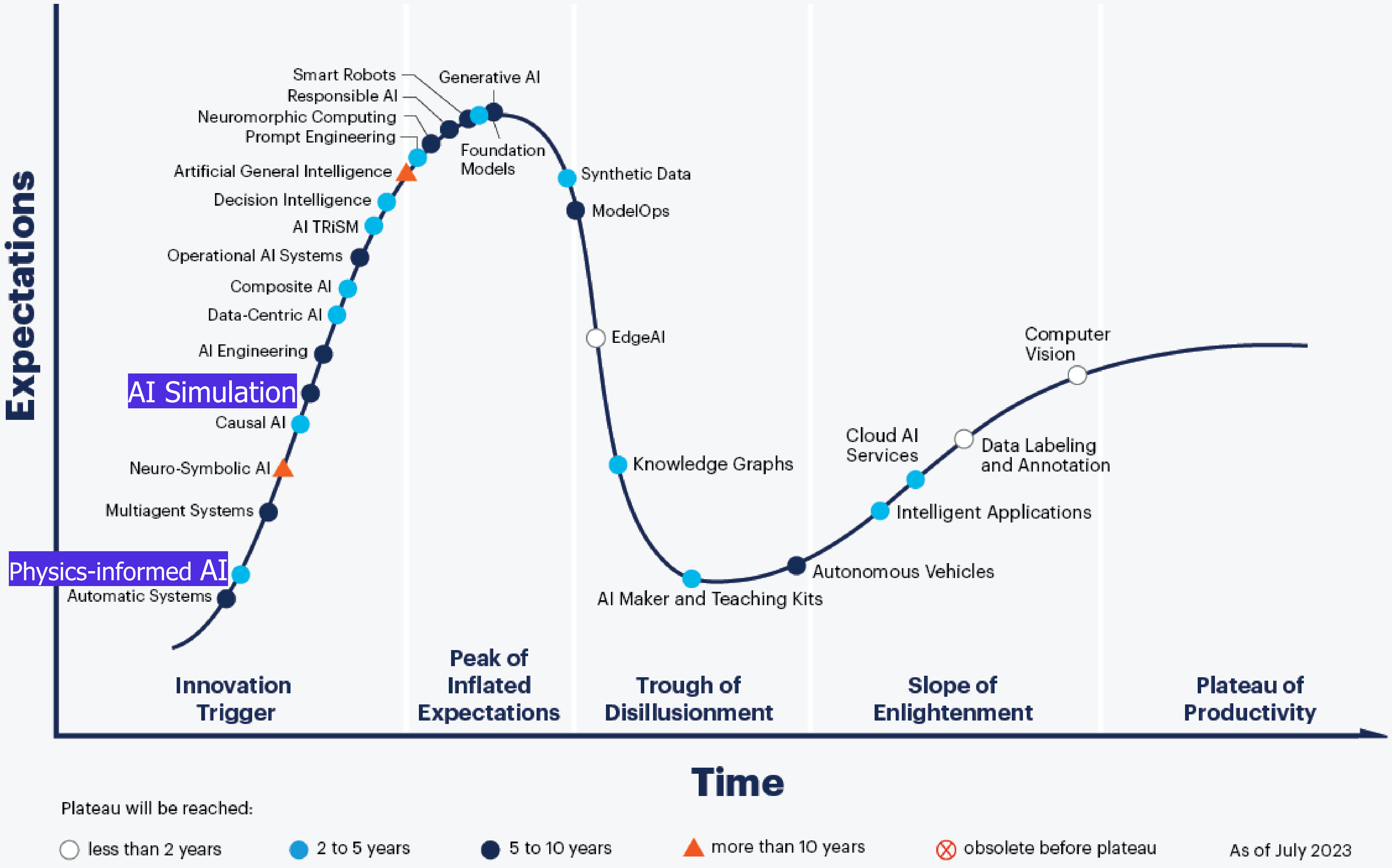
AI-assisted researchers: **
- discover 44% more materials,
- 39% increase in patent filings,
- 17% rise in downstream product innovation.



** Artificial Intelligence, Scientific Discovery, and Product Innovation, Aidan Toner-Rodgers, 2024

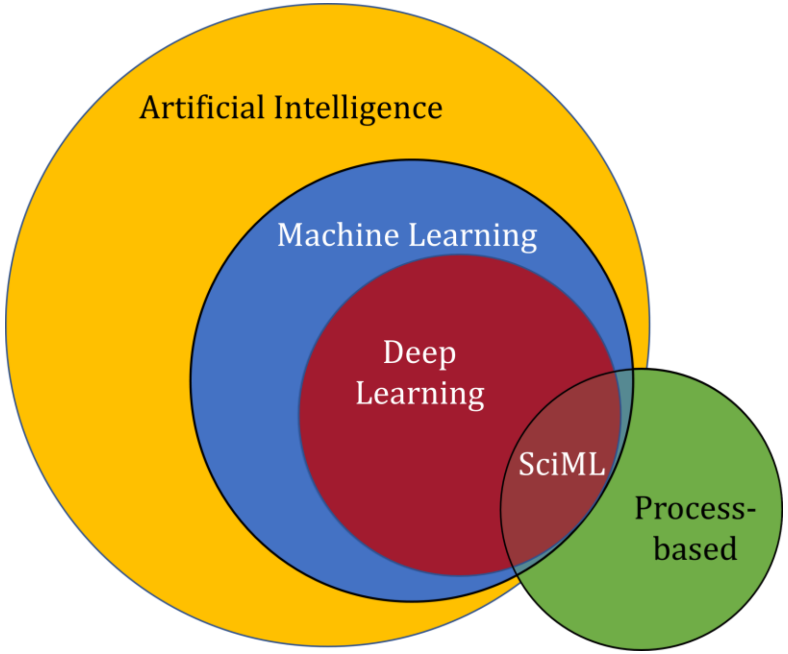
Artificial Intelligence Hype Cycle (2023)

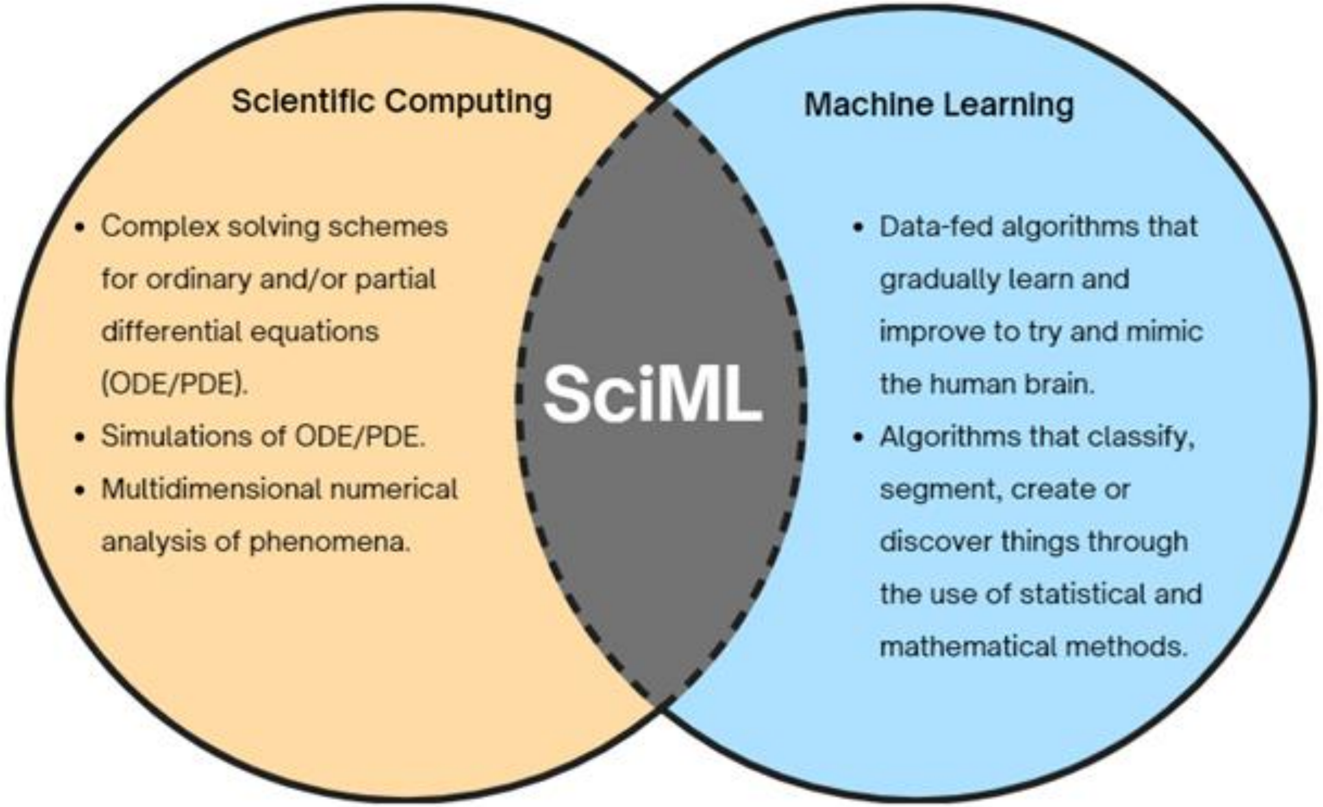
Just Beginning of an Era!



Gartner.

Scientific Computing & Machine Learning for PDEs

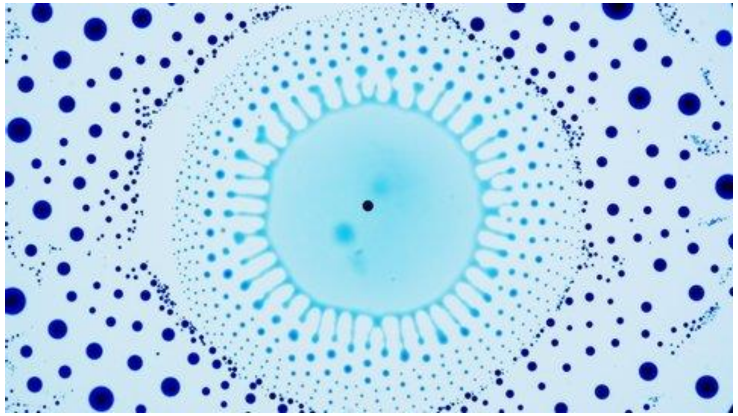




Partial Differential Equations (PDEs)

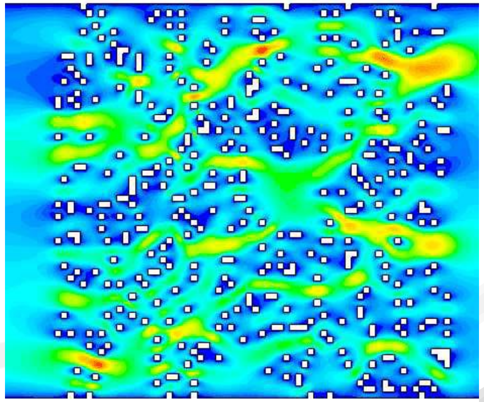
They are everywhere!

Black-holes

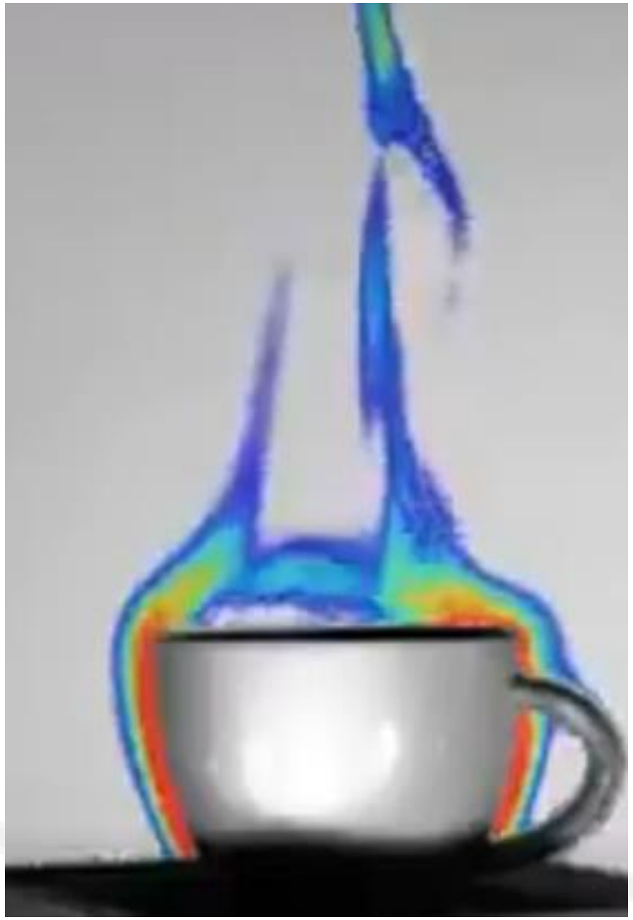


Surface Phenomena

Navier-Stokes Equations



Flow in Porous Media



Cup of Espresso!

Partial Differential Equations (PDEs)

They are everywhere!

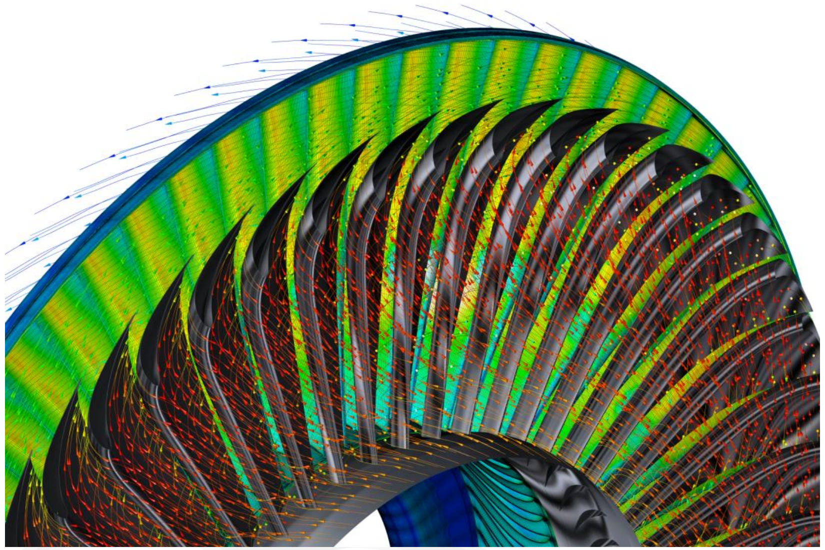
Classification of Second-Order PDEs:

1) Elliptic PDEs (e.g., Laplace's equation):
   Steady-state problems, no time dependency.
2) Parabolic PDEs (e.g., Heat equation):
   Time-dependent problems with diffusion-like behavior.
3) Hyperbolic PDEs (e.g., Wave equation):
   Time-dependent problems with wave-like behavior.

Key Scientific Tasks

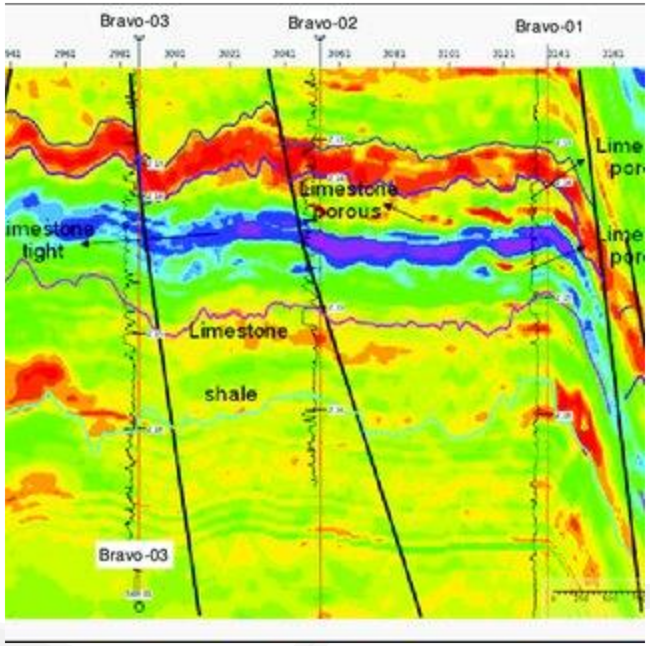
Forward Simulation

y = f(x)



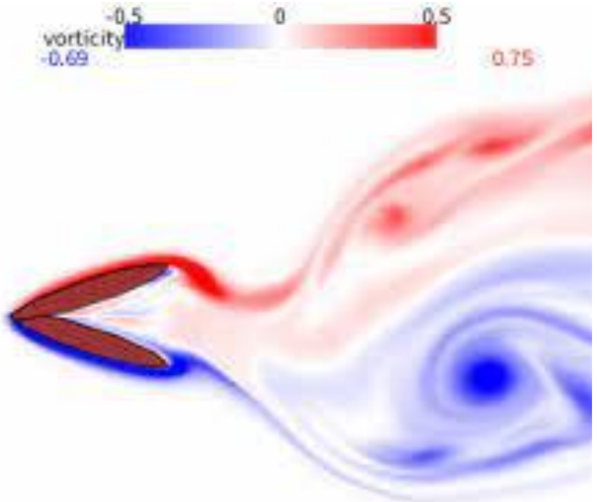
Inverse Simulation

y = f(x)



Equation Discovery

y = f(x)



