Scientific machine learning: ways to incorporate scientific principles into ML

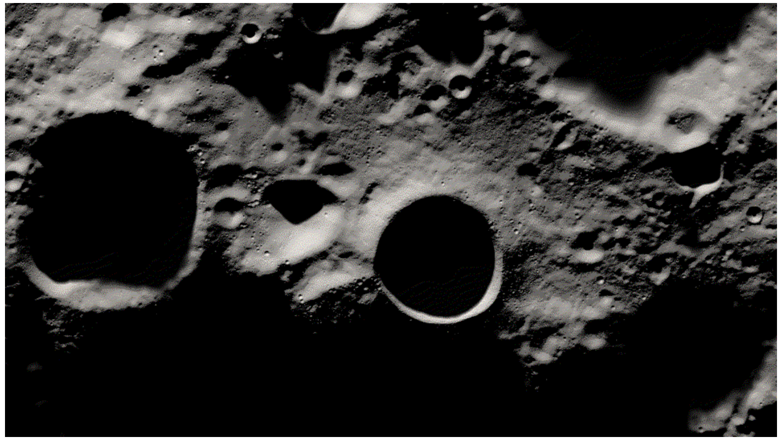
Ben Moseley
Postdoctoral fellow, ETH AI Center /
ETH Computational and Applied Mathematics Laboratory

Talk overview

- What is scientific machine learning (SciML)?

- Ways to incorporate scientific principles into ML

- Our research: scaling SciML techniques to complex, real-world problems



Credits: NASA Scientific Visualization Studio / QuickMap / LROC

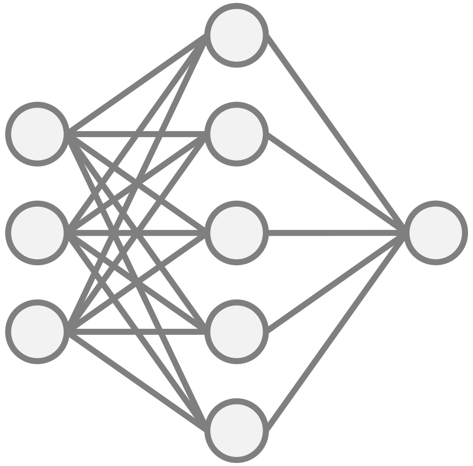
Later in the talk: peering into shadows on the Moon using SciML

Talk overview

- What is scientific machine learning (SciML)?
- Ways to incorporate scientific principles into ML
- Our research: scaling SciML techniques to complex, real-world problems

What is deep learning?

Trained using:
- (Stochastic) gradient descent
- An appropriate loss function
- Many (millions of) training examples



Probability = Dog

Input

Model

Output

x

$\mathrm{NN}(\mathrm{x}, \theta)$

$\mathrm{y} = \mathrm{NN}(\mathrm{x}, \theta)$

For example:

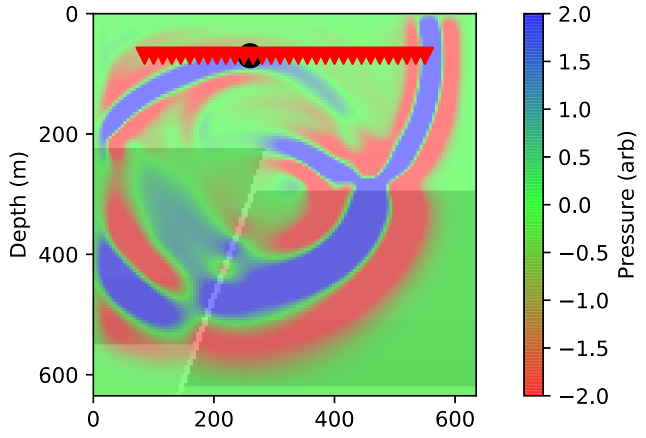
$$\mathrm{y} = \mathrm{W}_2 \sigma(\mathrm{W}_1 \mathrm{x} + \mathrm{b}_1) + \mathrm{b}_2$$

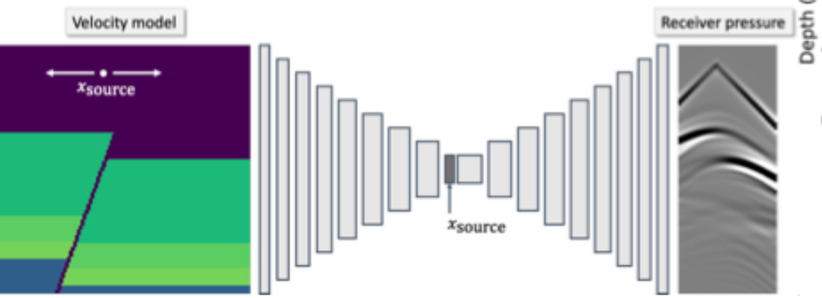
The challenge of generalisation

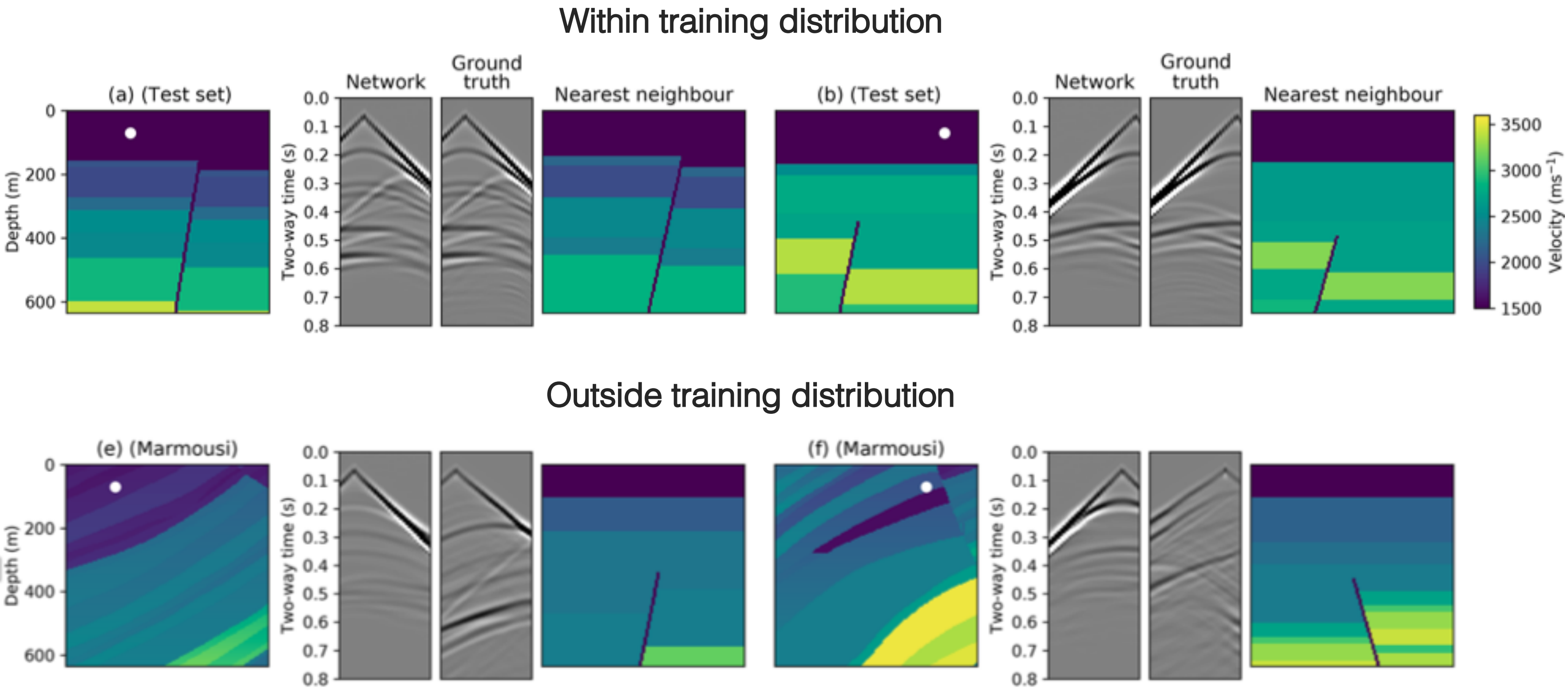
Labrador retriever 52%
Chesapeake Bay retriever 7%
golden retriever 5%
Canis dingo 4%
bloodhound, sleuthhound 3%

laboratory coat 40%
jeweler's loupe 8%
English foxhound 6%
soccer ball 4%
neck brace 3%

“Naive” application of ML







Moseley, B., Nissen-Meyer, T., & Markham, A. (2020). Deep learning for fast simulation of seismic waves in complex media. Solid Earth

Scientific machine learning (SciML)

Major problem

Despite big breakthroughs in science + AI

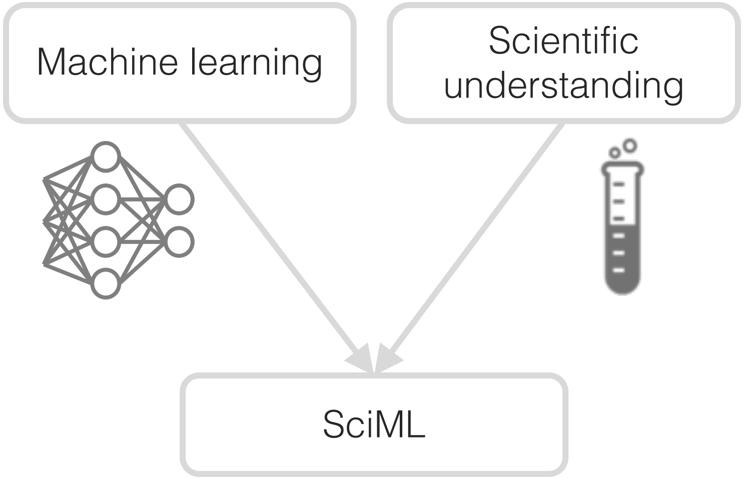
Naively using deep learning for scientific tasks usually leads to:

- Lack of interpretability
- Poor generalization
- Lots of training data required

Do neural networks really "**understand**" the scientific tasks they are being applied to?