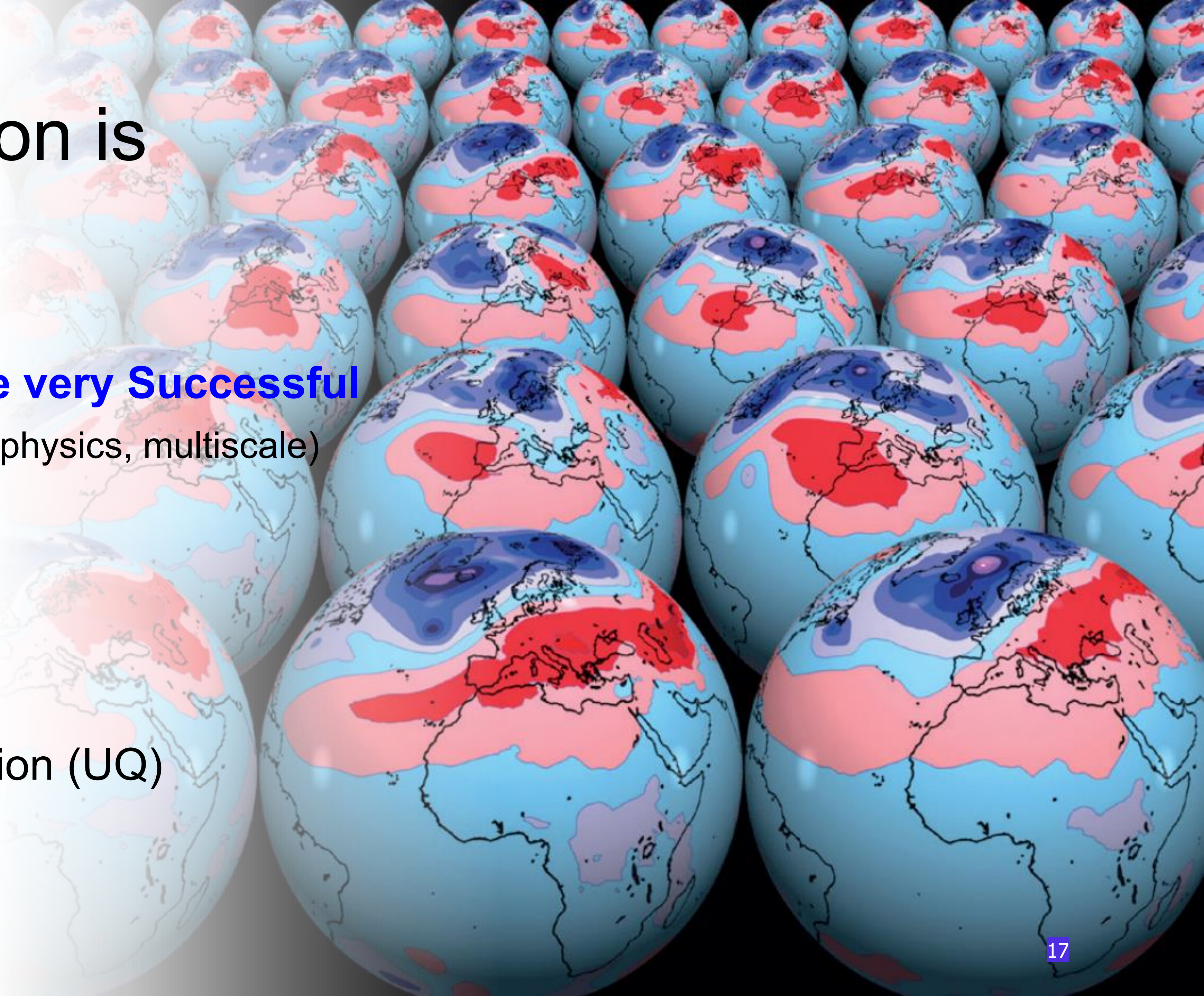
Why a revolution is needed?

Numerical Methods are very Successful

(issues: high resolution, multiphysics, multiscale)

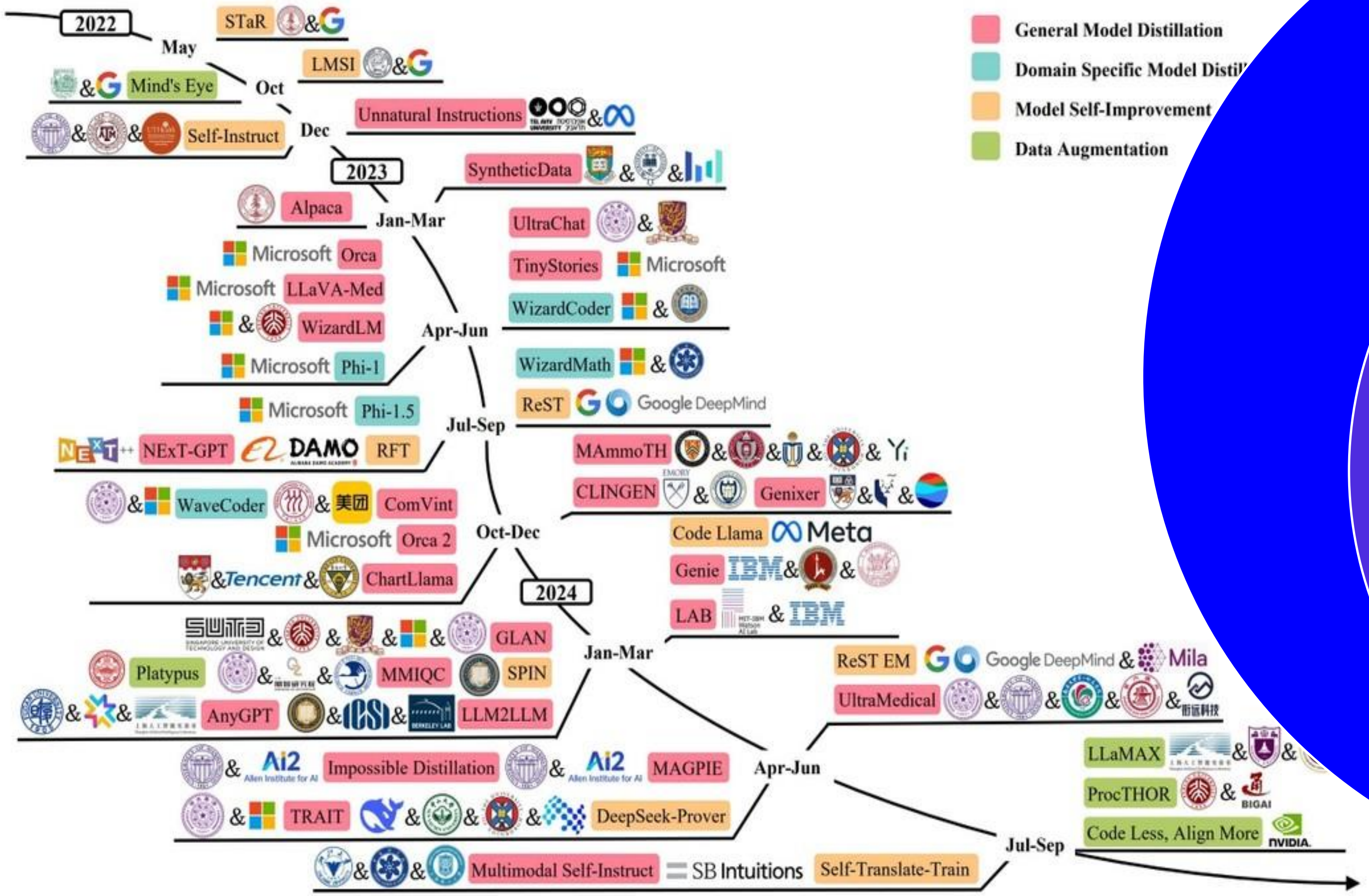
Challenging Tasks:

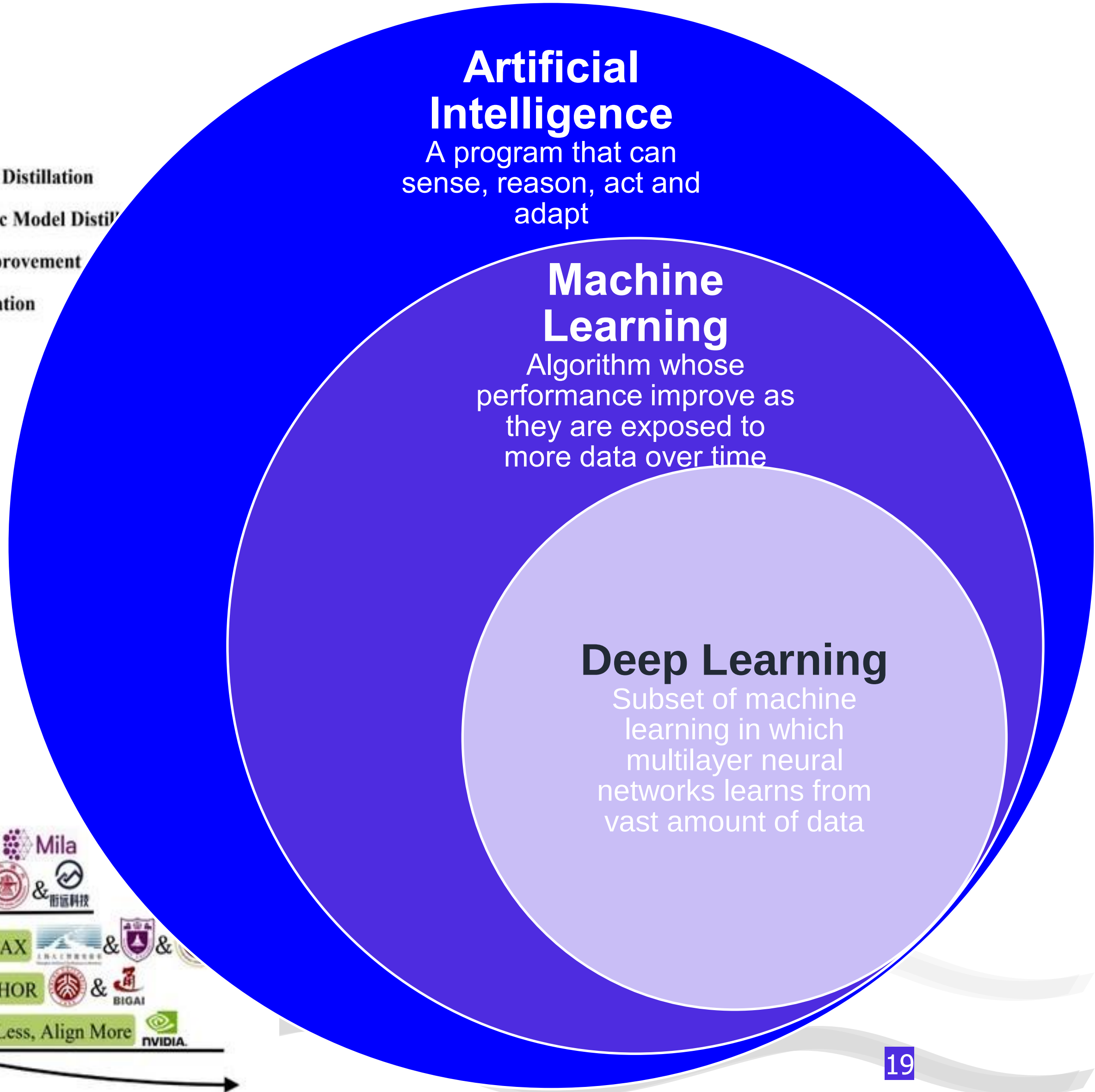
- Optimal Design
- Inverse Problem
- Uncertainty quantification (UQ)



Deep Learning

Deep Learning





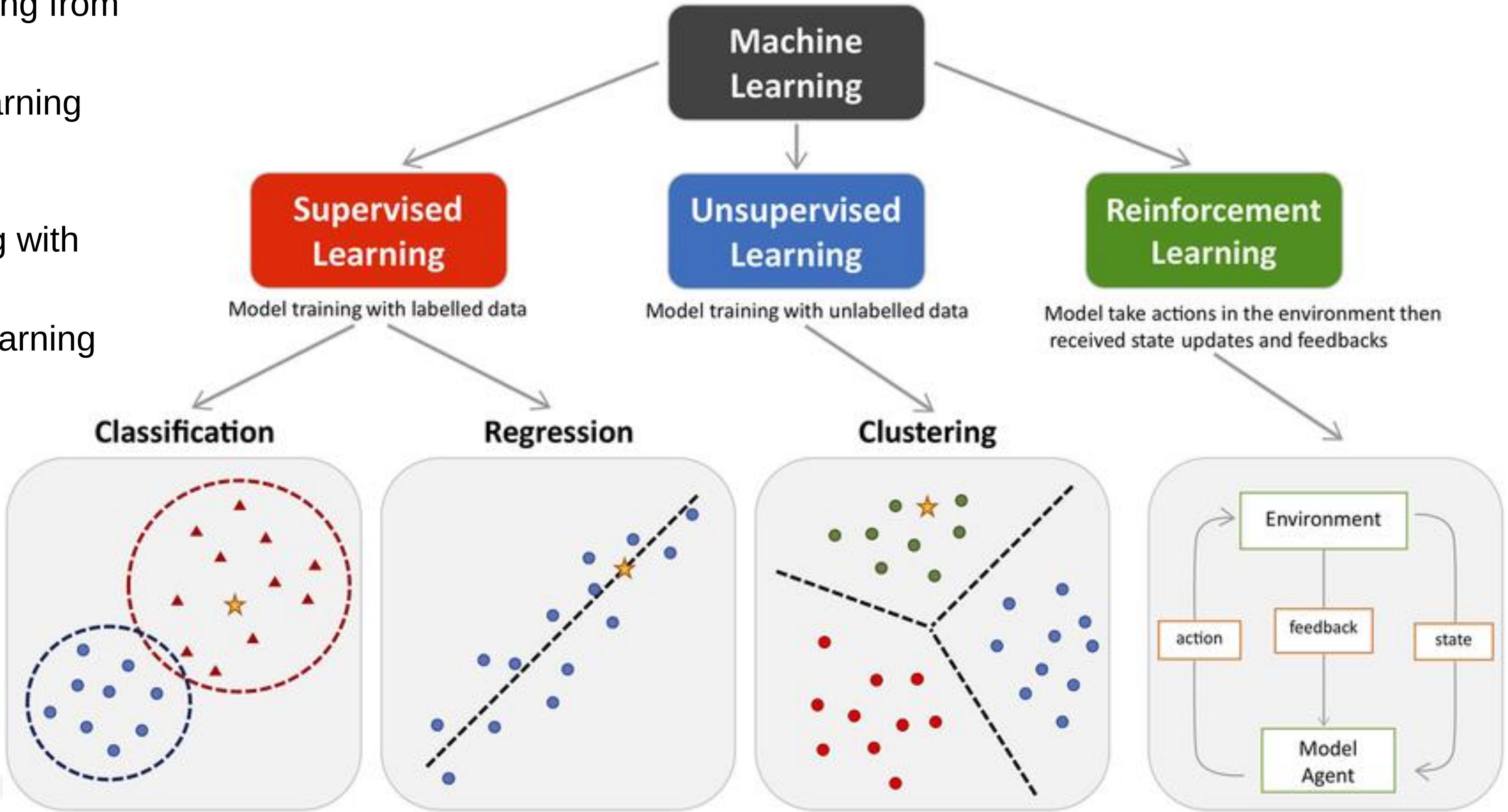
Deep Learning: Concepts

- **Neural Networks** (also, activation functions, loss function)
- **Model Architectures**
- **Training** and **Optimization** Techniques (also, Transfer Learning, Regularization, Hyperparameter Tuning)
- **Supervised** / **Unsupervised** / **Self-supervised** learning
- **Reinforcement Learning** (RL):
- **Bayesian Neural Networks** (BNNs)
- Emerging Topics: Multimodal Models, **Foundation models**, Generative AI

GitHub - rasbt/deeplearning-models

Types of Learning in Deep Learning

- **Supervised learning:** learning from samples with labels
- **Unsupervised learning:** learning from samples without labels
- **Semi-supervised learning:** augment supervised learning with unlabeled data
- **Reinforcement learning:** learning through the interaction with environment