A good scientific theory = makes novel predictions

Solution



more **powerful, robust, interpretable** models

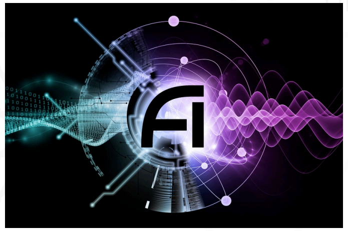
Scientific machine learning (SciML)

Physics-informed neural networks: # citations
(reproduced from Cuomo et al, 2022)

National Science Foundation announces MIT-led Institute for Artificial Intelligence and Fundamental Interactions

IAIFI will advance physics knowledge — from the smallest building blocks of nature to the largest structures in the universe — and galvanize AI research innovation.

Laboratory for Nuclear Science
August 26, 2020



The U.S. National Science Foundation (NSF) announced today an investment of more than $100 million to establish five artificial intelligence (AI) institutes, each receiving roughly $20 million over five years. One of these, the NSF AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI), will be led by MIT's Laboratory for Nuclear Science (LNS) and become the intellectual home of more than 25 physics and AI senior researchers at MIT and Harvard, Northeastern, and Tufts universities.

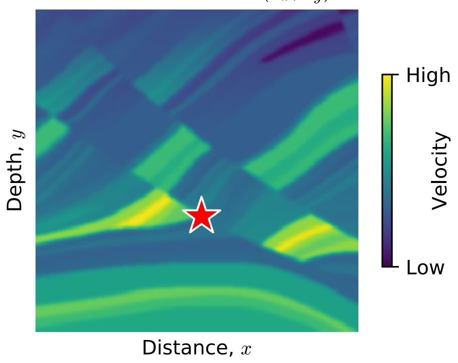
Baker, N. et al (2019). Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence.

Talk overview

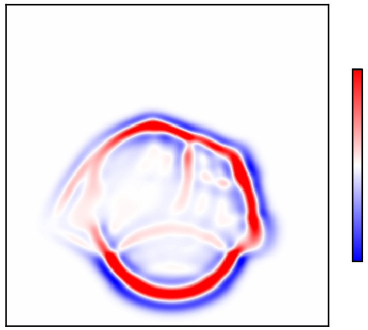
- What is scientific machine learning (SciML)?
- Ways to incorporate scientific principles into ML
- Our research: scaling SciML techniques to complex, real-world problems

Typical SciML tasks

Velocity, $c(x, y)$
Source location, (s_x, s_y)



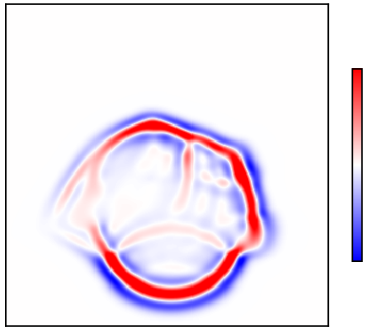
Wavefield, $u(x, y, t)$



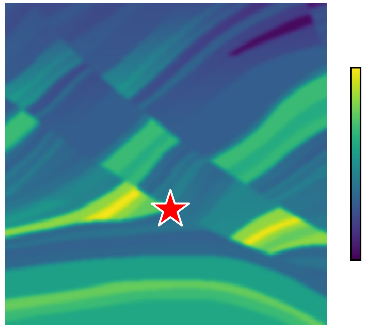
Forward simulation

Estimate wavefield $u(x, y, t)$
Given velocity $c(x, y)$
and source location (s_x, s_y)

Wavefield, $u(x, y, t)$



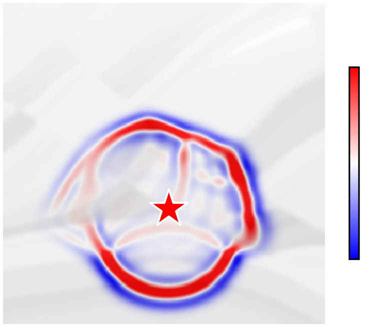
Velocity, $c(x, y)$
Source location, (s_x, s_y)



Inversion

Estimate velocity $c(x, y)$
and source location (s_x, s_y)
Given wavefield $u(x, y, t)$

Wavefield, $u(x, y, t)$
Velocity, $c(x, y)$
Source location, (s_x, s_y)



Wave equation

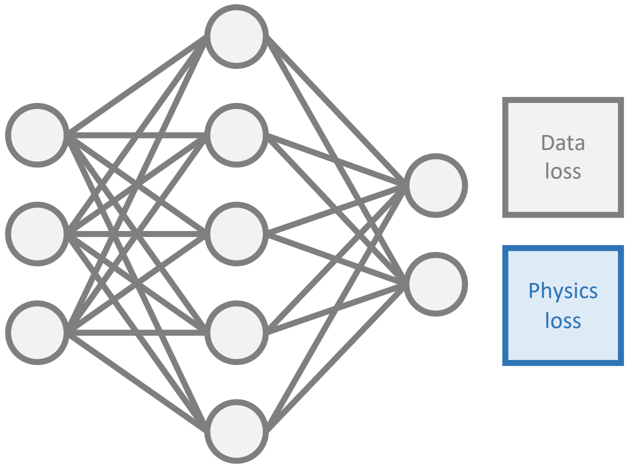
$$\nabla^2 u - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = f$$

Equation discovery

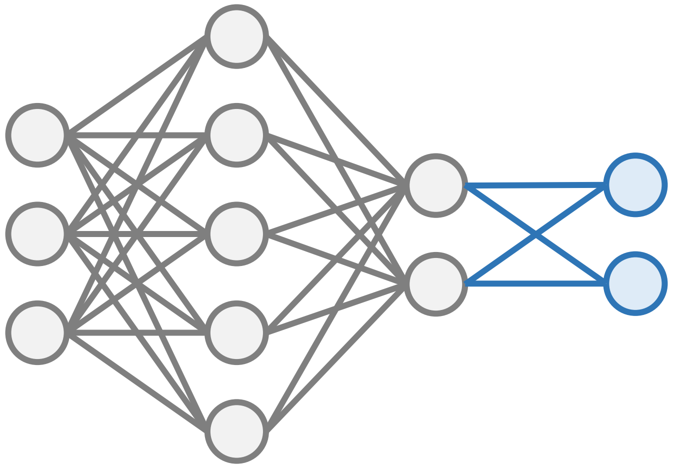
Estimate governing equation
Given wavefield $u(x, y, t)$,
velocity $c(x, y)$,
and source location (s_x, s_y)

Ways to incorporate scientific principles into machine learning

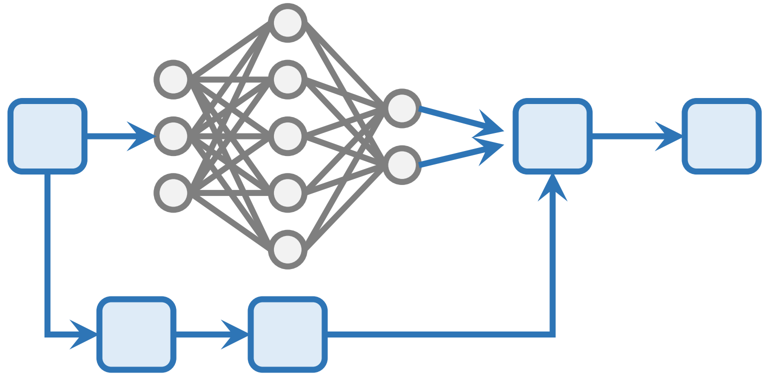
Loss function



Architecture



Hybrid approaches



	Forward simulation	Inversion	Equation discovery
Architecture			
Adding physical variables	Daw et al.		
Encoding symmetries	Ling et al., Wang et al., Anderson et al., Schütt et al.		Udrescu et al.
Physics-inspired NAS	Ba et al., Panju and Ghodsi		
ML inspired by Koopman theory	Geneva and Zabaras, Lusch et al.		
Physically constrained GPs		Raissi et al., Raissi and Karniadakis	
Other approaches	Jumper et al., Mohan et al.		
Loss function			
Constraining physical quantities	Karpatne et al., Zhang et al., Benjamin Erichson et al., Xie et al., Brehmer et al.		
Encoding conservation laws	Beucler et al., Zeng et al.		Greydanus et al., Toth et al., Cranmer et al.
Auxiliary tasks	de Oliveira et al.		
Encoding governing equations	Raissi et al., Jin et al., Kharazmi et al., Yang et al., Wang et al., Wang et al., Li et al., Zhu et al., Geneva and Zabaras, Gao et al.	Jin et al., Chen et al., Yang et al.	Chen et al., Champion et al.
Hybrid approaches			
Residual modelling	Pawar et al.	Jiang et al.	
Differentiable physics		Ren et al., Minkov et al., Würfl et al., Zhang et al.	
Neural differential equations			Chen et al., Rackauckas et al., Long et al.
In-the-loop methods	Um et al., Rasp et al.	Adler and Öktem, Morningstar et al., Hammernik et al., Li et al., Lunz et al., Bora et al., Mosser et al.	

AI Feynman

Covariant neural networks

Hidden physics models

AlphaFold

“Mining gold” from implicit models

Hamiltonian/ Lagrangian neural networks

Physics-informed neural networks

Physics-informed Fourier neural operators

Physics-informed DeepONets

PINNs on meshes

Algorithm unrolling

Universal differential equations

Learned sub-grid processes

”Solver-in-the-loop”

Learned gradient descent

Adversarial regularisers

Source: My PhD thesis: Physics-informed machine learning: from concepts to real-world applications read it here: tinyurl.com/mw39wdps

A wide plethora of techniques

Naive ML

Constraining physical quantities
Encoding conservation laws
Auxiliary tasks
Encoding governing equations

Loss function

Residual modelling
Differentiable physics
Neural differential equations
In-the-loop methods

Hybrid approaches

Adding physical variables
Encoding symmetries
Physics-inspired NAS
ML inspired by Koopman theory
Physically constrained GPs

Architecture

Traditional workflows

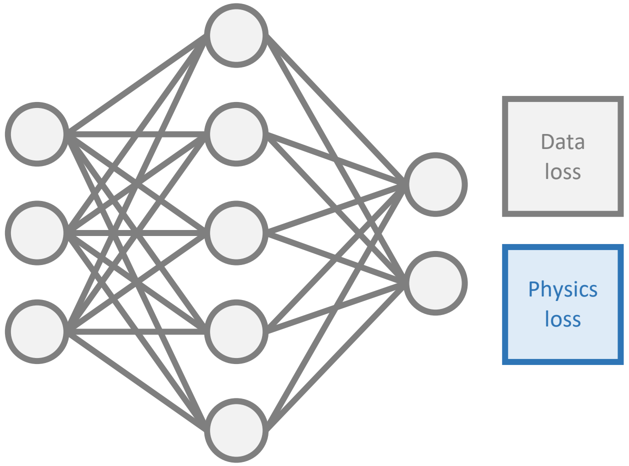
Source: My PhD thesis: Physics-informed machine learning: from concepts to real-world applications read it here: tinyurl.com/mw39wdps

	Forward simulation	Inversion	Equation discovery
Architecture			
Adding physical variables	Daw et al.		
Encoding symmetries	Ling et al., Wang et al., Anderson et al., Schütt et al.		Udrescu et al.
Physics-inspired NAS	Ba et al., Panju and Ghodsi		
ML inspired by Koopman theory	Geneva and Zabaras, Lusch et al.		
Physically constrained GPs		Raissi et al., Raissi and Karniadakis	
Other approaches	Jumper et al., Mohan et al.		
Loss function			
Constraining physical quantities	Karpatne et al., Zhang et al., Benjamin Erichson et al., Xie et al., Brehmer et al.		
Encoding conservation laws	Beucler et al., Zeng et al.		Greydanus et al., Toth et al., Cranmer et al.
Auxiliary tasks	de Oliveira et al.		
Encoding governing equations	Raissi et al., Jin et al., Kharazmi et al., Yang et al., Wang et al., Wang et al., Li et al., Zhu et al., Geneva and Zabaras, Gao et al.	Jin et al., Chen et al., Yang et al.	Chen et al., Champion et al.
Hybrid approaches			
Residual modelling	Pawar et al.	Jiang et al.	
Differentiable physics		Ren et al., Minkov et al., Würfl et al., Zhang et al.	
Neural differential equations			Chen et al., Rackauckas et al., Long et al.
In-the-loop methods	Um et al., Rasp et al.	Adler and Öktem, Morningstar et al., Hammernik et al., Li et al., Lunz et al., Bora et al., Mosser et al.	

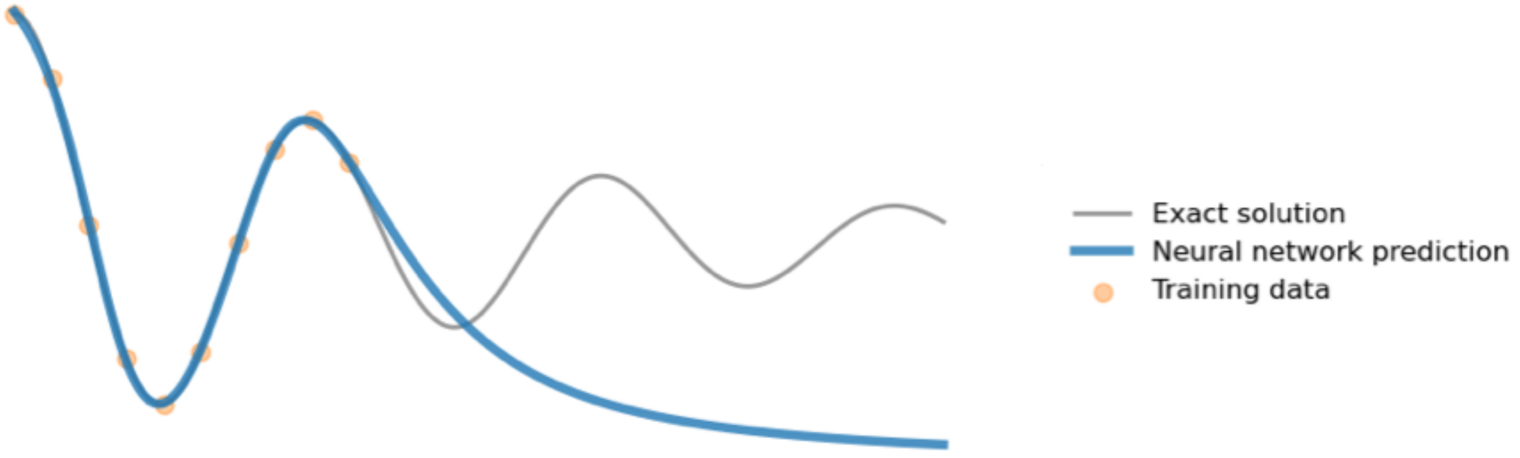
Source: My PhD thesis: Physics-informed machine learning: from concepts to real-world applications read it here: tinyurl.com/mw39wdps

Loss function:

Physics-informed neural networks



“Naive” neural network

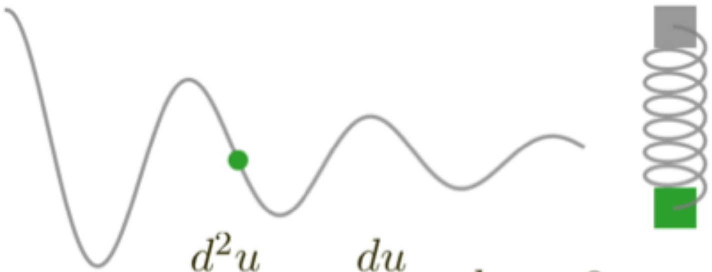


$$\min_{\theta} \frac{1}{N} \sum_{i}^{N} (u_{\mathrm{NN}}(x_i;\theta) - \underline{u_{\mathrm{true}}(x_i)})^2$$

Source: ”So, what is a physics-informed neural network?” (**benmoseley.blog/**)

“Naive” neural network

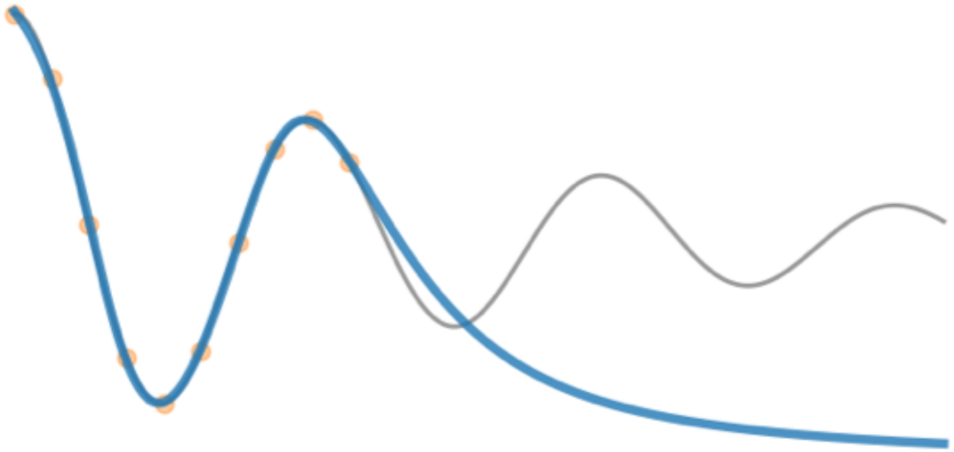
$$\min_{\theta} \frac{1}{N}\sum_{i}^{N}\left(u_{\mathrm{NN}}(x_i;\theta) - \underline{u_{\mathrm{true}}(x_i)}\right)^2$$



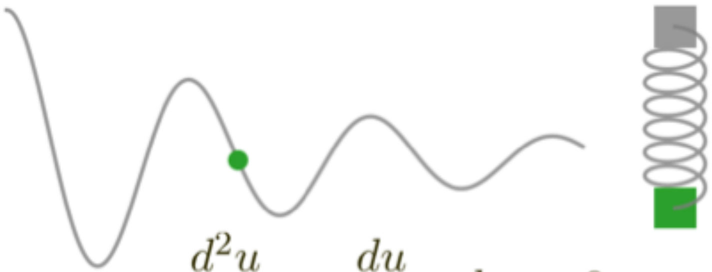
$$m\frac{d^2u}{dx^2} + \mu\frac{du}{dx} + ku = 0$$

Source: ”So, what is a physics-informed neural network?” (**benmoseley.blog/**)

“Naive” neural network

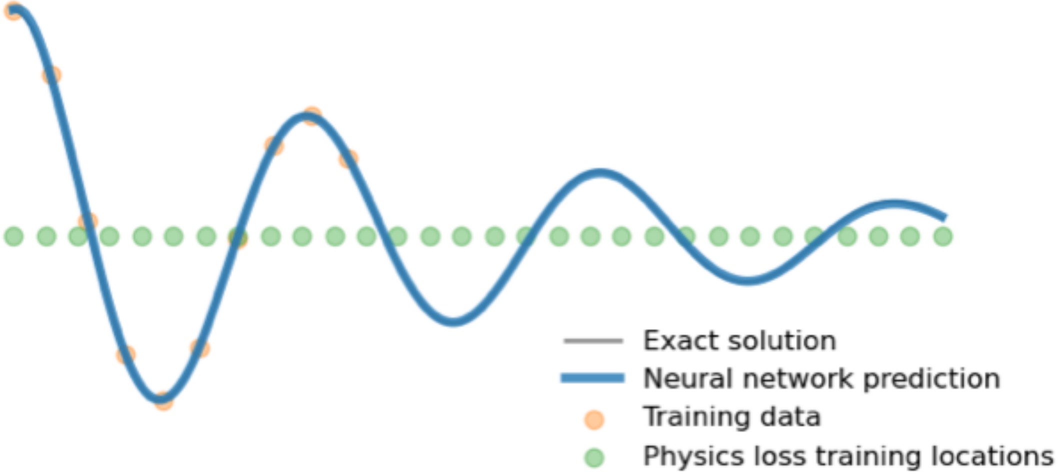


$$\min_{\theta} \frac{1}{N}\sum_{i}^{N}\left(u_{\mathrm{NN}}(x_i;\theta) - \underline{u_{\mathrm{true}}(x_i)}\right)^2$$



$$m\frac{d^2u}{dx^2} + \mu\frac{du}{dx} + ku = 0$$

Physics-informed neural network



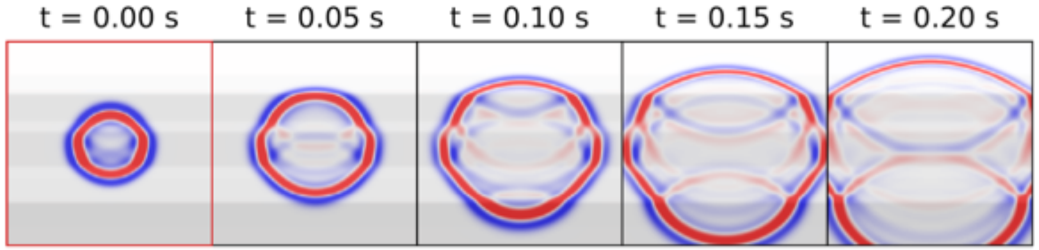
$$\min_{\theta} \frac{1}{N}\sum_{i}^{N}\left(u_{\mathrm{NN}}(x_i;\theta) - \underline{u_{\mathrm{true}}(x_i)}\right)^2$$
$$+\frac{1}{M}\sum_{j}^{M}\left(\left[m\frac{d^2}{dx^2} + \mu\frac{d}{dx} + k\right]u_{\mathrm{NN}}(\underline{x_j};\theta)\right)^2$$