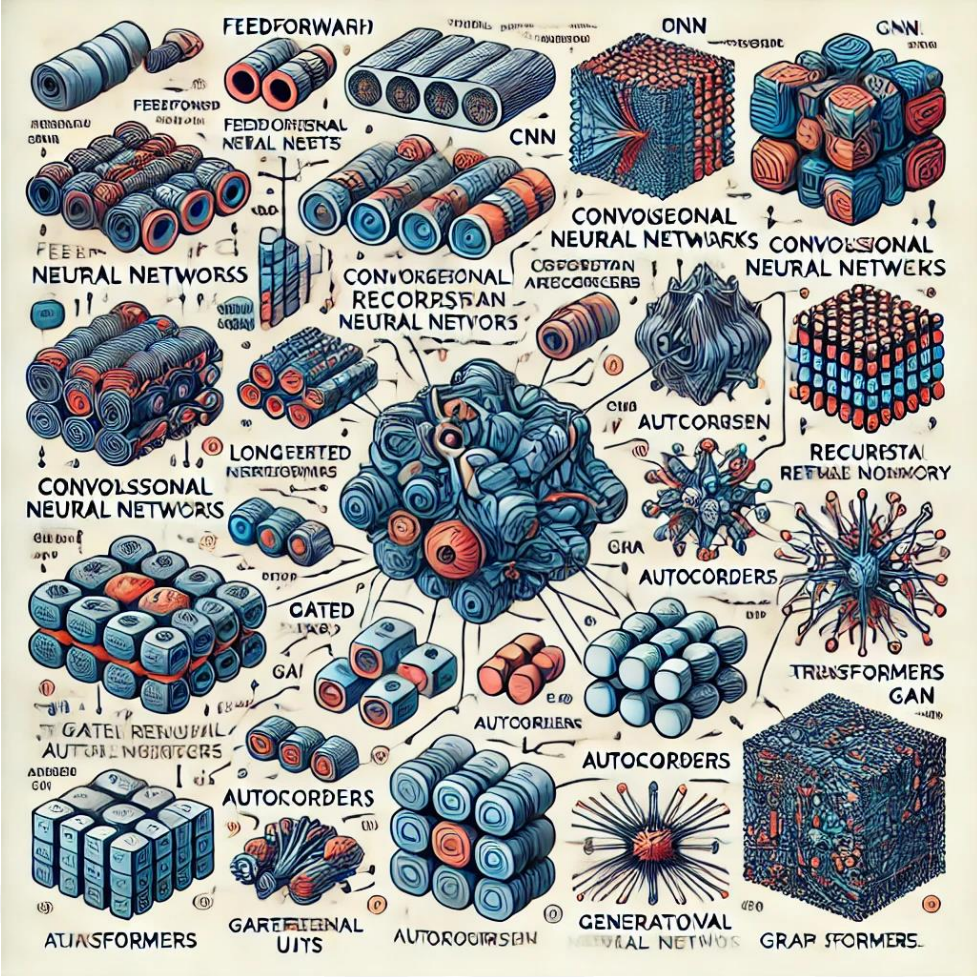


http://dx.doi.org/10.3389/fphar.2021.720694

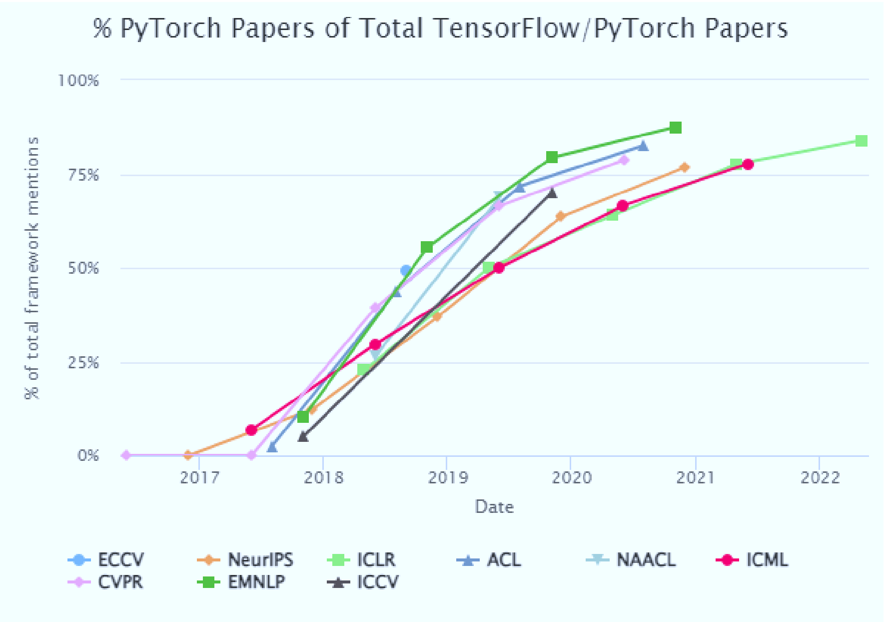
Deep Learning: Architecture

- Feedforward Neural Networks (FNNs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Long Short-Term Memory Networks (LSTMs)
- Gated Recurrent Units (GRUs)
- Autoencoders
- Variational Autoencoders (VAEs)
- Generative Adversarial Networks (GANs)
- Transformers
- Graph Neural Networks (GNNs)

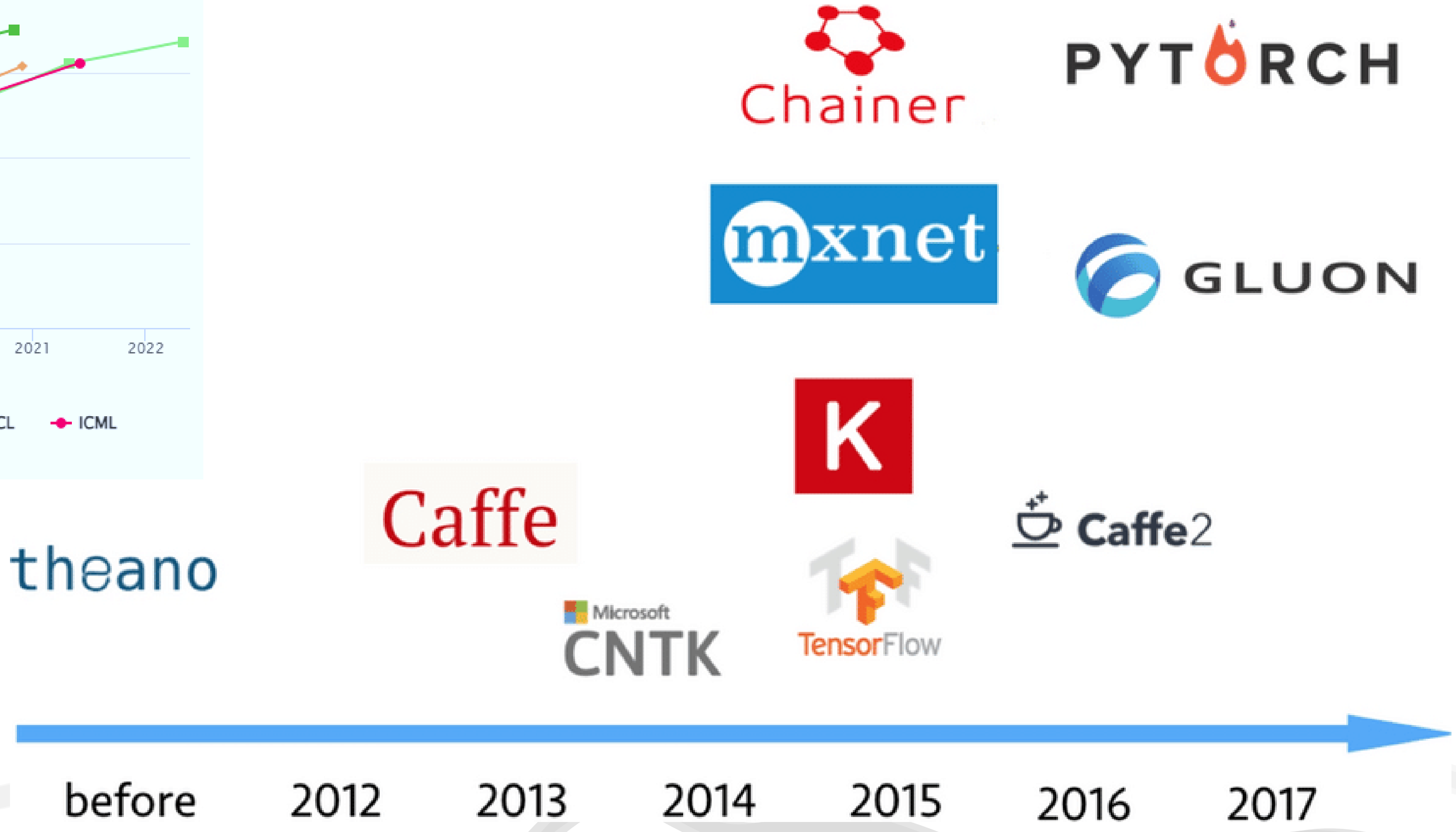
ChatGPT 4o



Libraries & Frameworks



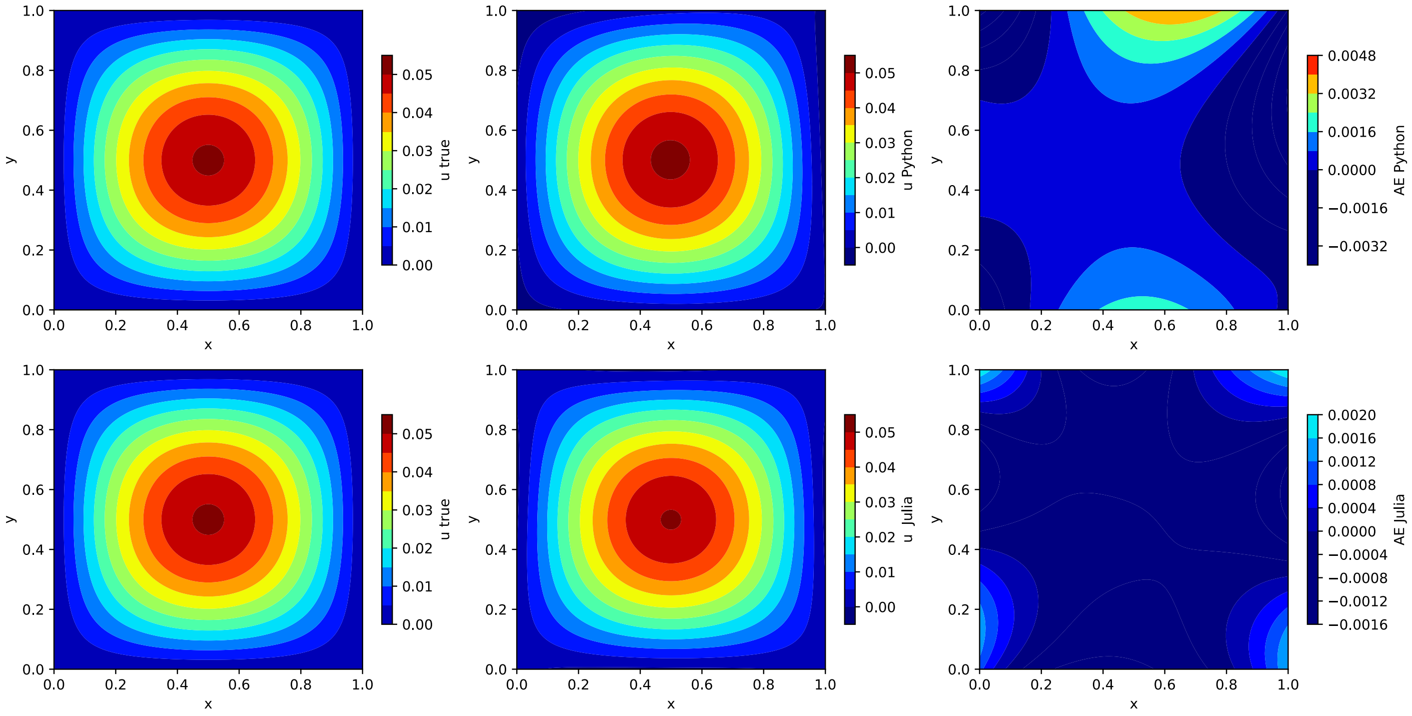
https://miguelgfierro.com/blog/2022/an-analysis-of-the-adoption-of-top-deep-learning-frameworks/

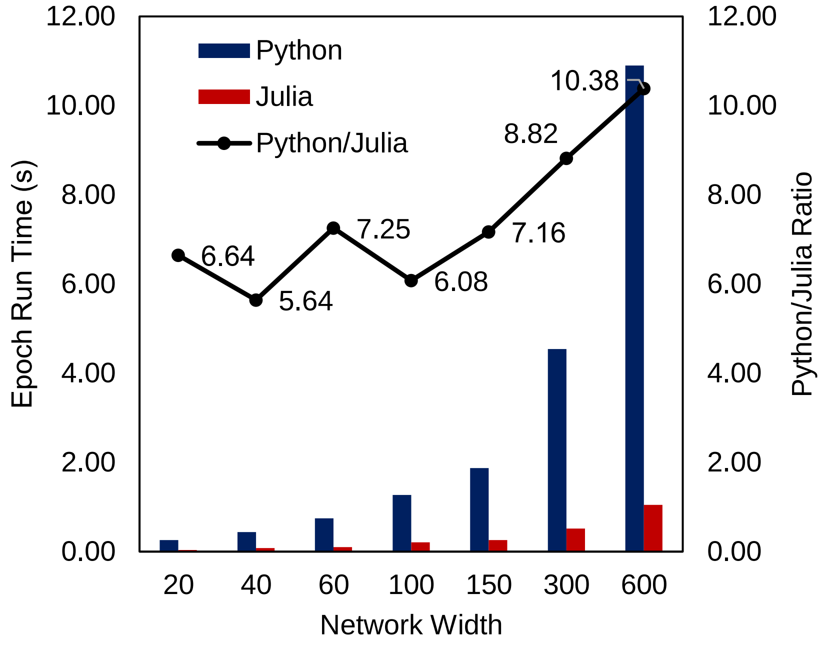


Deep Learning Framework Power Scores 2018

Programming Language also Matters

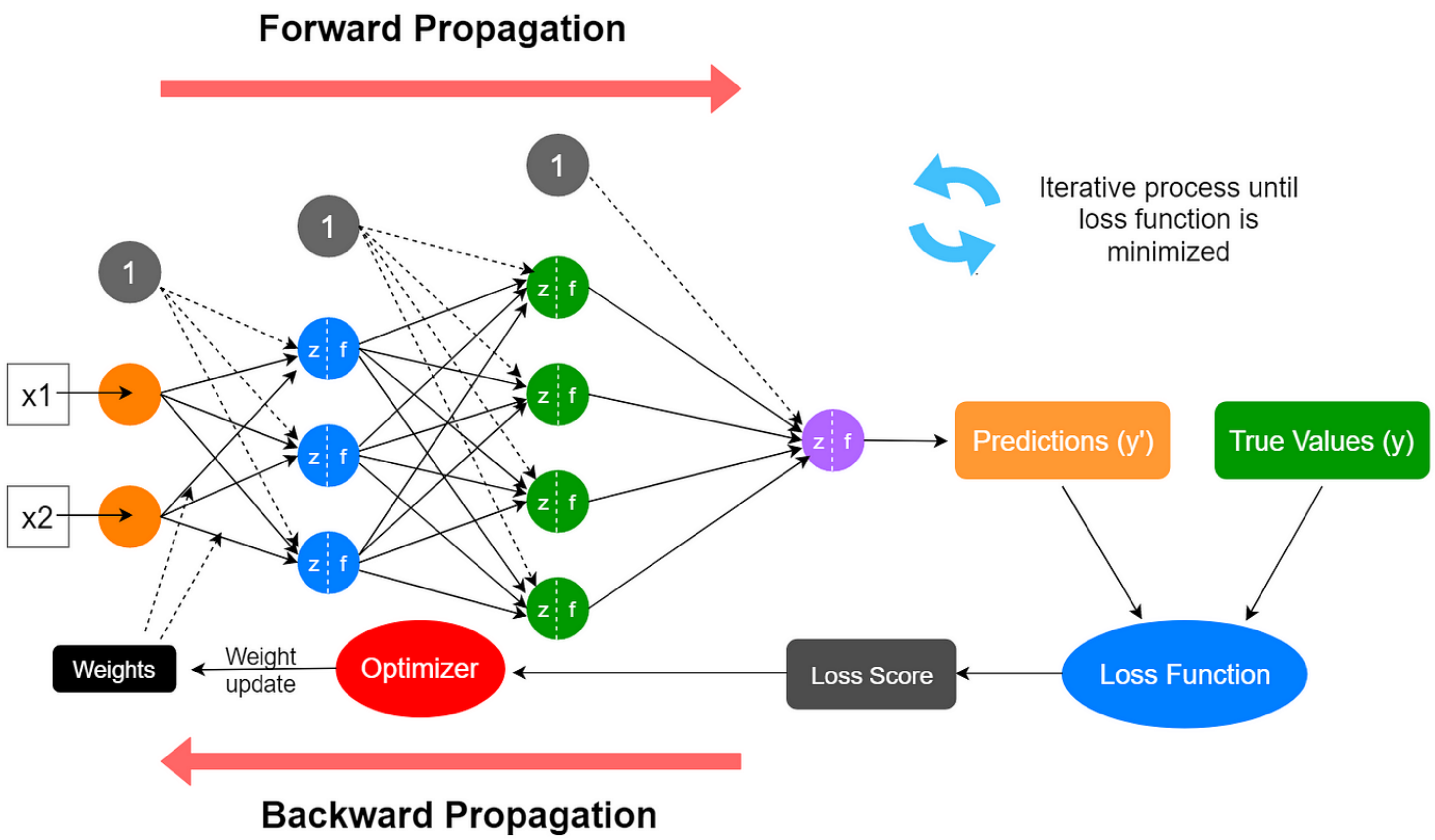
Python vs. Julia

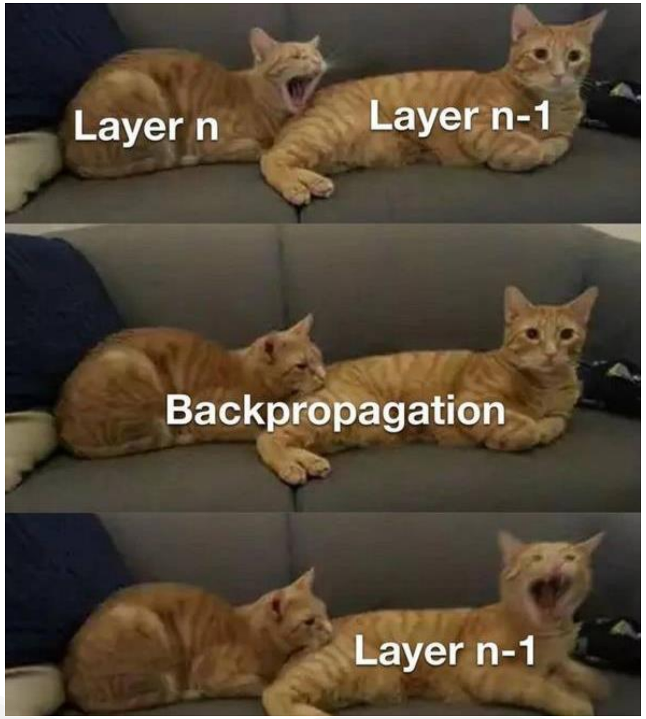




•When Python code is highly optimized with frameworks and GPU acceleration, the performance gap narrows significantly.