Source: ”So, what is a physics-informed neural network?” (**benmoseley.blog/**)

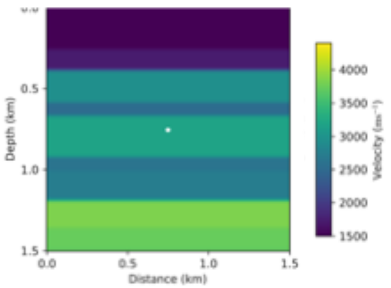
Raissi et al (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.
Lagaris et al (1998). Artificial neural networks for solving ordinary and partial differential equations.

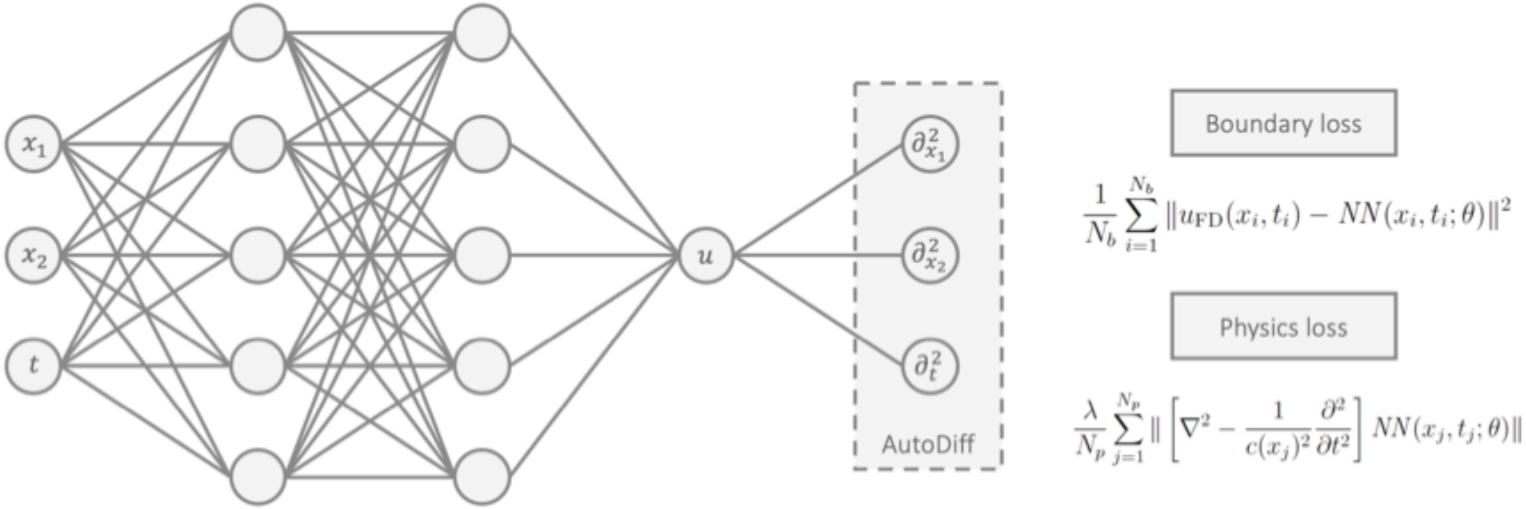
PINNs for simulation

Ground truth FD



Velocity model





$$\mathcal{L}(\theta) = \frac{1}{N_b}\sum_{i=1}^{N_b} \|u_{\mathrm{FD}}(x_i,t_i) - NN(x_i,t_i;\theta)\|^2 + \frac{\lambda}{N_p}\sum_{j=1}^{N_p} \| \left[\nabla^2 - \frac{1}{c(x_j)^2}\frac{\partial^2}{\partial t^2}\right] NN(x_j,t_j;\theta)\|$$

Moseley, B., Markham, A., & Nissen-Meyer, T. (2020). Solving the wave equation with physics-informed deep learning. ArXiv

PINNs for simulation

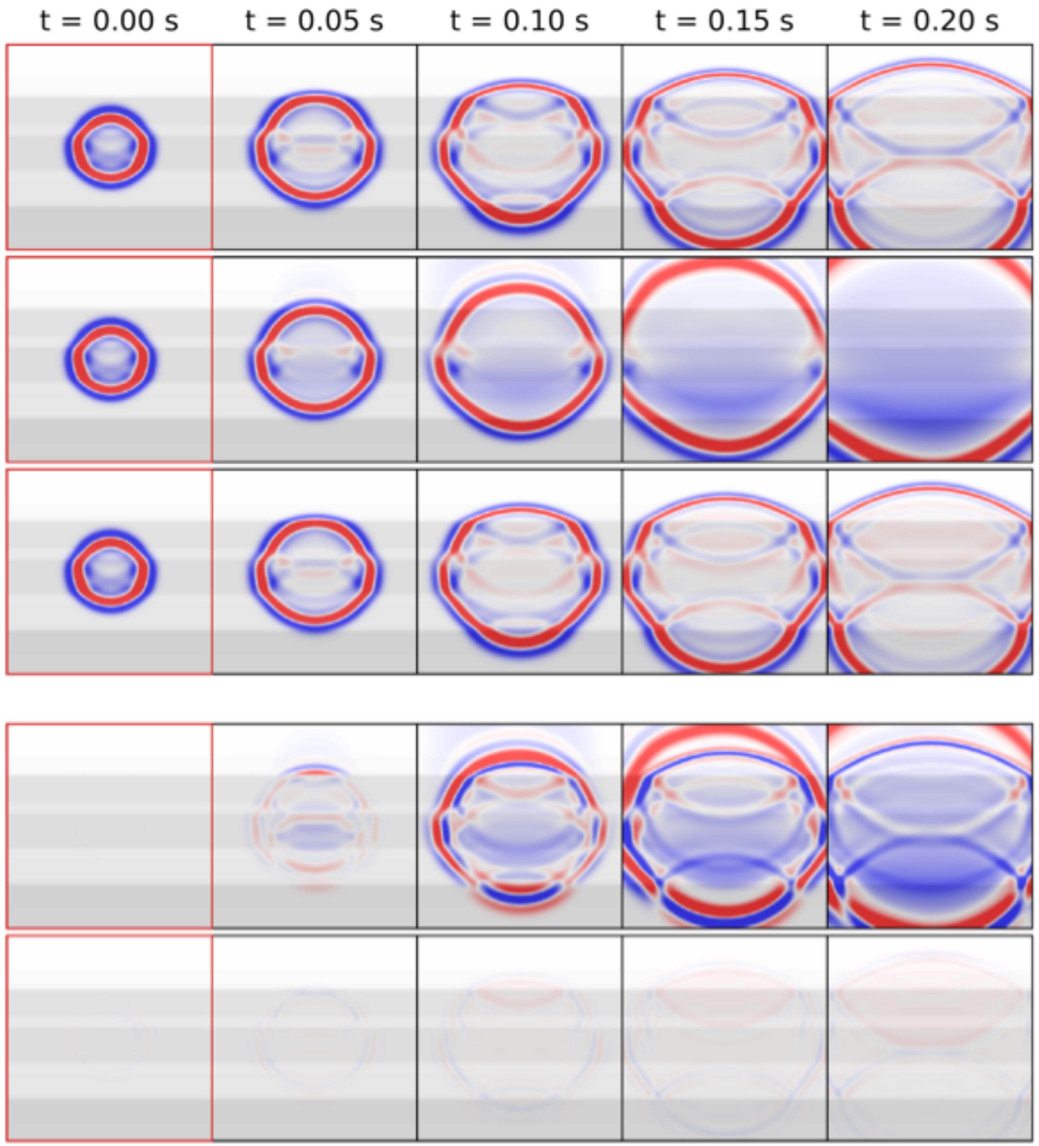
Ground truth FD

"Naïve" NN

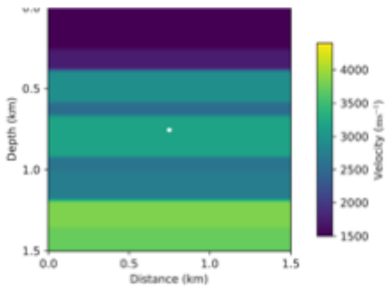
PINN

Difference (NN)

Difference (PINN)