Training Process





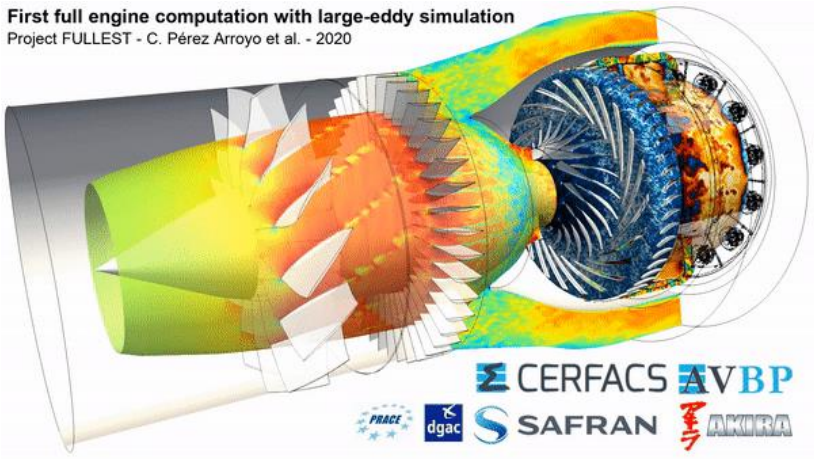
A short break

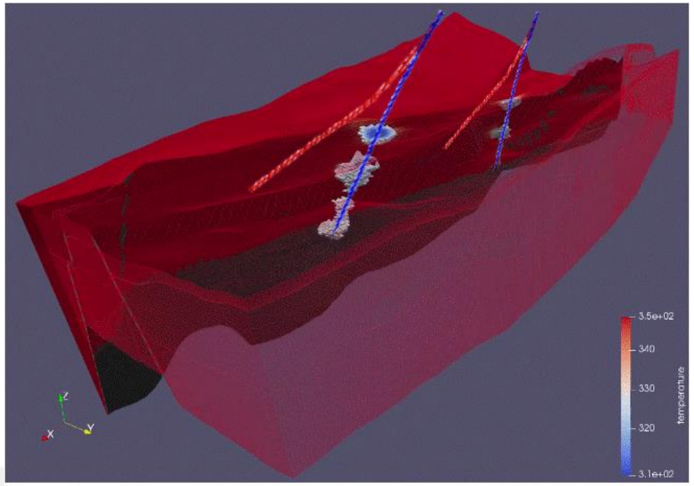
Scientific Machine Learning

Incorporating Physics into Learning

Why Numerical Methods?

- Overcome Analytical Limitations: Handles complex geometries and real-world problems where analytical methods fail.
- Improved Modeling: Approximate solutions often outperform oversimplified "exact" ones.
- Flexibility: Supports parametric studies and diverse scenarios.
- Simplify Complex Solutions: Reduces the effort of dealing with intimidating analytical formulas.
- Enhanced Usability: Provides efficient computations and visually appealing outputs for interpretation.





DARTS - Delft Advanced Research Terra Simulator

Why NOT Numerical Methods?

Current numerical methods come with several limitations and challenges:

1. **Computational Complexity**: Solving PDEs numerically can be computationally demanding, particularly for large-scale and real-world problems.
2. **Independent Computations**: Each instance of the input space needs to be processed independently when using numerical methods (no interaction and mutual computations).
3. **Exact Physics Knowledge**: Numerical methods heavily rely on a precise understanding of the physics governing the problem.
4. **Data Agnostic**: Traditional numerical methods are *not designed to leverage existing datasets* for learning or data-driven approaches.

Incorporating Physics into Learning

- **Semi-supervised Learning**

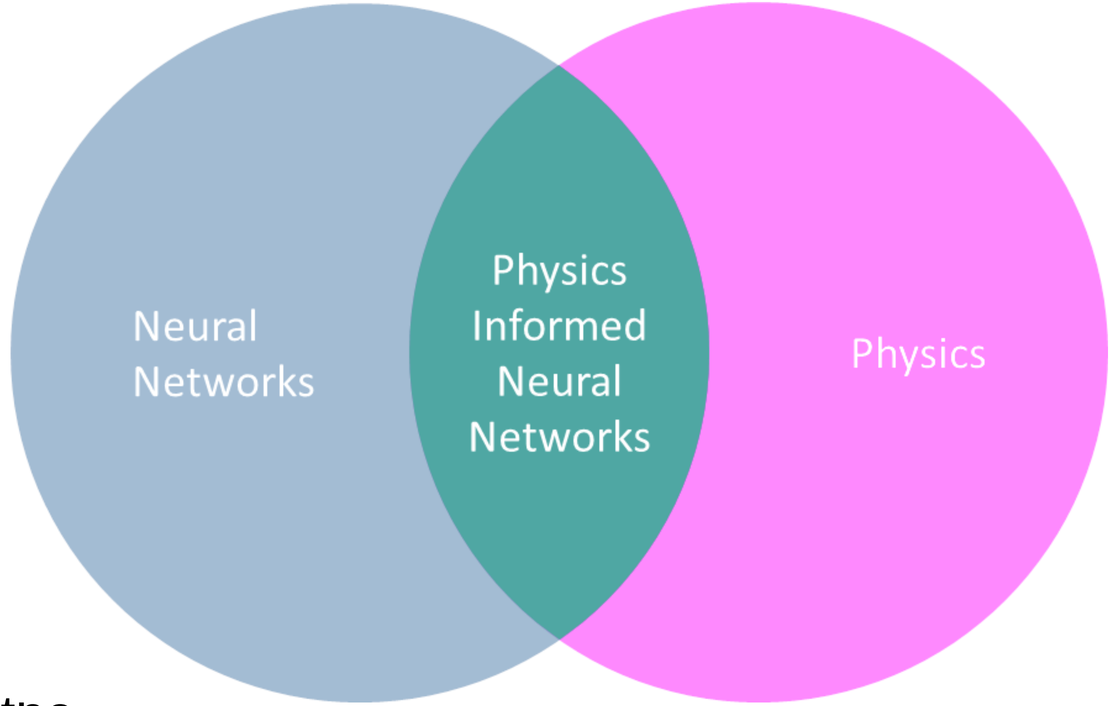
(Physics Informed Learning)

Enforce physical laws as hard/soft constraints

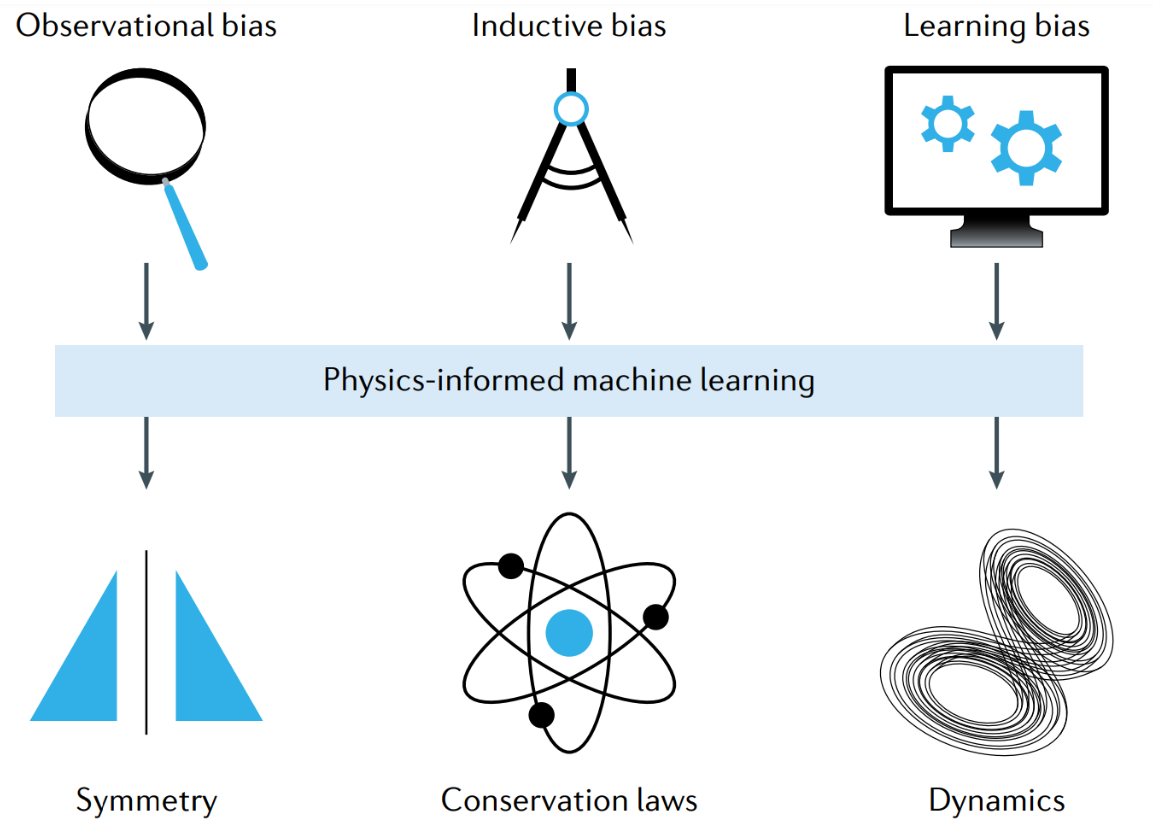
- **Supervised Learning**

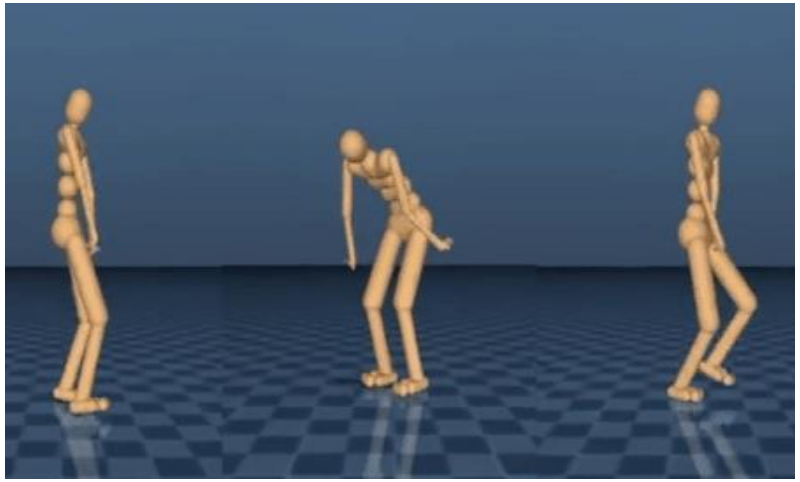
(Operator Learning)

Train on large amount of data and let the NN learn the physics based operators



Physics Informed Machine Learning





> **PINNs:** Great for some PDEs, particularly with low complexity, and low training data. Have several negatives. We need alternatives

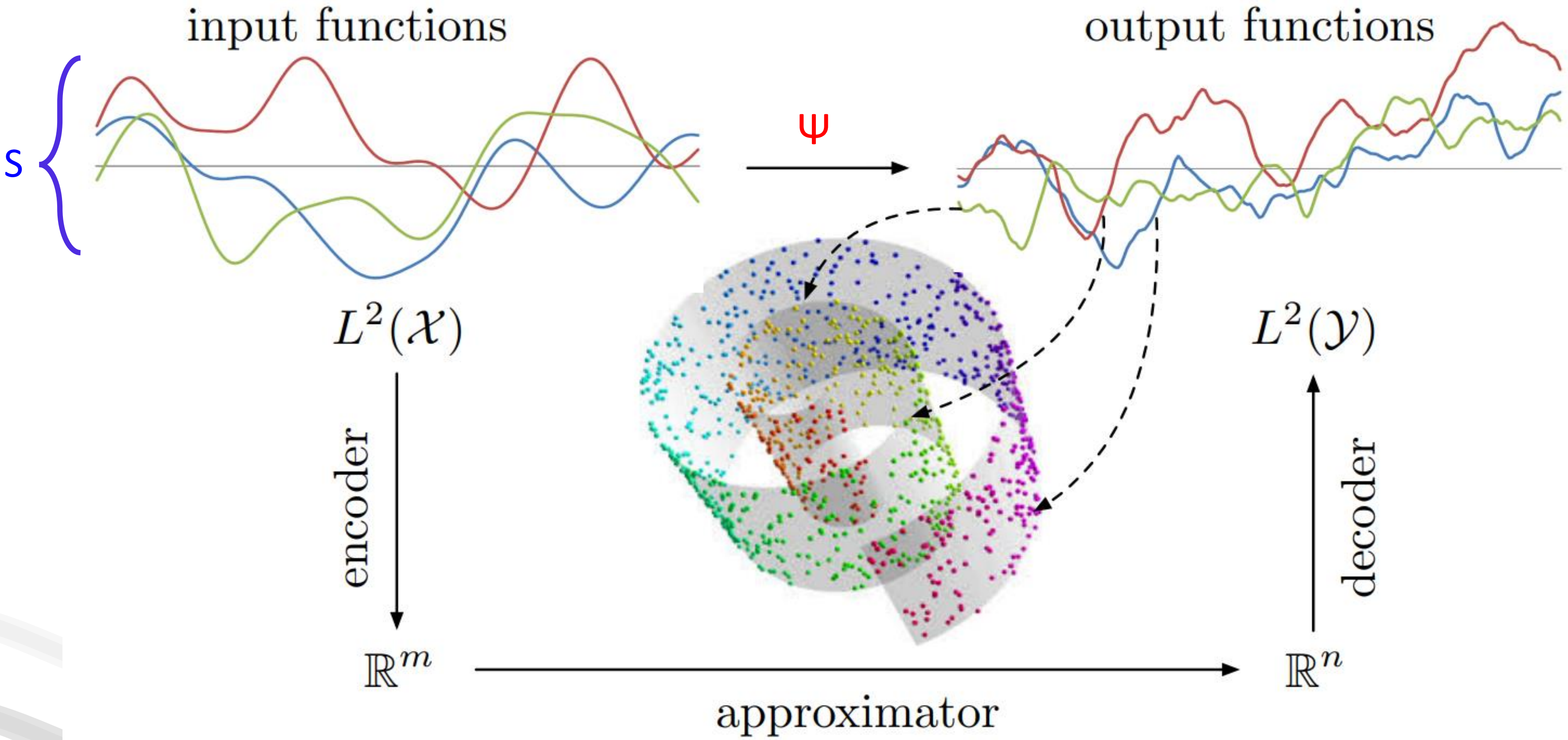
Karniadakis et al., 2021

Operator Learning ($\Psi : X \to Y$)

- Goal: Learn Ψ from samples S: Supervised Deep Learning

> Train on large amount of data (and hope) that the NN learns the constraints

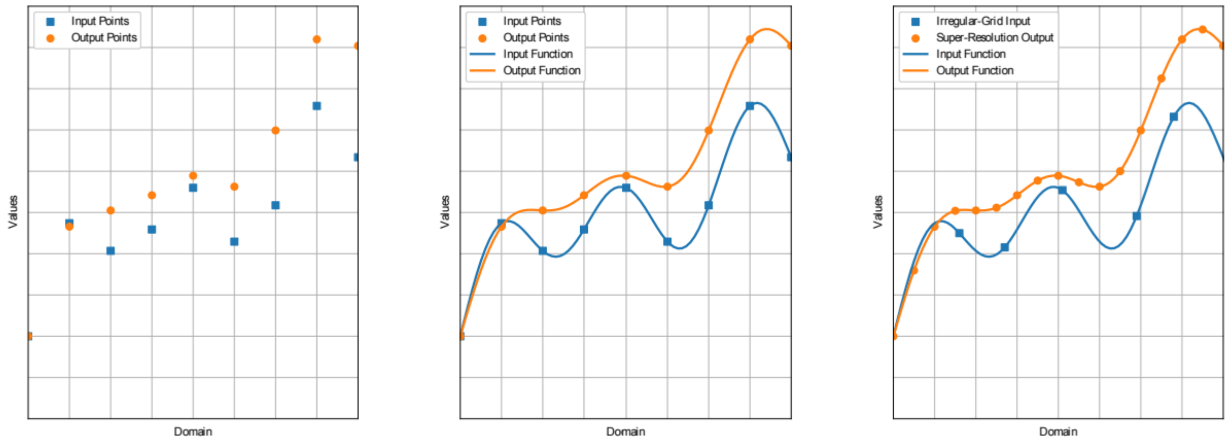
Task: Find a Surrogate (based on DNNs) from data.



mapping from function spaces

Operator Learning ($\Psi : X \to Y$)

Comparison of neural networks with neural operators



(a) NN learns a mapping between input and output points on a fixed, discrete grid.

(b) NO maps between functions on continuous domains, even when training data is on a fixed grid.

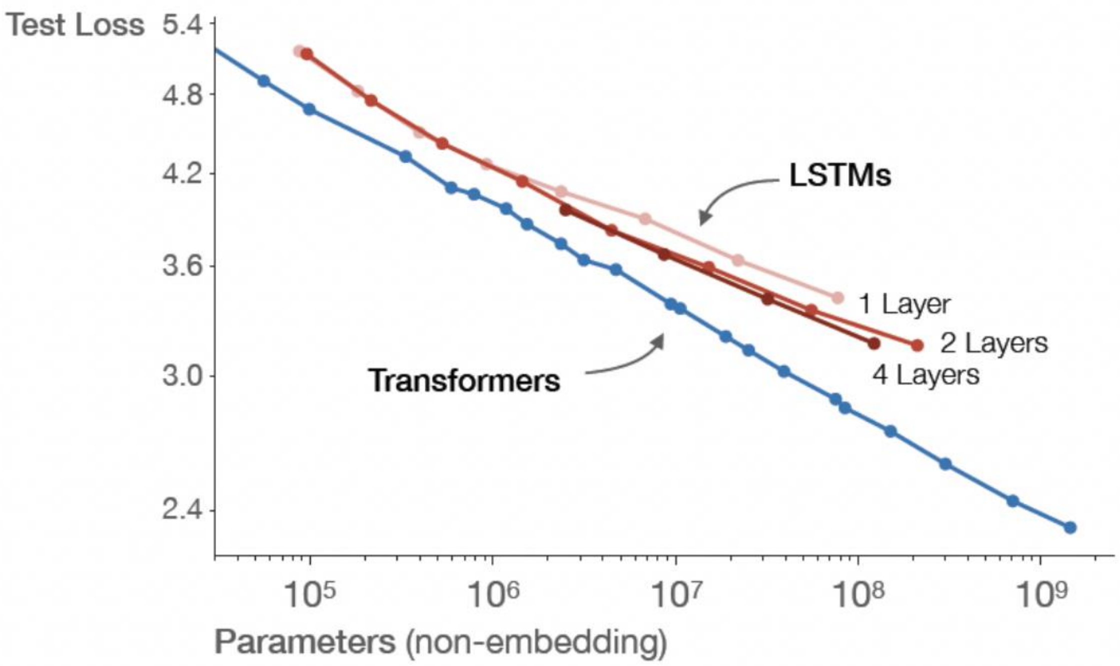
(c) NO maps between functions, so it accepts inputs outside the training grid, and can do super-resolution.