Velocity model



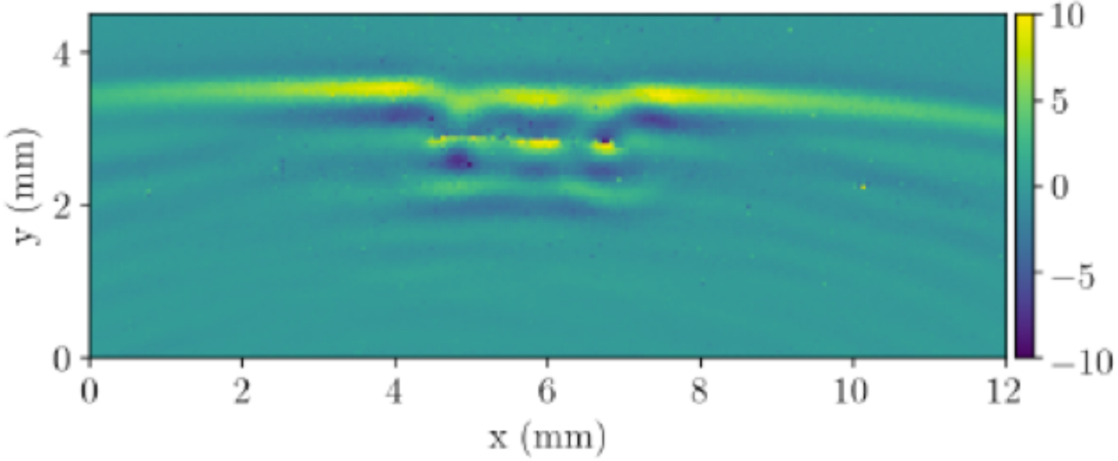
NN architecture:
Fully-connected, 10 layers, 512 hidden channels

Training time: ~3 hours

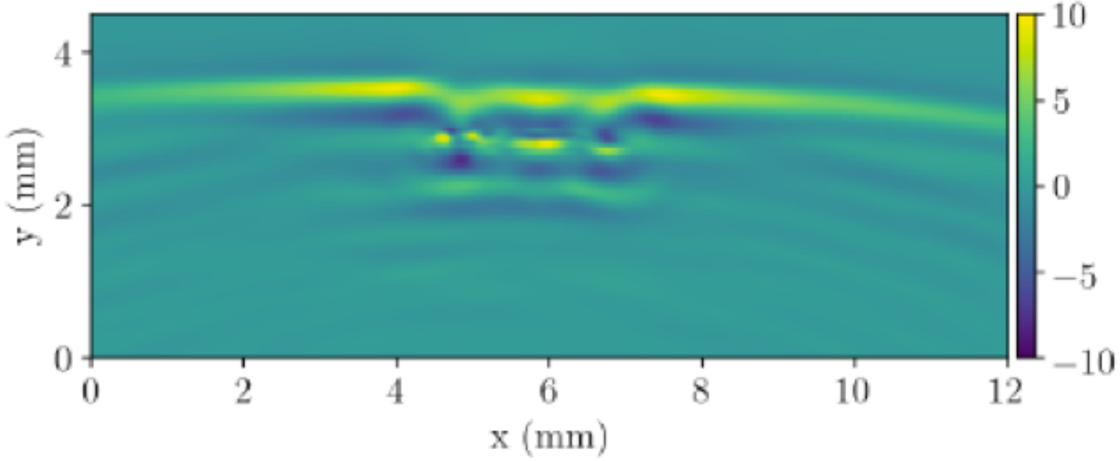
Moseley, B., Markham, A., & Nissen-Meyer, T. (2020). Solving the wave equation with physics-informed deep learning. ArXiv

PINNs for inversion

Shukla K et al, Physics-Informed Neural Network for Ultrasound Nondestructive Quantification of Surface Breaking Cracks, Journal of Nondestructive Evaluation (2020)

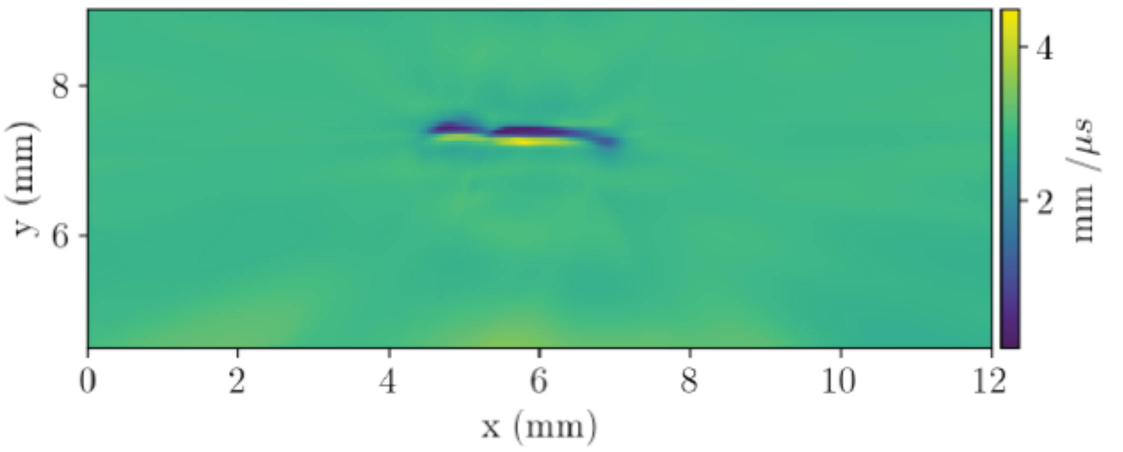


(a) Actual data at $t = 12.38\ \mu s$.



(b) Data recovered from PINN simulation at $t = 12.38\ \mu s$.

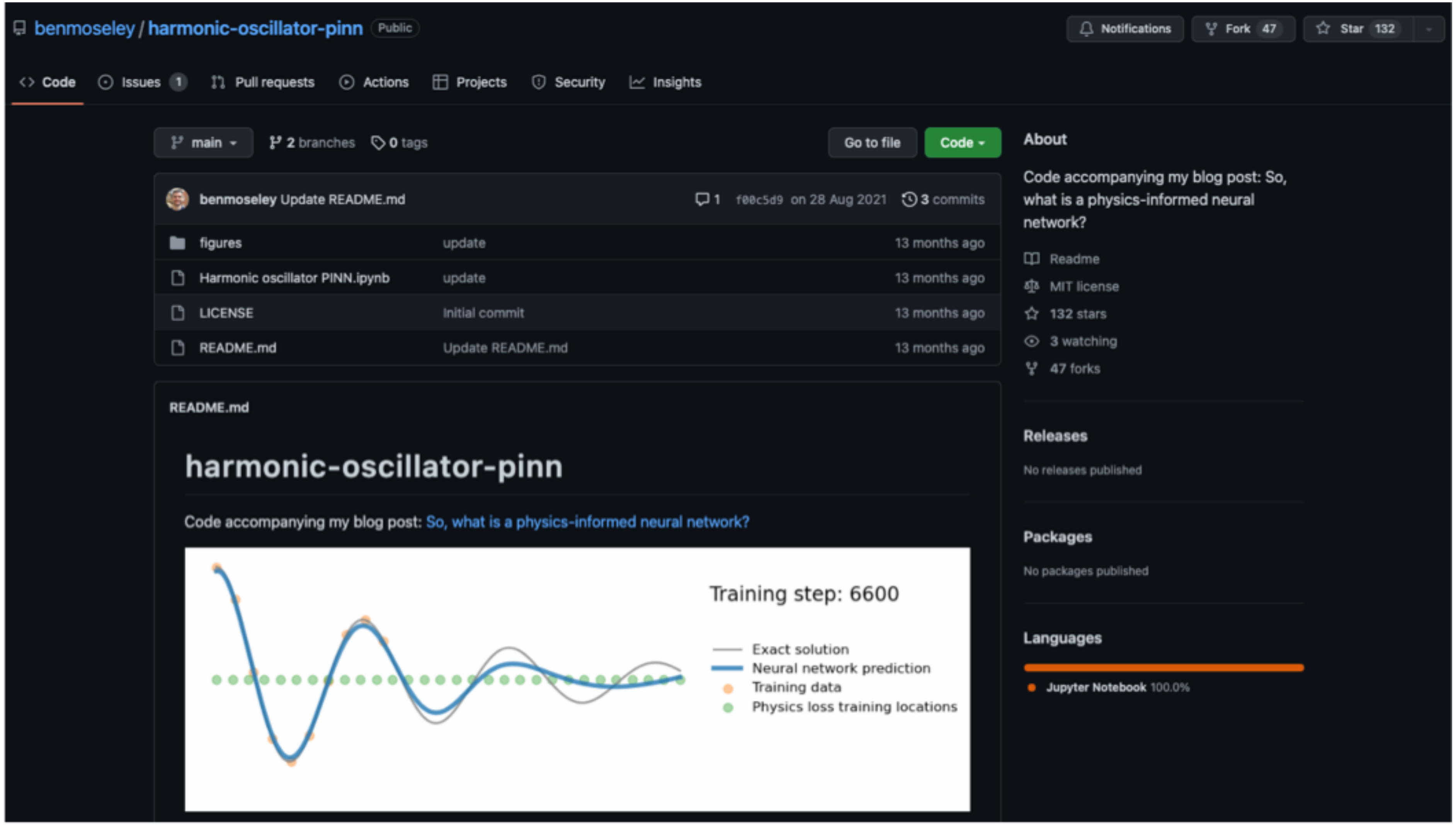
$$\mathcal{L}(\theta) = \frac{1}{N_b}\sum_{i=1}^{N_b} \| u_{\mathrm{FD}}(x_i, t_i) - NN(x_i, t_i; \theta) \|^2$$
$$+ \frac{\lambda}{N_p}\sum_{j=1}^{N_p} \left\| \left[\nabla^2 - \frac{1}{c(x_j)^2}\frac{\partial^2}{\partial t^2} \right] NN(x_j, t_j; \theta) \right\|$$



(d) Speed $v(x, y)$ recovered from PINN simulation.

- Simultaneously learn velocity model too!

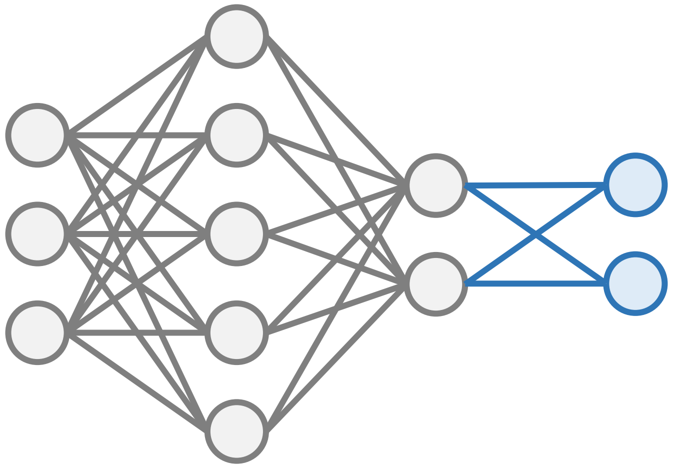
Try PINNs for yourself!



github.com/benmoseley

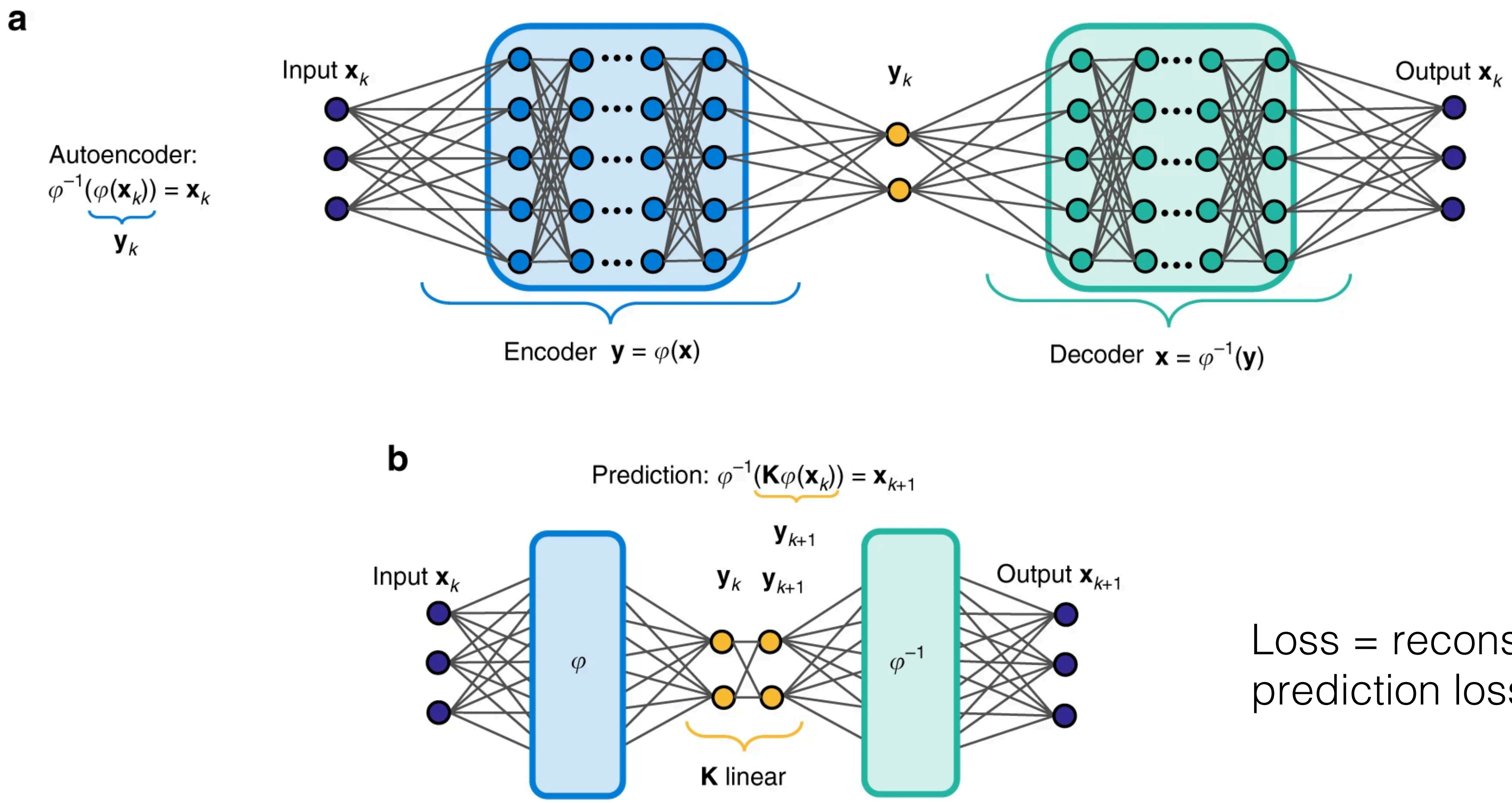
Architecture:

ML inspired by Koopman theory



Lusch, B., Kutz, J. N., & Brunton, S. L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. Nature Communications

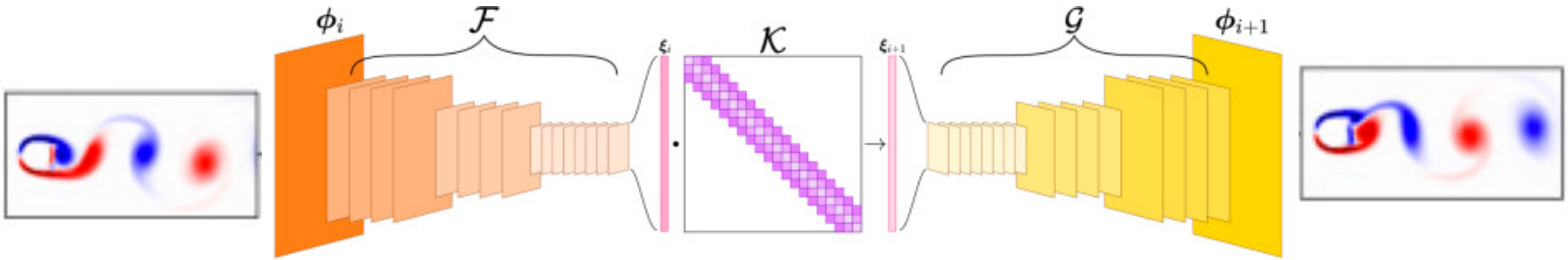
Roughly: Koopman theory states that any (potentially nonlinear) dynamical system can be represented in terms of an infinite-dimensional linear operator

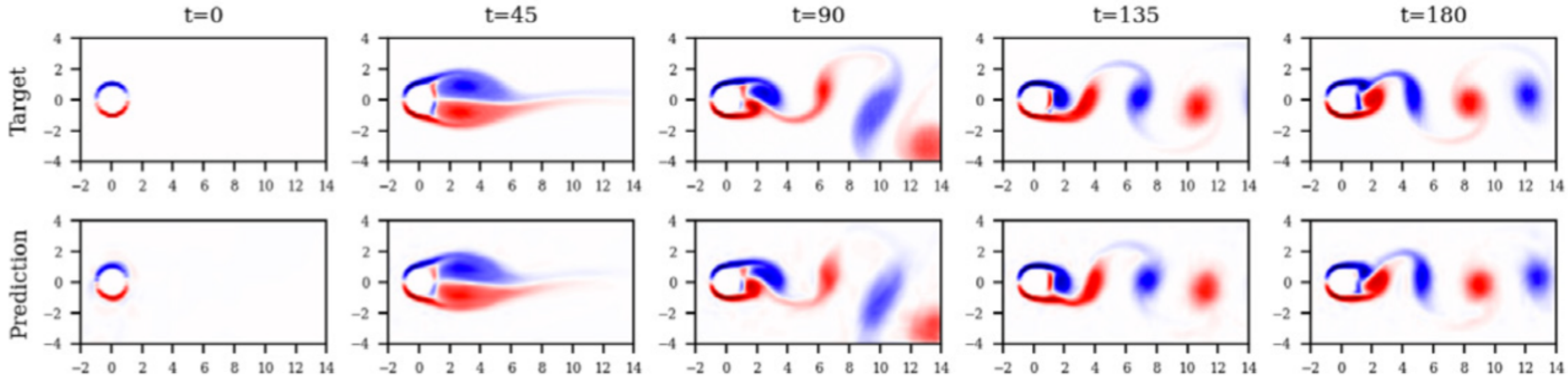


Loss = reconstruction loss + prediction loss

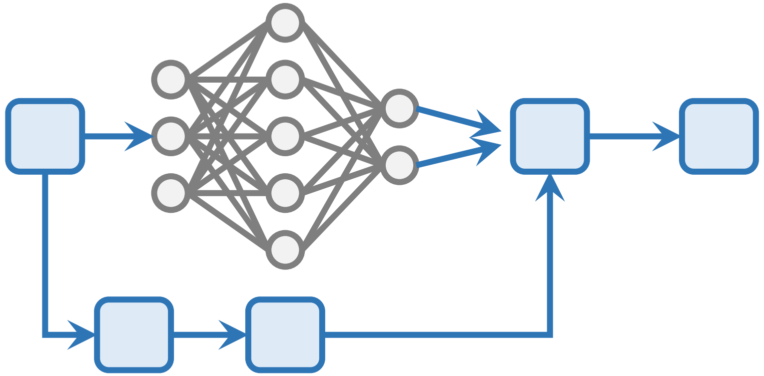
Roughly: Koopman theory states that any (potentially nonlinear) dynamical system can be represented in terms of an infinite-dimensional linear operator

Geneva, N., & Zabaras, N. (2022). Transformers for modeling physical systems. Neural Networks





Hybrid approaches:
Differentiable physics



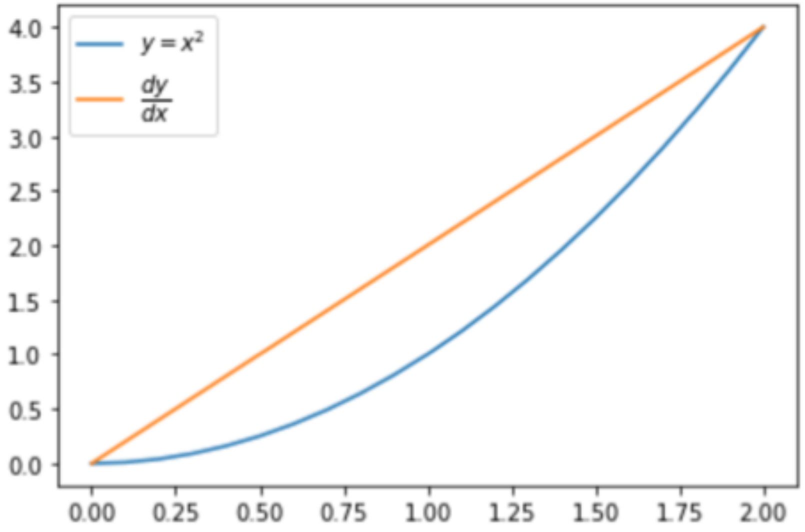
Differentiable physics

```
In [1]: import torch

x = torch.arange(0,2.1,0.1, requires_grad=True)
y = x**2

y.backward(torch.ones_like(x))

import matplotlib.pyplot as plt
plt.plot(x.detach(), y.detach(), label="$y=x^2$")
plt.plot(x.detach(), x.grad.detach(), label="$\dfrac{dy}{dx}$")
plt.legend()
plt.show()
```