Property \ Model	NNs	DeepONets	Interpolation	Neural Operators
Discretization Invariance	✗	✗	✓	✓
Is the output a function?	✗	✓	✓	✓
Can query the output at any point?	✗	✓	✓	✓
Can take the input at any point?	✗	✗	✓	✓
Universal Approximation	✗	✓	✗	✓

Azizzadenesheli, Kamyar, et al. "Neural operators for accelerating scientific simulations and design." *Nature Reviews Physics* (2024): 1-9.

Operator Learning ($\Psi : X \to Y$)

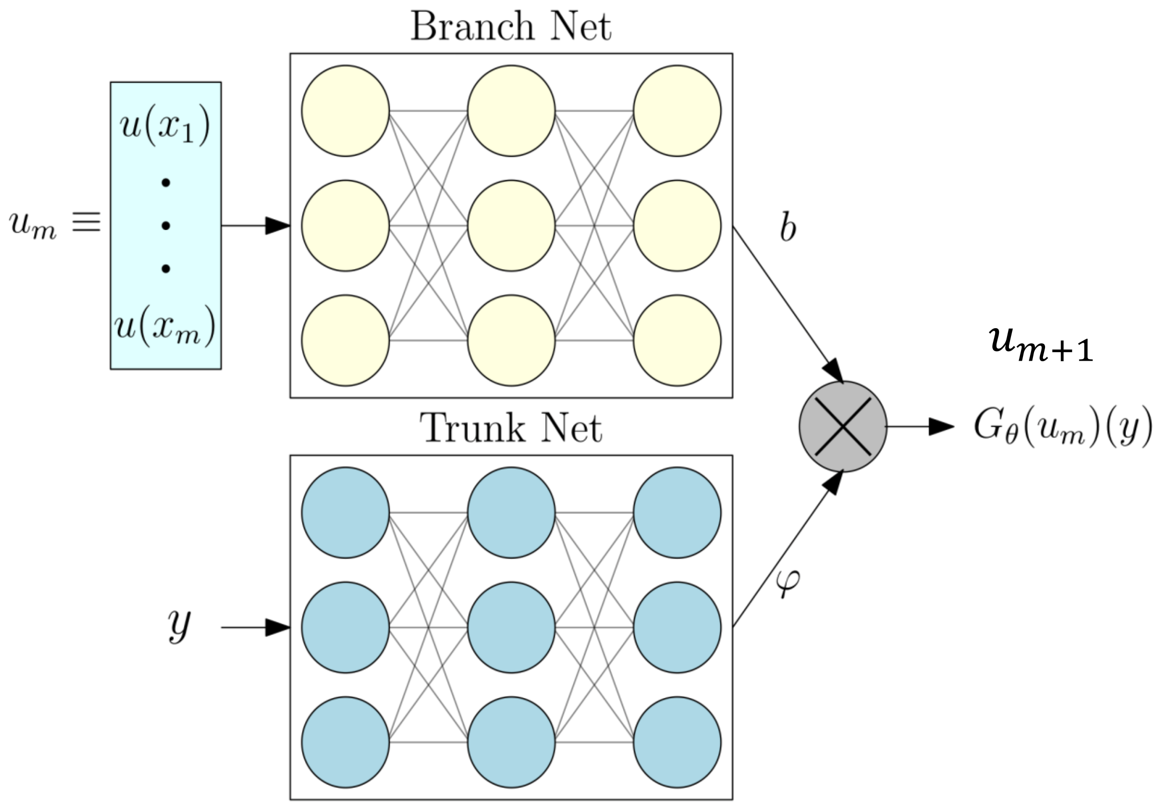
- DeepONet
- Fourier Neural Operator (FNO)
- Convolutional Neural Operator (CNO)
- Graph Neural Operator (GNO)
- Operator U-Net architecture
- GenCFD
- …



https://neuraloperator.github.io/dev/index.html

DeepONet (Deep Operator Network)

- Branch Network: Processes the input function by evaluating it at a set of fixed sensor points. These evaluations are then transformed into a feature representation.
- Trunk Network: Takes the coordinates where the output function is to be evaluated and generates corresponding basis functions.

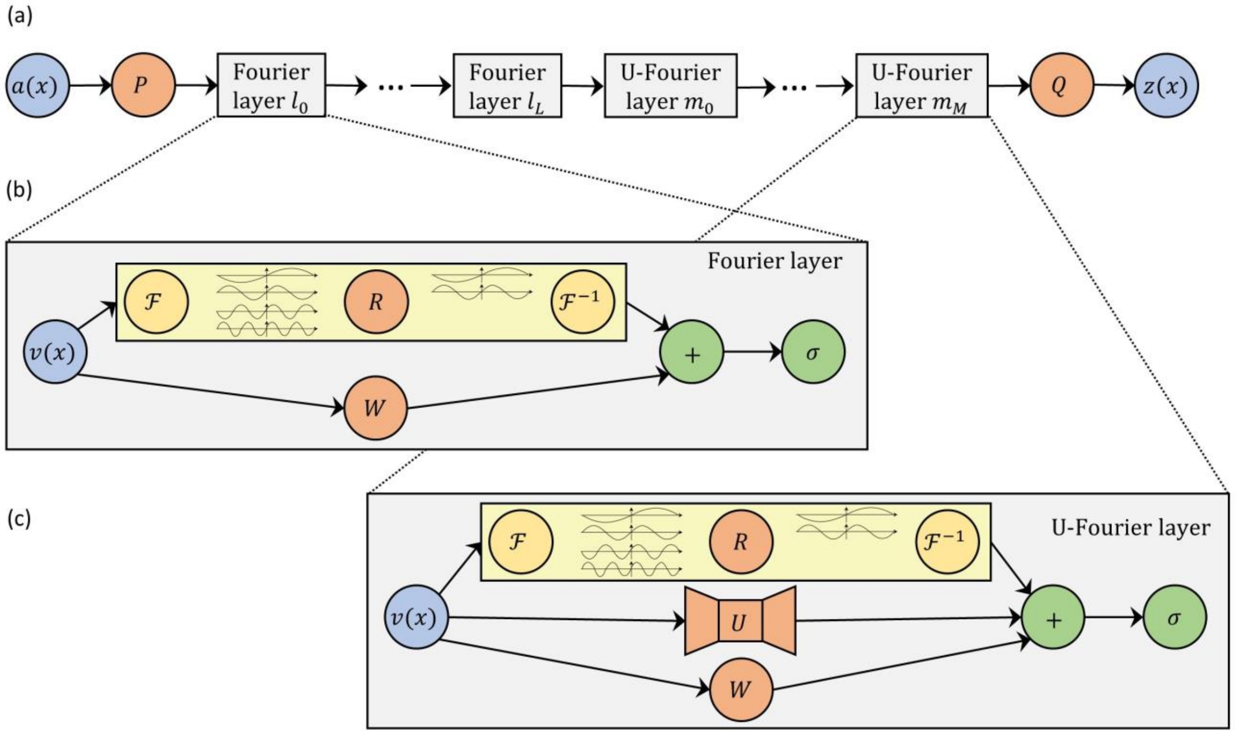


https://www.mdpi.com/1999-4893/15/9/325

https://arxiv.org/abs/1910.03193

Fourier Neural Operator (FNO) (2020)

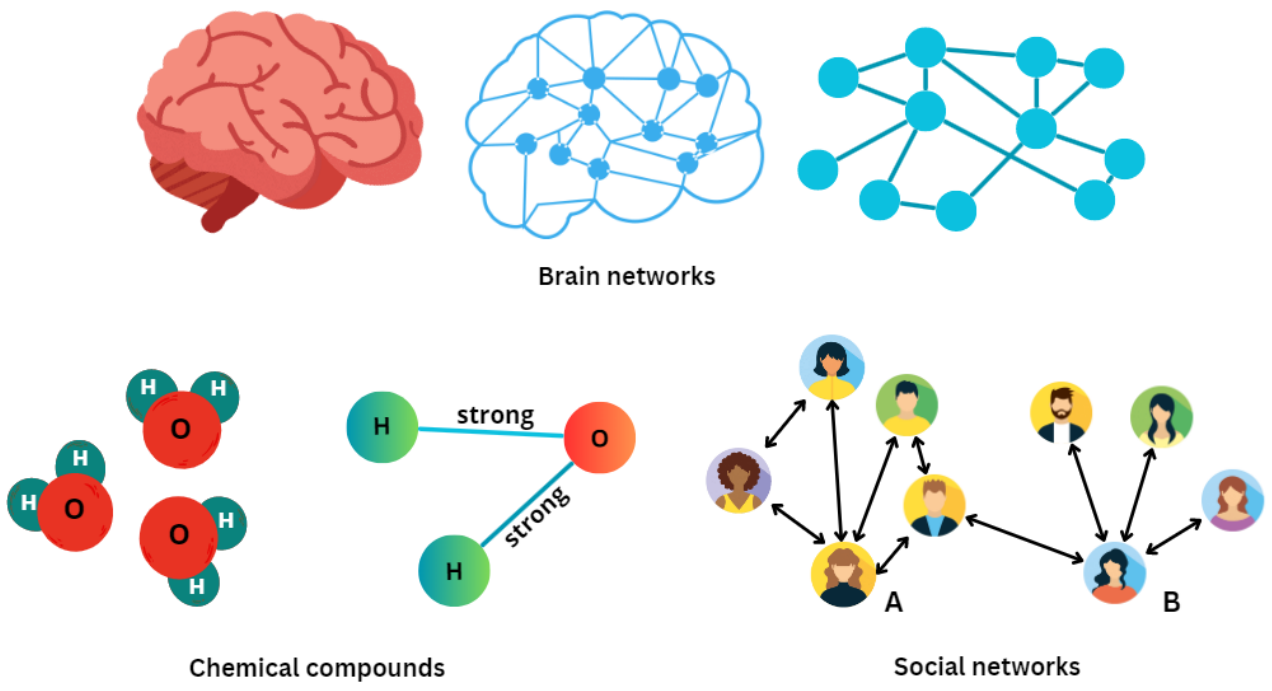
- **Architecture**: A series of Fourier transforms, convolutional layers, and Fourier layers to model long-range dependencies and complex spatial patterns.
- **Parameter Count**: Ranges from **millions to tens of millions** of parameters.
- **Application**: Designed to handle high-dimensional PDEs, particularly those involving turbulence or nonlinear dynamics, like the Navier-Stokes equations.
- **Use Case**: Applications in complex fluid dynamics and atmospheric modeling due to FNO's ability to capture intricate spatial structures over larger domains.



https://arxiv.org/abs/2010.08895

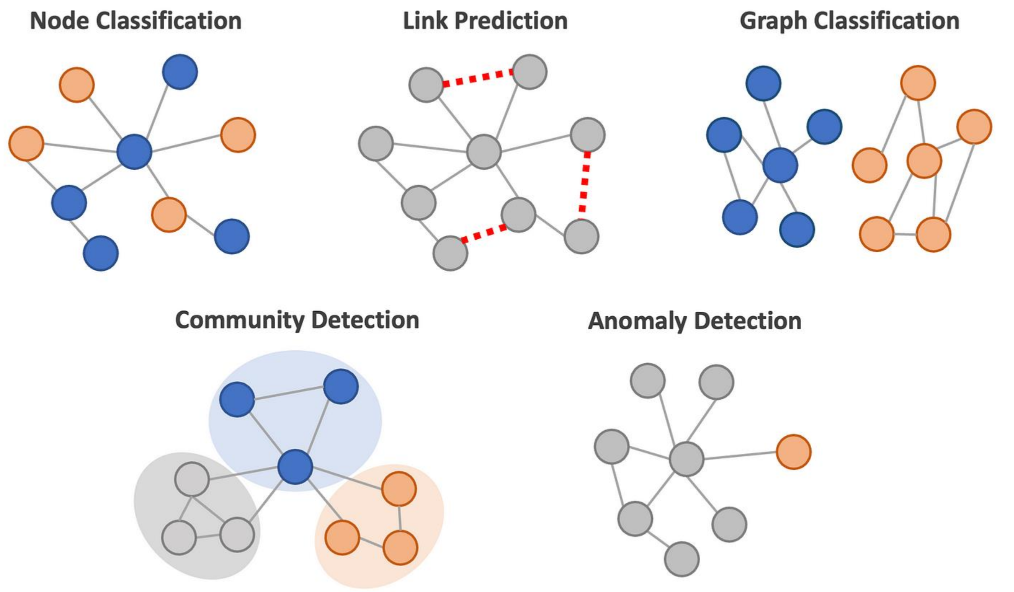
Graph Neural Networks

Graphs : Nodes + Edges



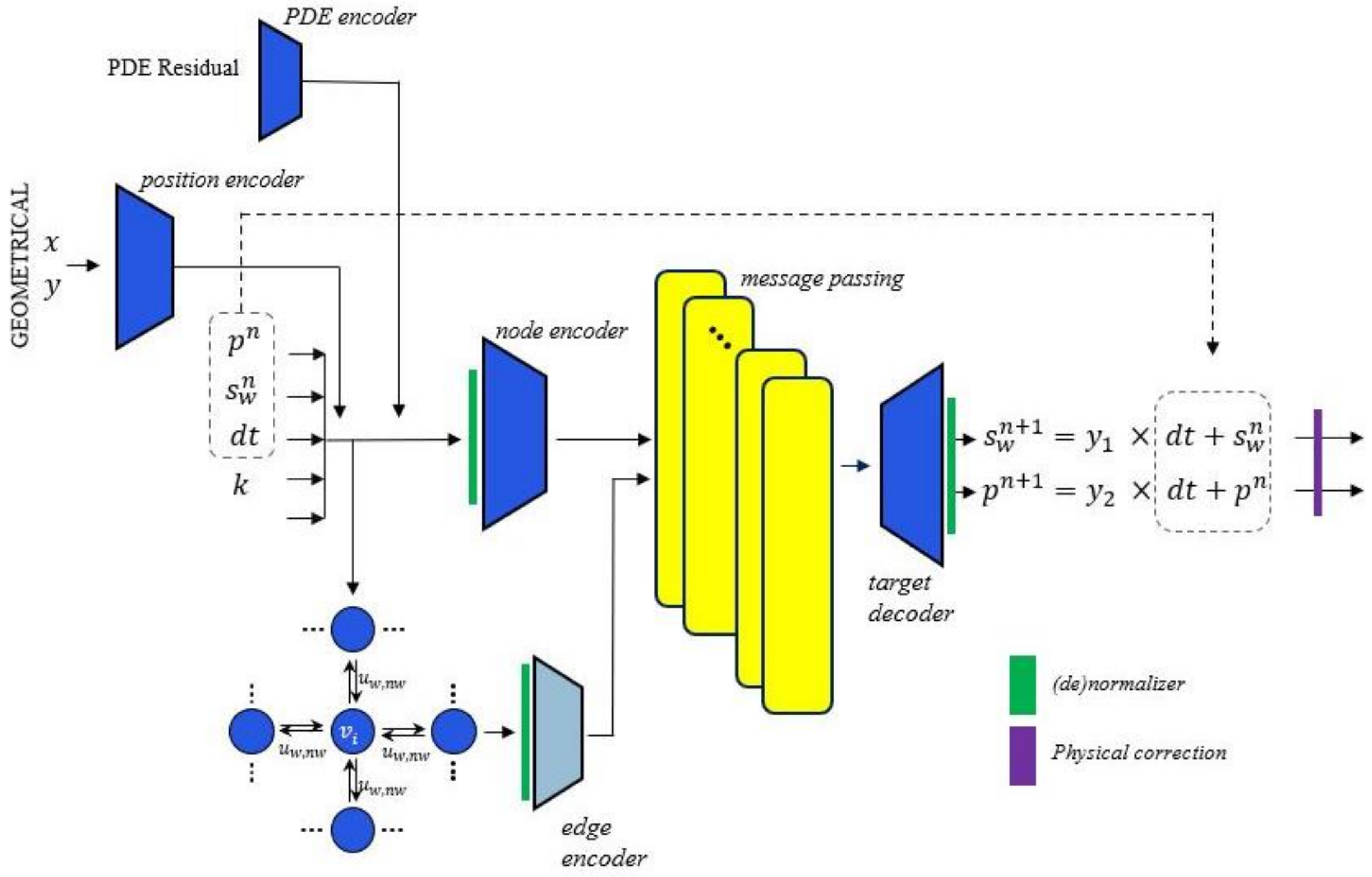
https://medium.com/@bscarleth.gtz/introduction-to-graph-neural-networks-an-illustrated-guide-c3f19da2ba39