Key ideas:

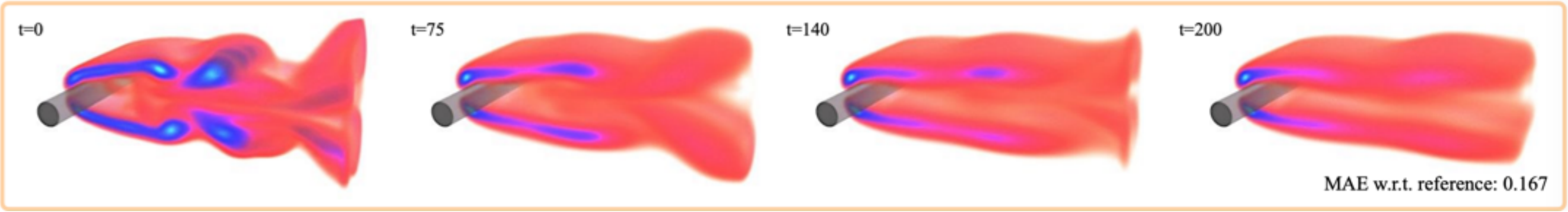
1. Auto-differentiation is very powerful!!

2. We can write (nearly) any traditional scientific algorithm in a differentiable programming language

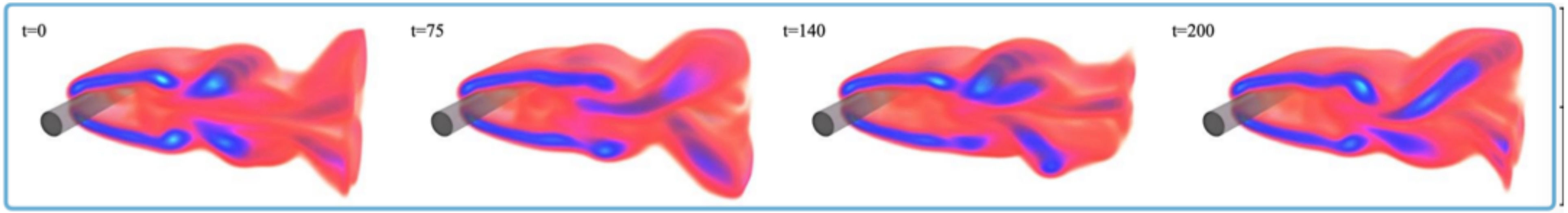
3. And insert and train ML components / flexible parameters anywhere inside them

Um, K., Brand, R., Fei, Y., Holl, P., & Thuerey, N. (2020). Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. Advances in Neural Information Processing Systems

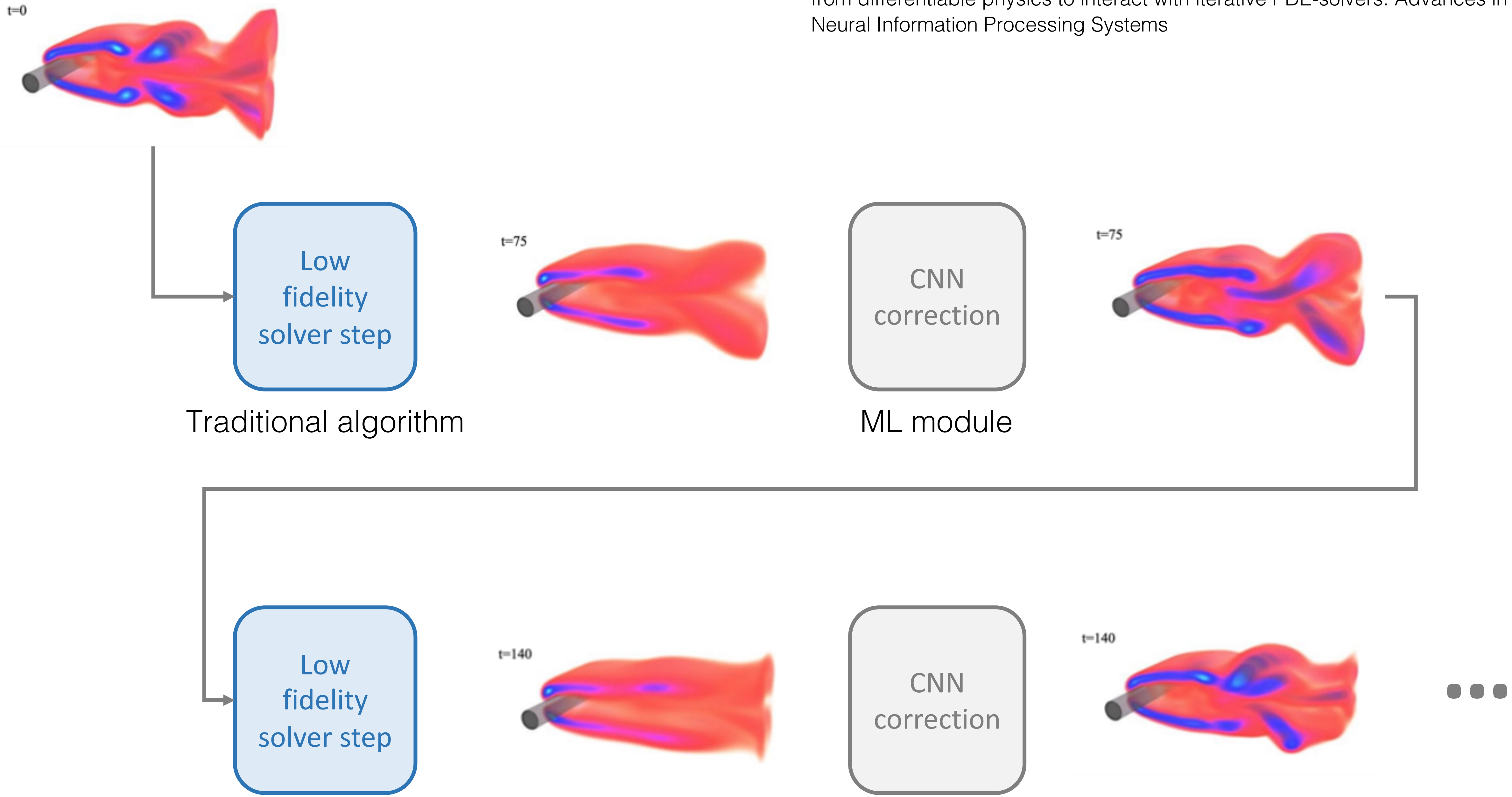
Low fidelity solver



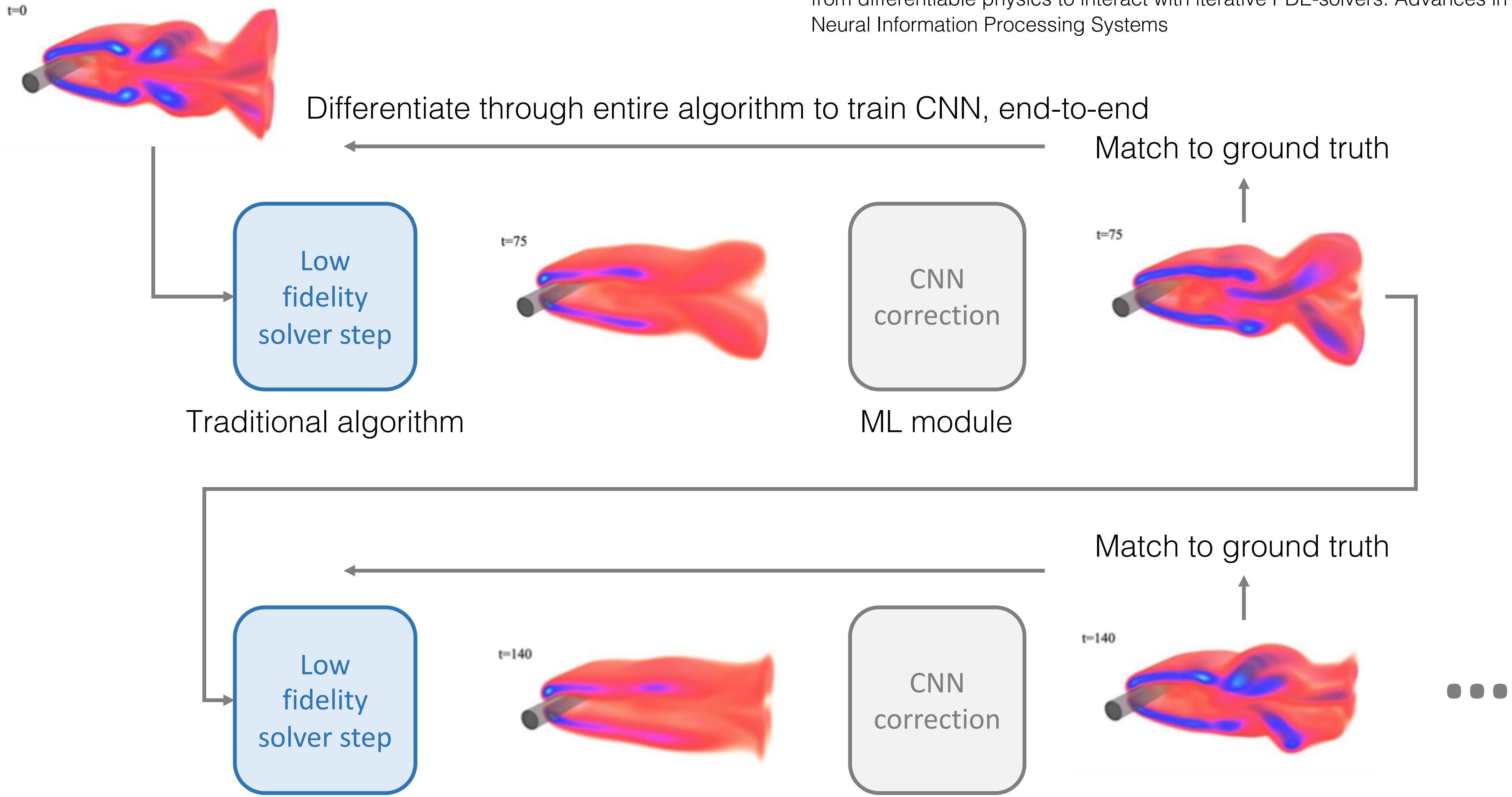
High fidelity solver
“Ground truth”