Interesting Possibilities



A nice course: https://gnn.seas.upenn.edu/

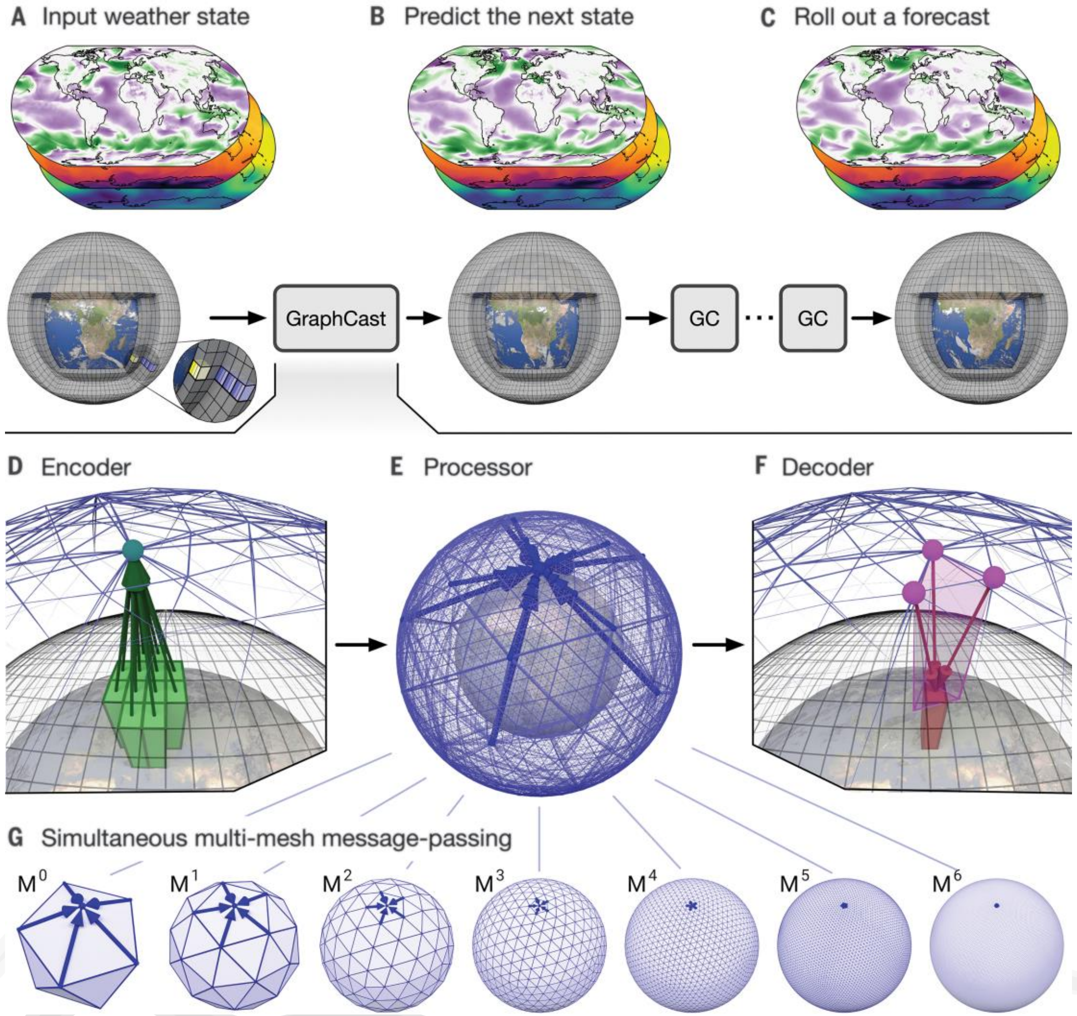
Graph Neural Networks



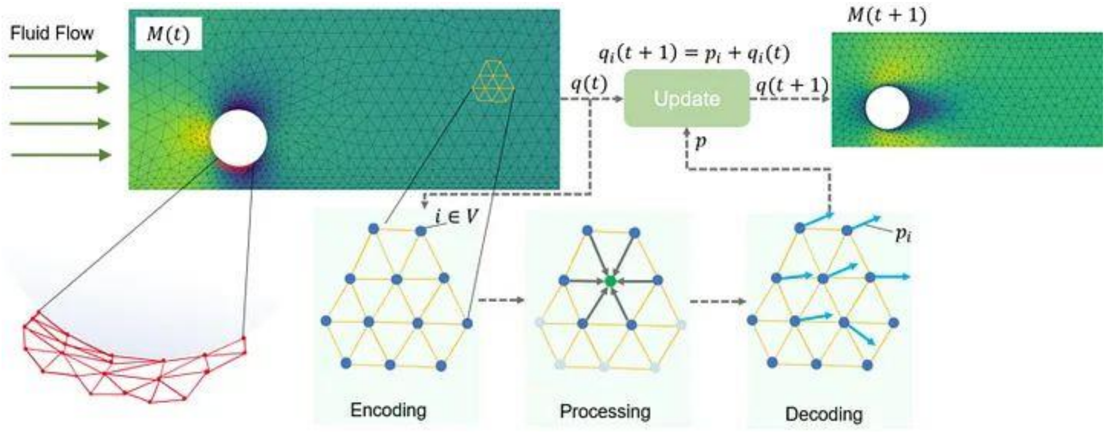
Graph Neural Networks

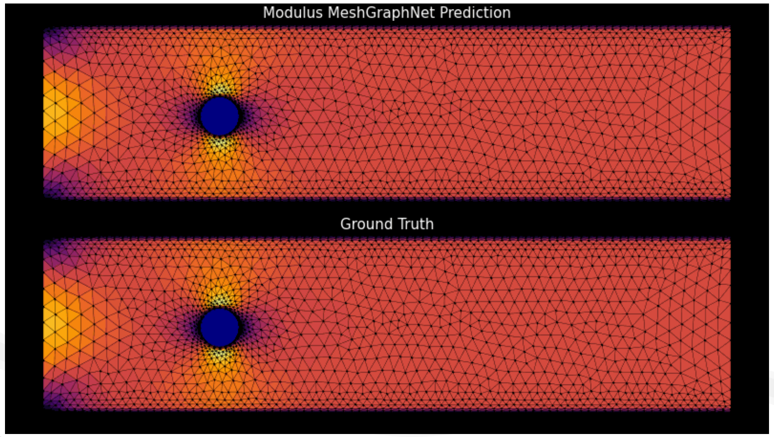
Google DeepMind

GraphCast



https://www.science.org/stoken/author-tokens/ST-1550/full

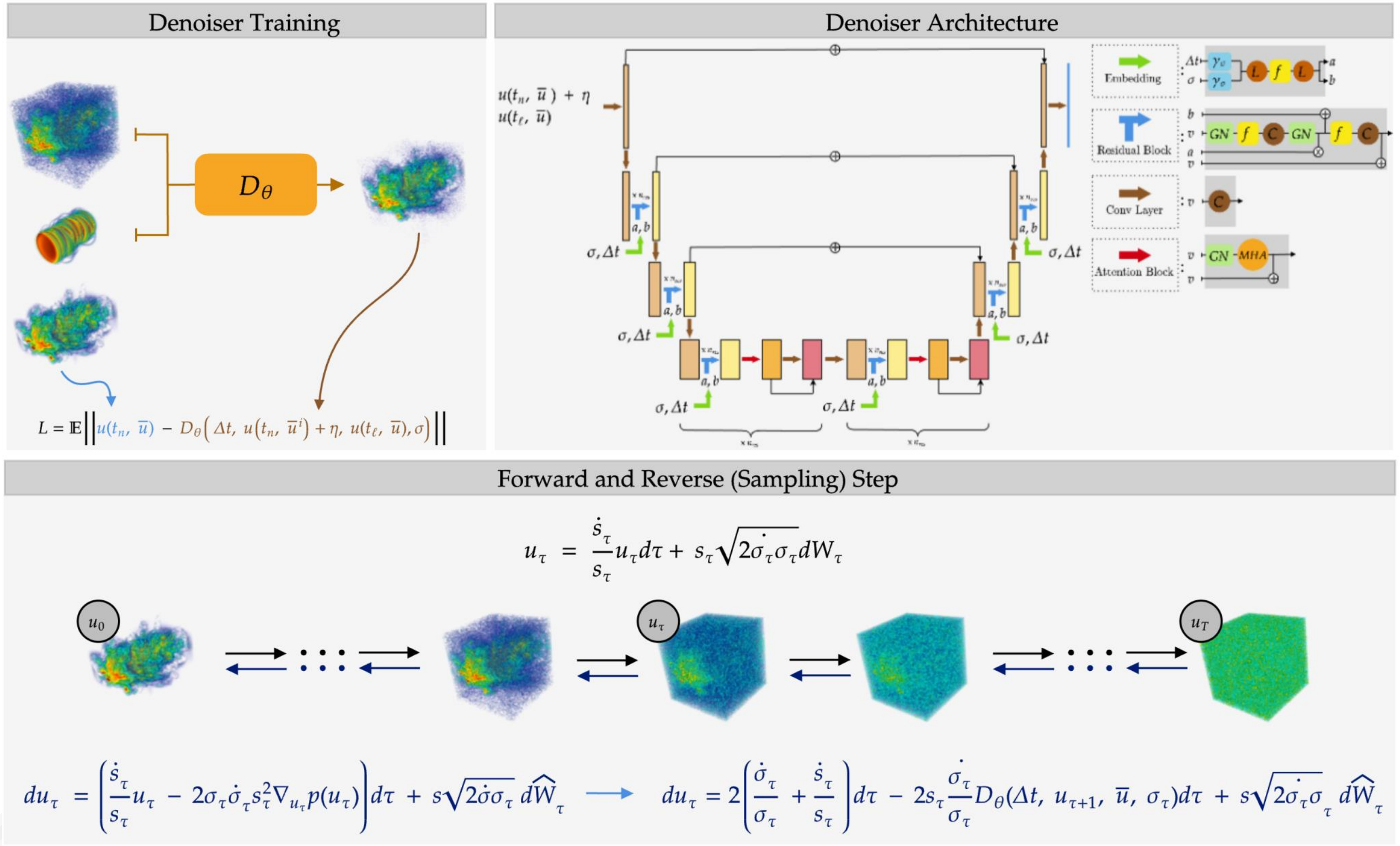


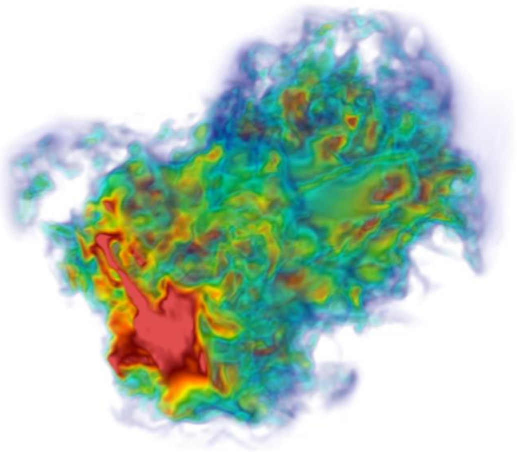


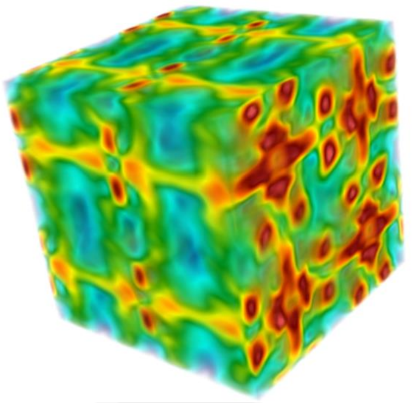
https://developer.nvidia.com/blog/develop-physics-informed-machine-learning-models-with-graph-neural-networks/

GenCFD

Generative AI for Statistical Computation of Fluids





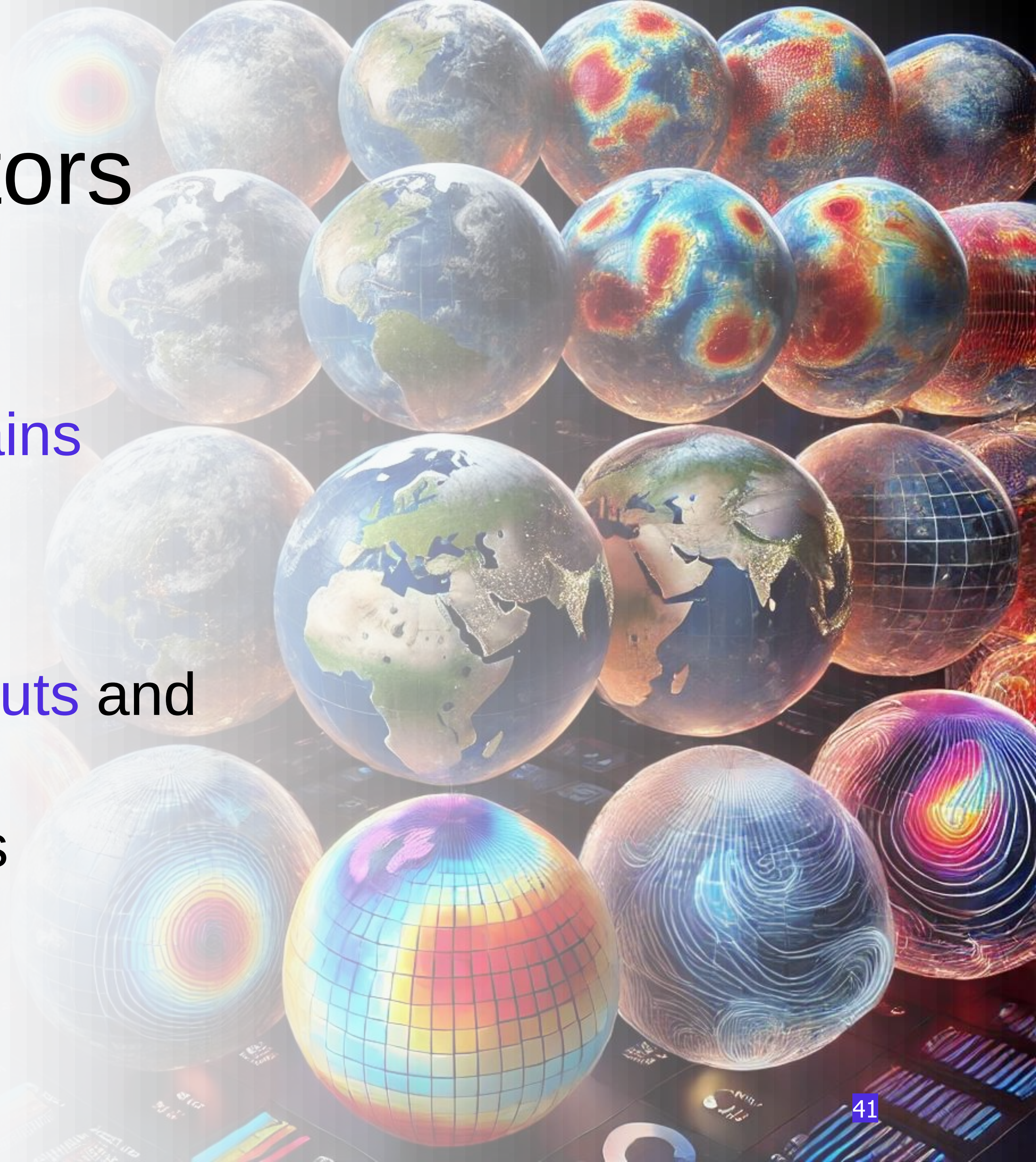


https://camlab.ethz.ch/ https://arxiv.org/pdf/2409.18359

Limitations of Operators

- They need lots of training data
- Generalization to unseen domains
- Computational cost of training
- Limited Interpretability
- Handling High-Dimensional Inputs and Outputs
- Scalability to Complex Systems

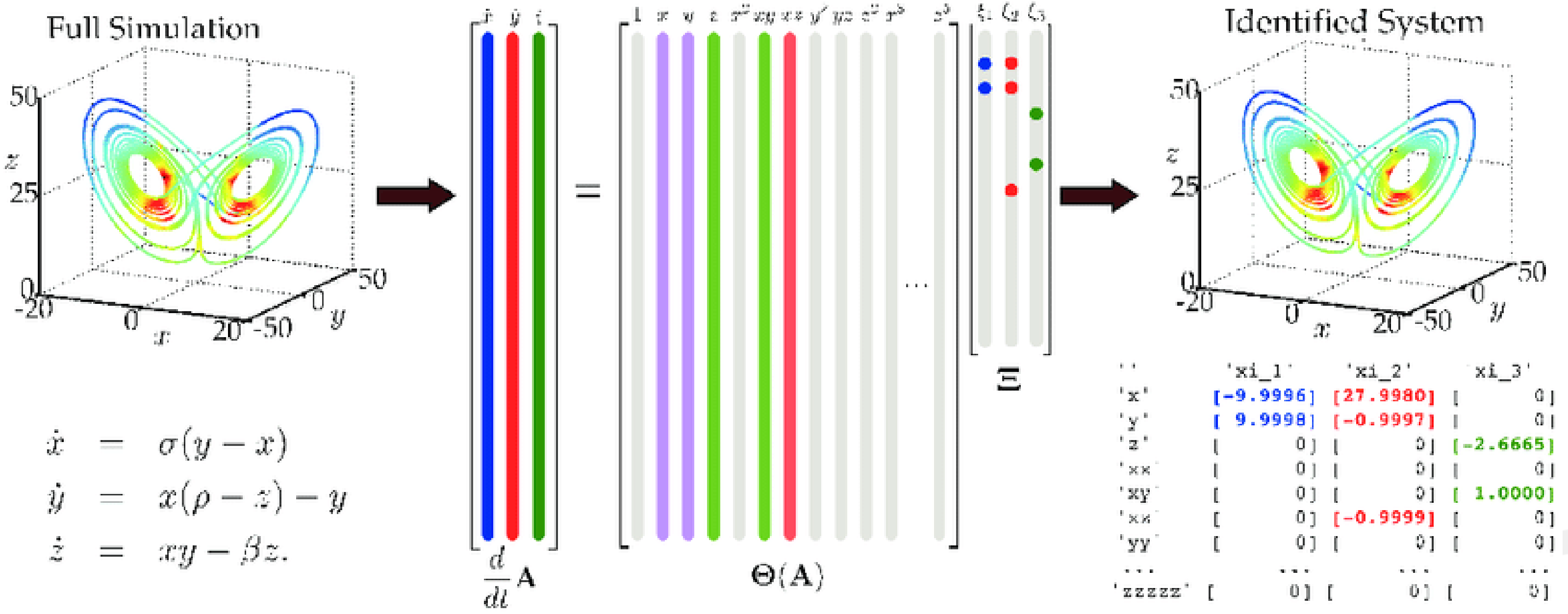
They are still under development....



A short break ☺

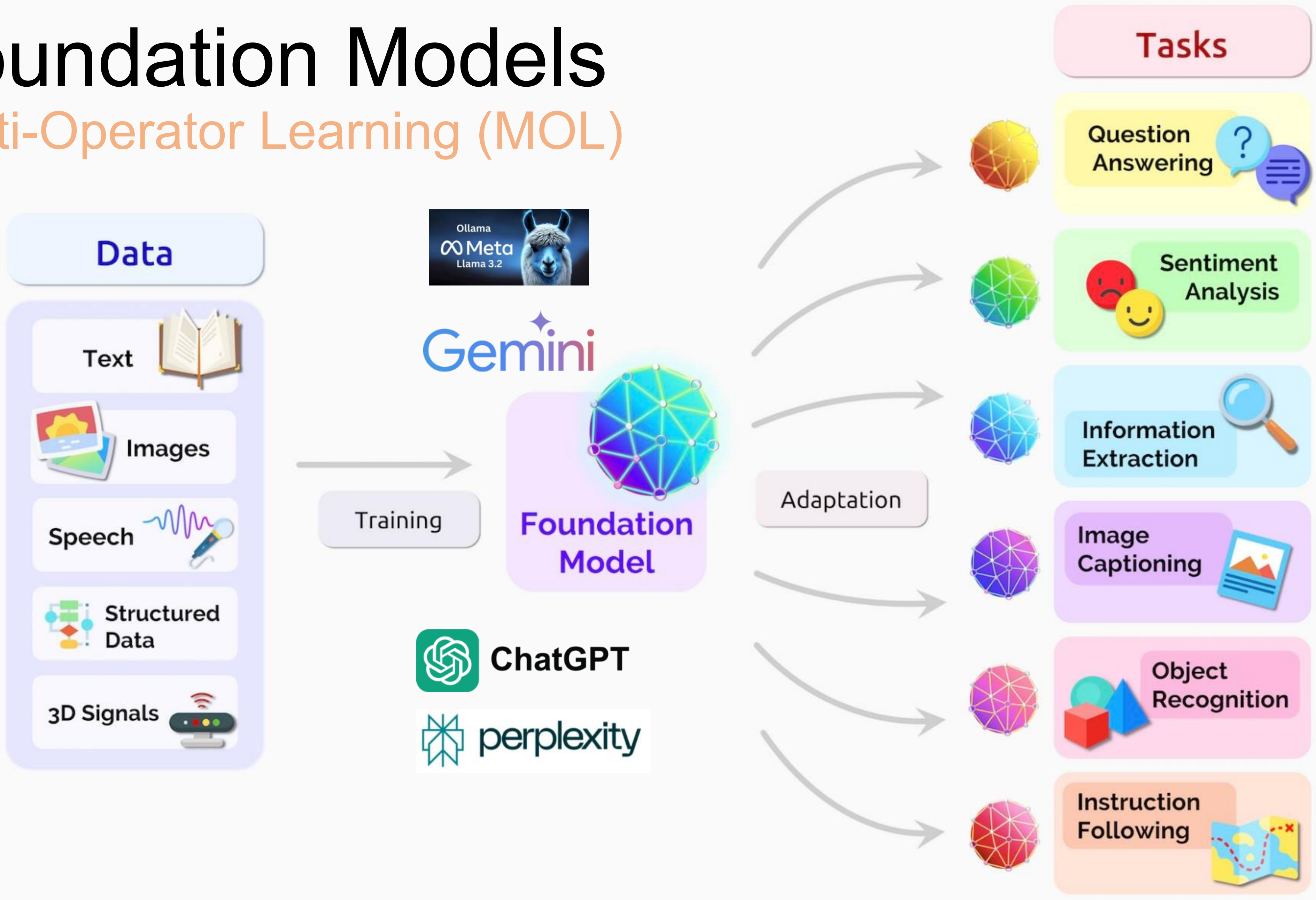
Sparse Identification of Nonlinear Dynamics (SINDy)

- **Purpose:** SINDy identifies governing equations of a dynamical system directly from data.
- **Key Idea:** It combines sparse regression techniques with a library of candidate functions to discover concise, interpretable models.



Foundation Models

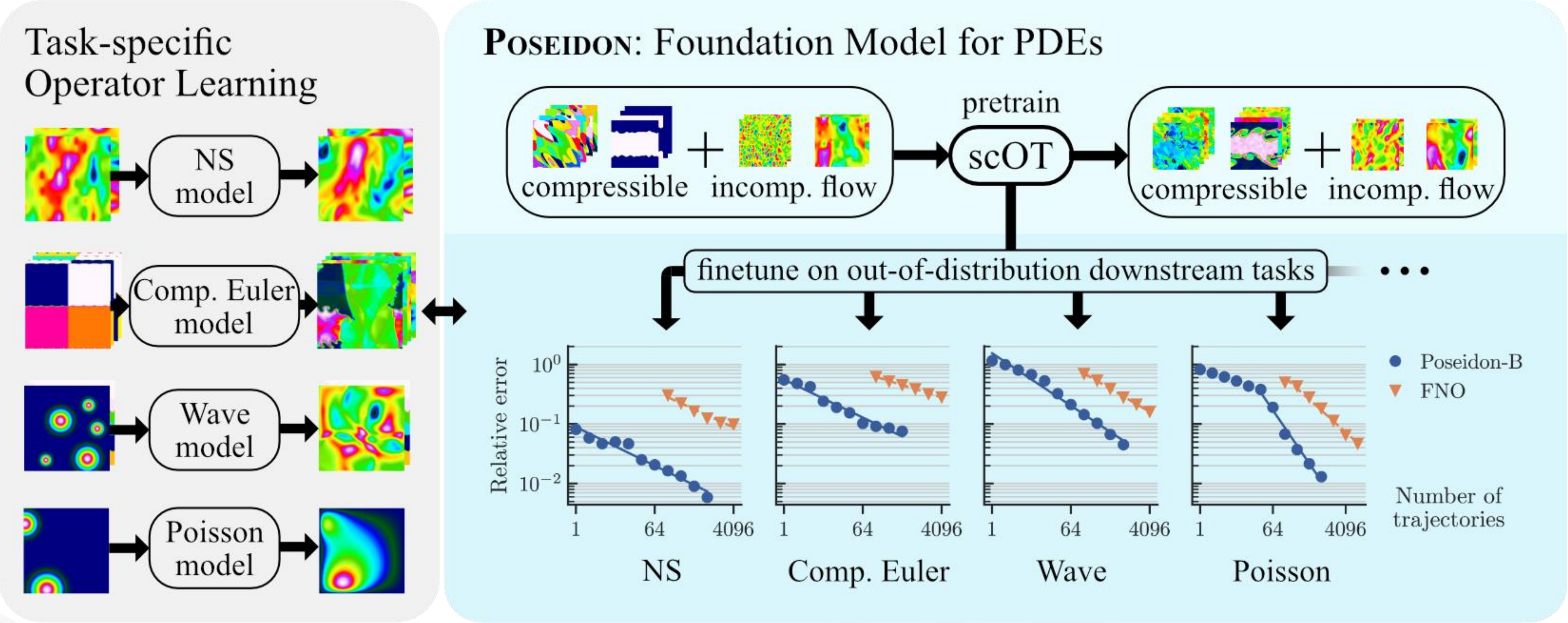
Multi-Operator Learning (MOL)



[The Foundation Model: Key Facts and Insights | BasicAI's Blog]

Foundation Models for PDEs

Poseidon: Efficient Foundation Models for PDEs (2024)

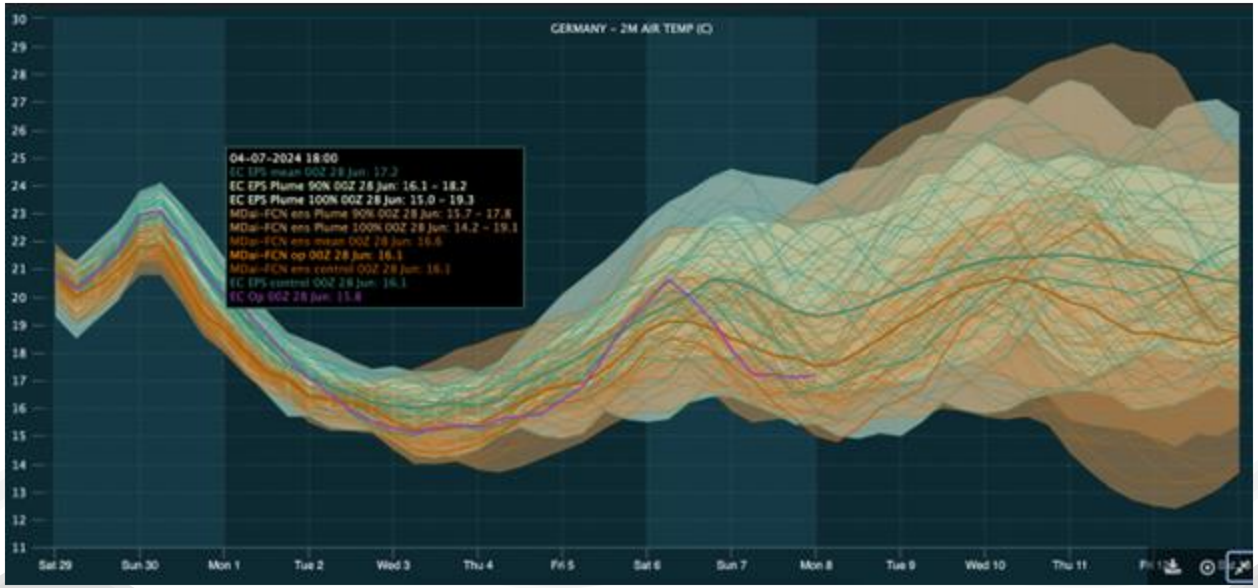


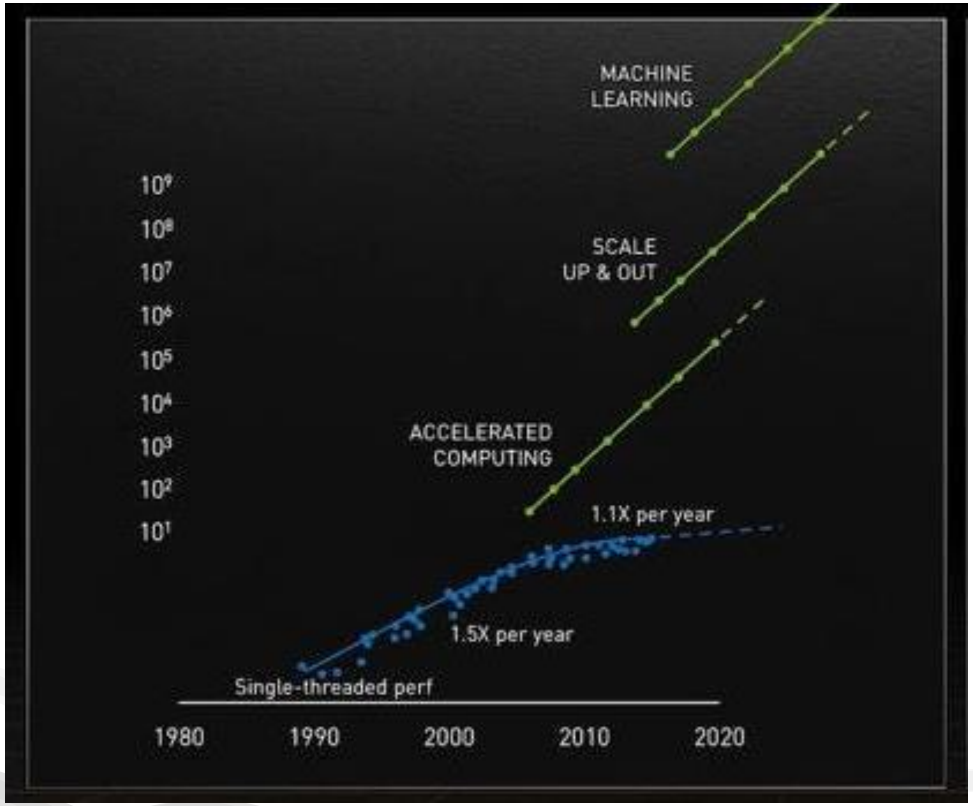
https://github.com/camlab-ethz/poseidon

Examples of Real World Applications

Caltech: Predicting Extreme Weather Predictions

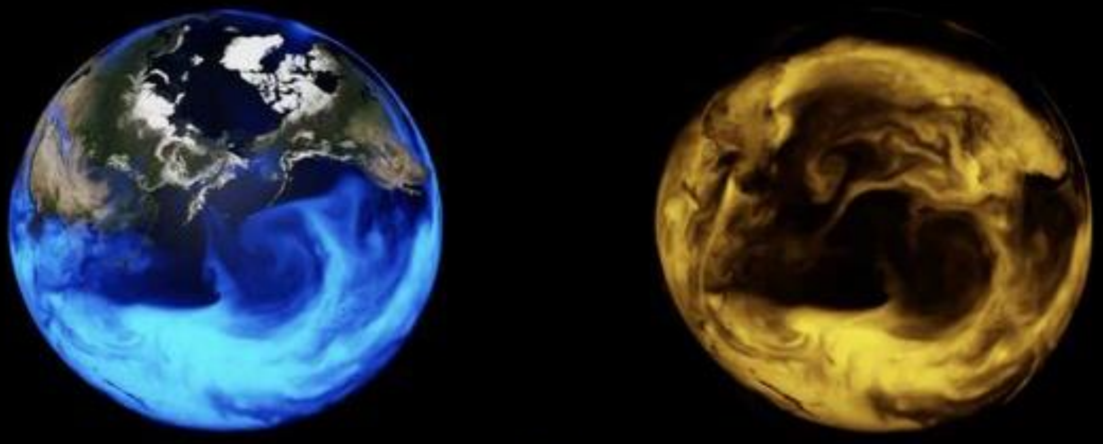
FourCastNet: Fourier ForeCasting Neural Network

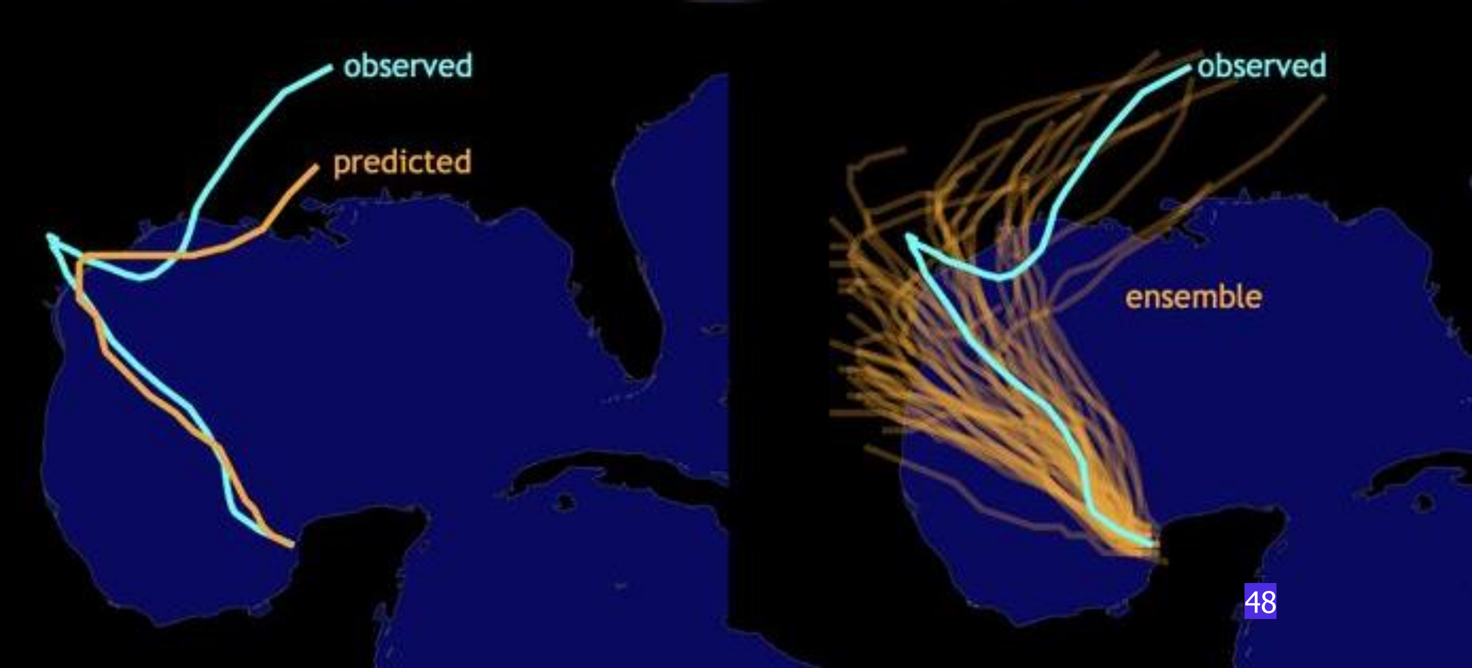




Caltech: Predicting Extreme Weather Predictions

- The speedup allows weather predictions to be obtained in a matter of seconds vs. traditional weather models taking hours on a large CPU cluster (45,000 times faster).
- What matters is not just a single run of a forecast but rather a large number of slightly different scenarios and seeing how the trajectories evolve over time.
- In such cases, deterministic forecasts (the plot on the left) are just not sufficient.

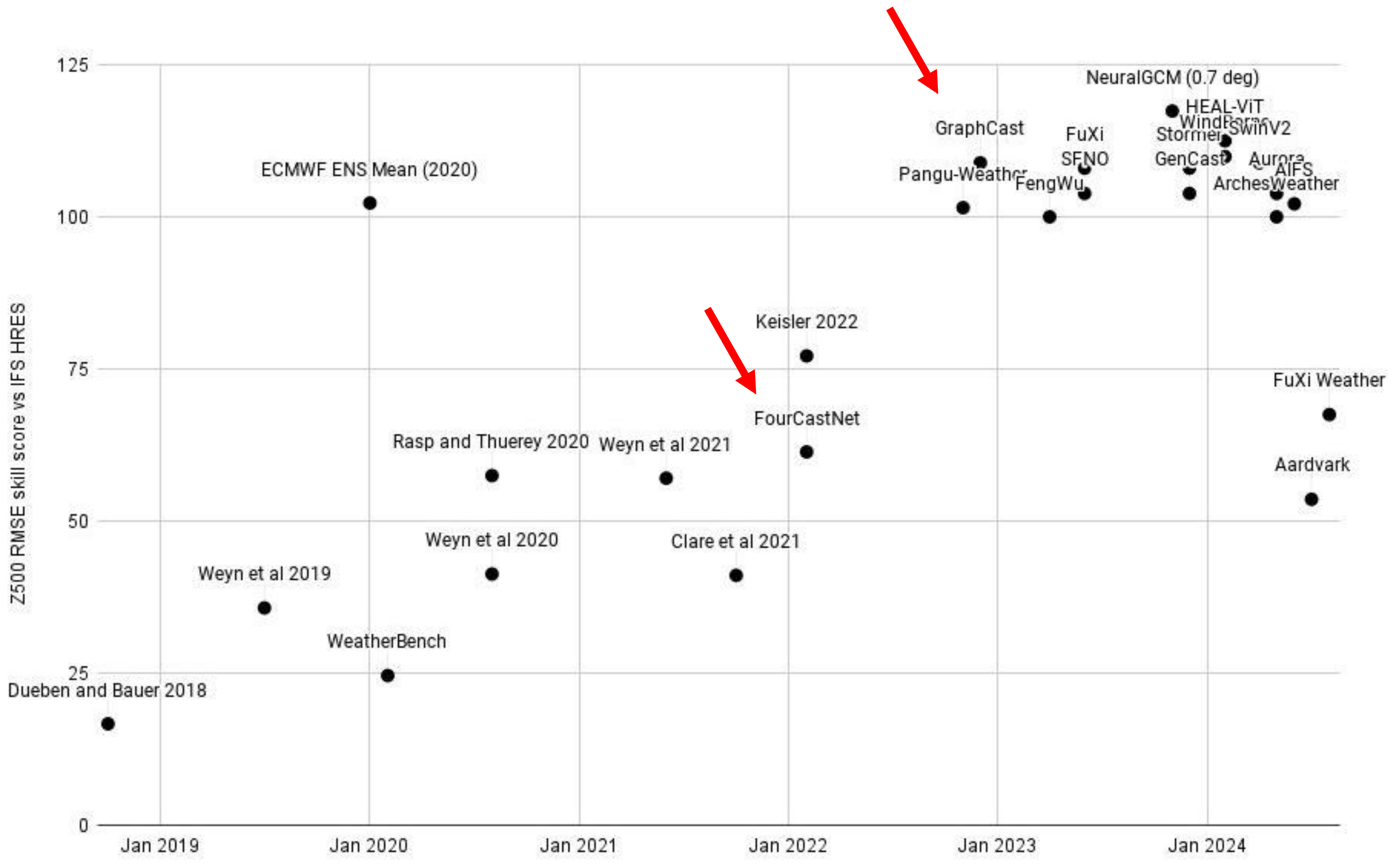




Anima Anandkumar, FourCastNet

NVIDIA Caltech

Progress of AI-weather models over time



https://docs.google.com/spreadsheets/d/1n30zDDjEzlXl5nAGF8uD_dbZWJAamqImQGCZjfOMuDg/edit?gid=0#gid=0

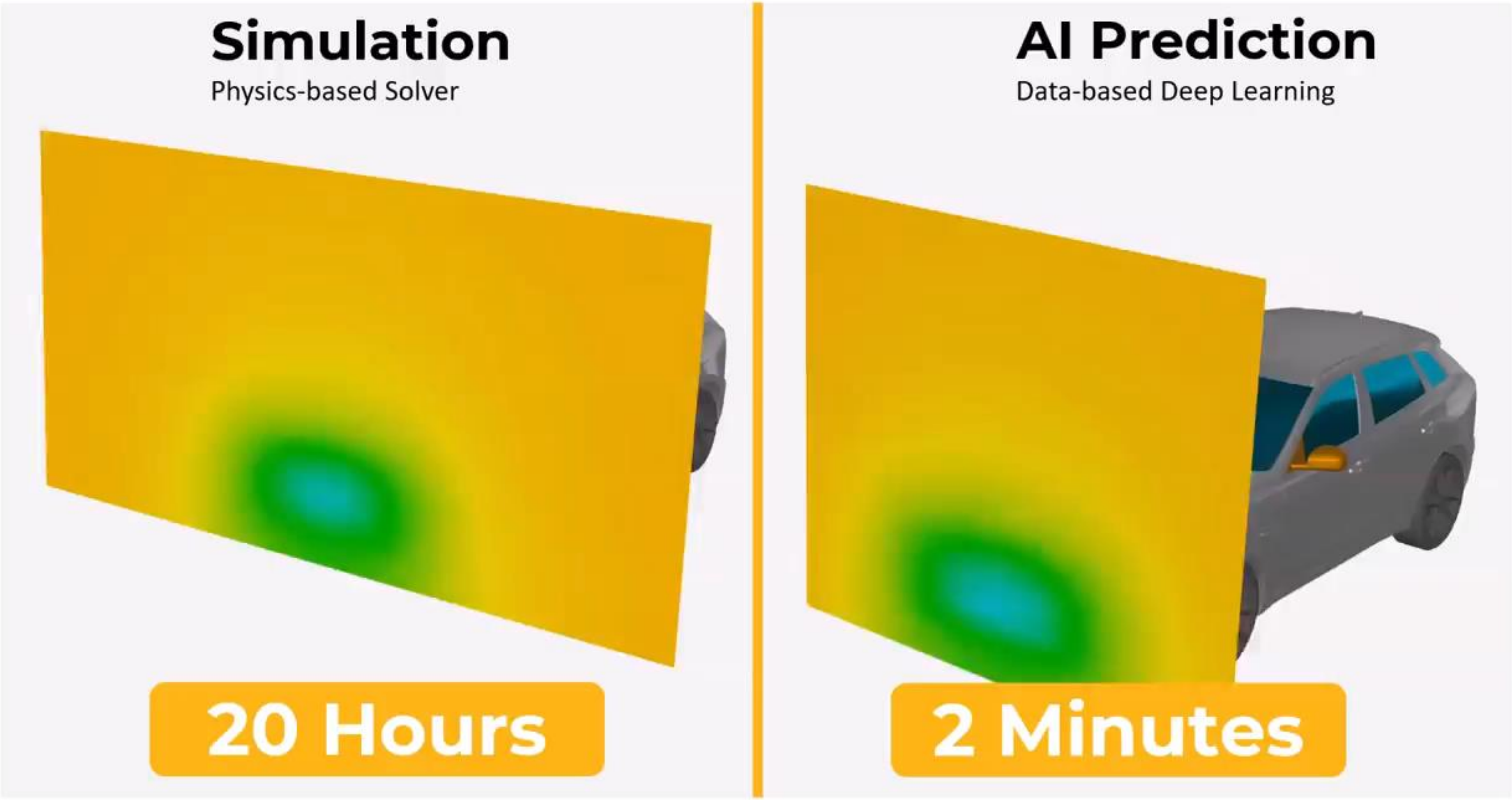
Author: Stephan Rasp (srasp@google.com)

Computational Costs

- FourCastNet's training utilizes 3,072 A100 GPUs
- 40GB PCIe Version: ~ $7,400.
- 80GB PCIe Version: ~ $24,300.
- Total: 23-76 M$
- Training time is 67.4 minutes

Ansys SimAI vs. Fluent

- Ansys #SimAI Prediction on new SUV geometry takes less than 1 min.
- SimAI Drag error compared to CFD is less than 0.5% (5 to 10 drag counts) and accurate skin friction field and wake topology prediction.

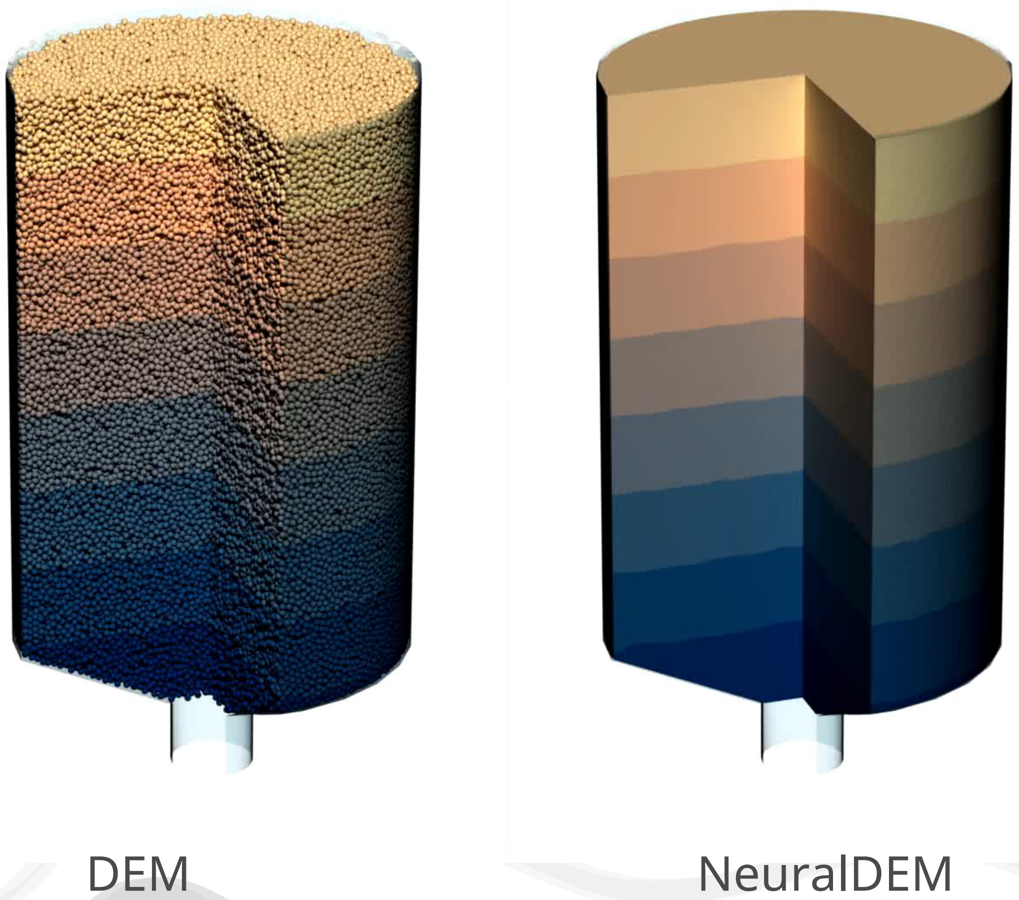


https://jousef.substack.com/archive?sort=new

NeuralDEM

Real-time Simulation of Particulate Flows

- NeuralDEM presents an end-to-end approach to replace Discrete Element Method (DEM) routines and coupled multiphysics simulations with deep learning surrogates.

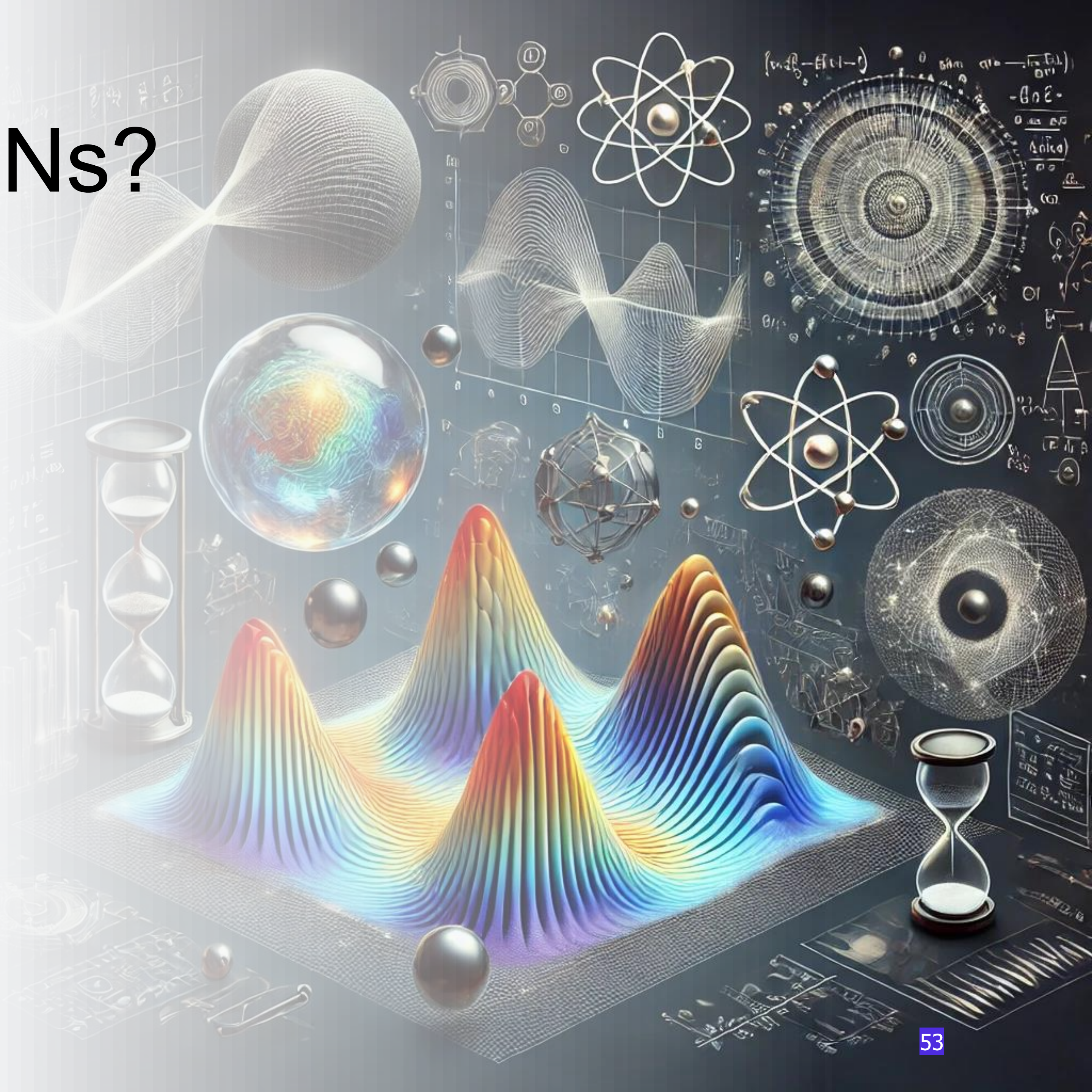


Mass flow

https://nx-ai.github.io/NeuralDEM/

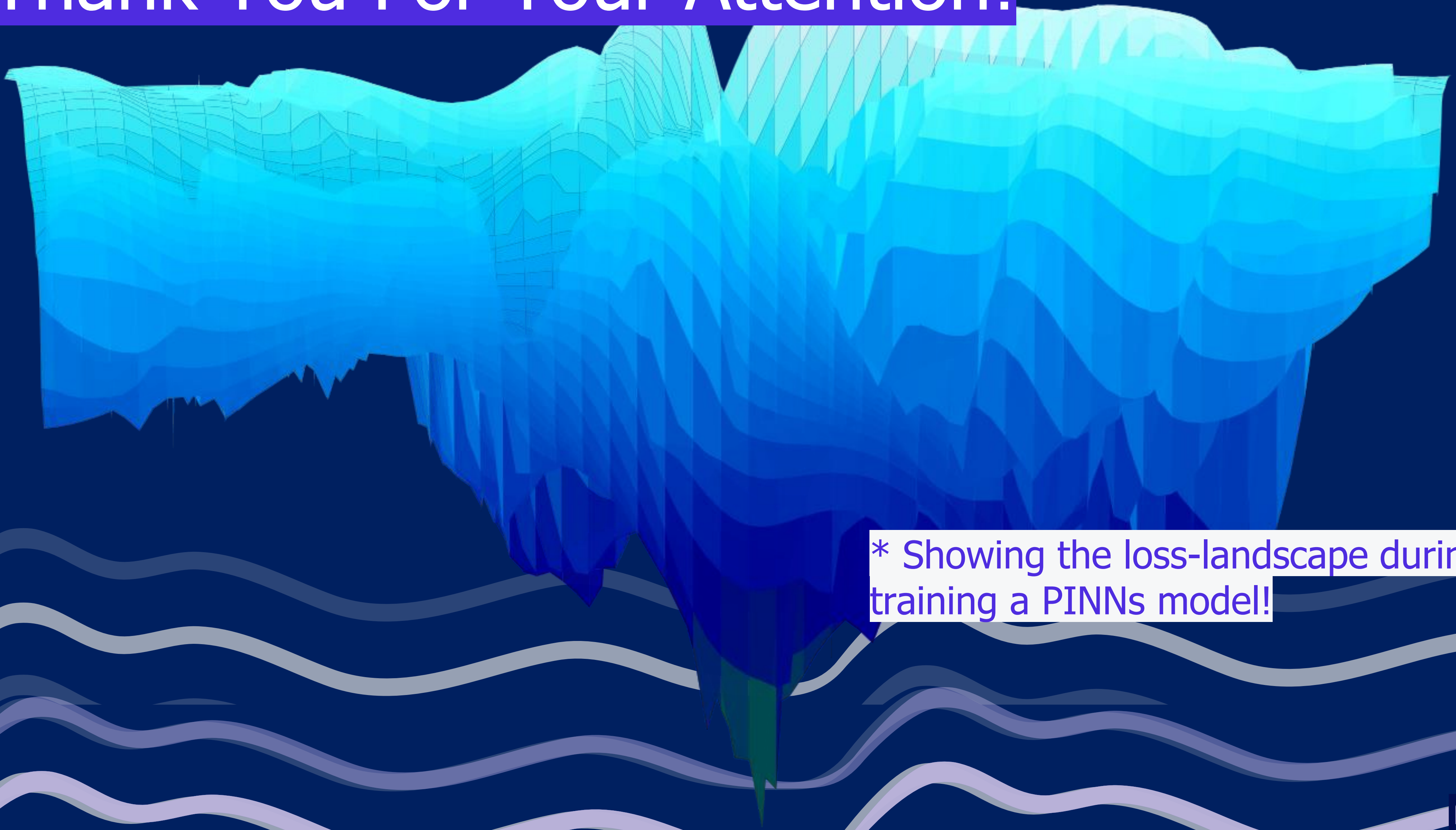
Then, what is PINNs?

- Next Session ☺



Before that,
We need a real break!

Thank You For Your Attention!



* Showing the loss-landscape during training a PINNs model!