Um, K., Brand, R., Fei, Y., Holl, P., & Thuerey, N. (2020). Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. Advances in Neural Information Processing Systems



Um, K., Brand, R., Fei, Y., Holl, P., & Thuerey, N. (2020). Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. Advances in Neural Information Processing Systems

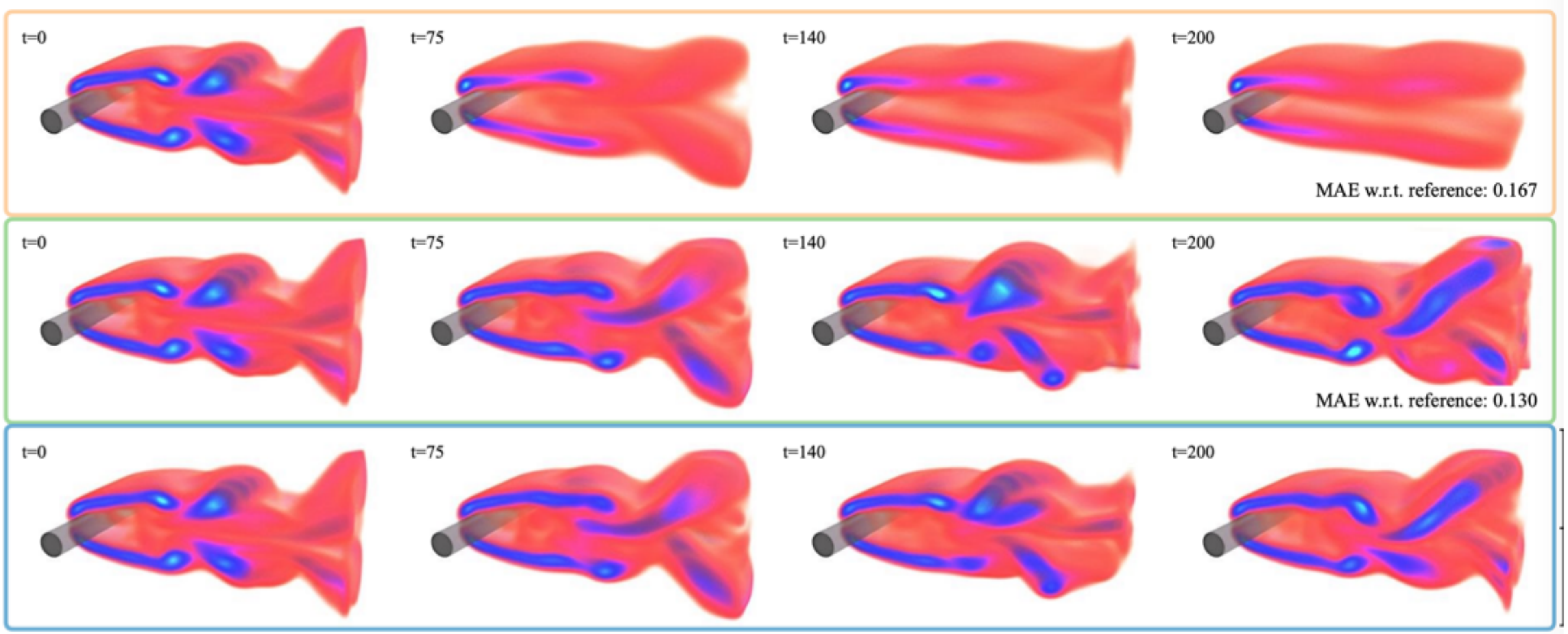


Um, K., Brand, R., Fei, Y., Holl, P., & Thuerey, N. (2020). Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. Advances in Neural Information Processing Systems

Low fidelity solver

Solver-in-the-loop

High fidelity solver
"Ground truth"



Talk overview

- What is scientific machine learning (SciML)?

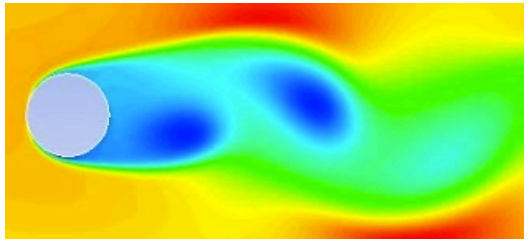
- Ways to incorporate scientific principles into ML

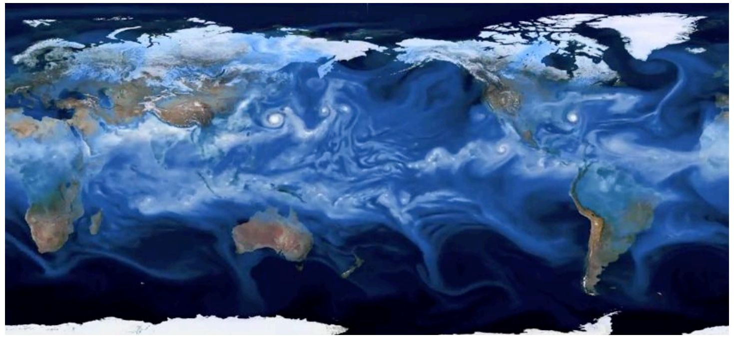
- Our research: scaling SciML techniques to complex, real-world problems

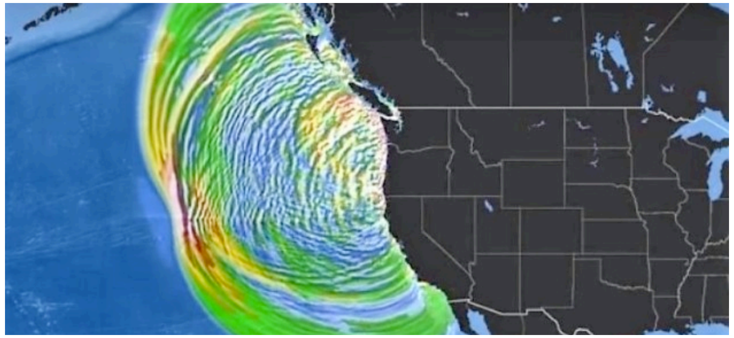
Our research:

Many SciML works focus on solving **toy/simplified problems** to validate their techniques

How well do SciML techniques scale to complex, real-world problems?







Traditional scientific methods struggle to scale when:

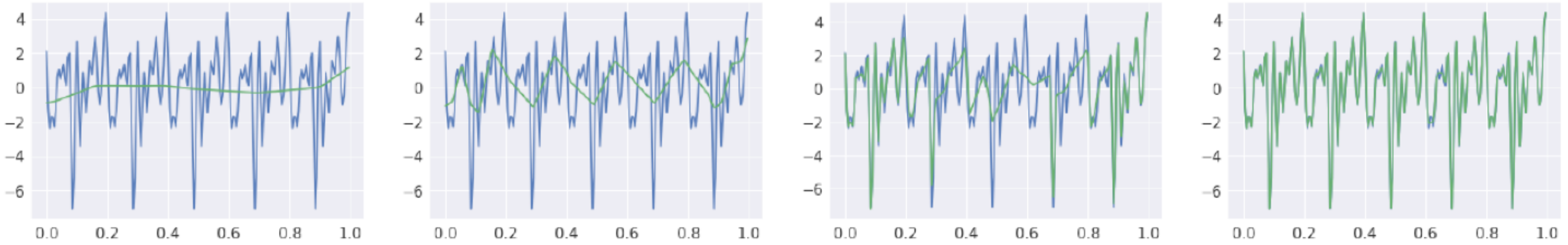
Adding more complex phenomena (multi-scale, multi-physics)

Increasing the domain size / adding higher frequencies (high computational cost)

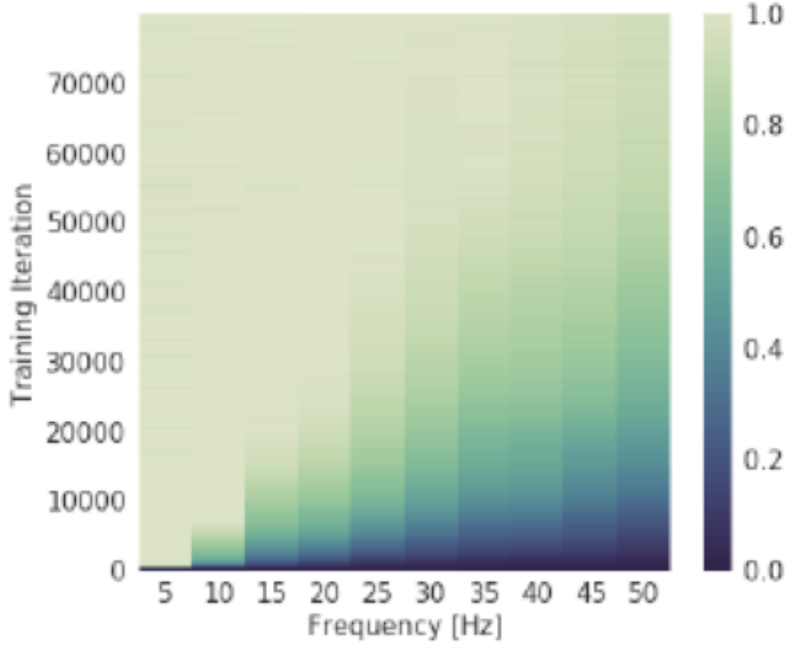
Incorporating real, noisy and sparse data

Image credits: Lawrence Berkeley National Laboratory / NOAA / NWS / Pacific Tsunami Warning Center

Scaling to large problems: The spectral bias issue



(a) Iteration 100 (b) Iteration 1000 (c) Iteration 10000 (d) Iteration 80000

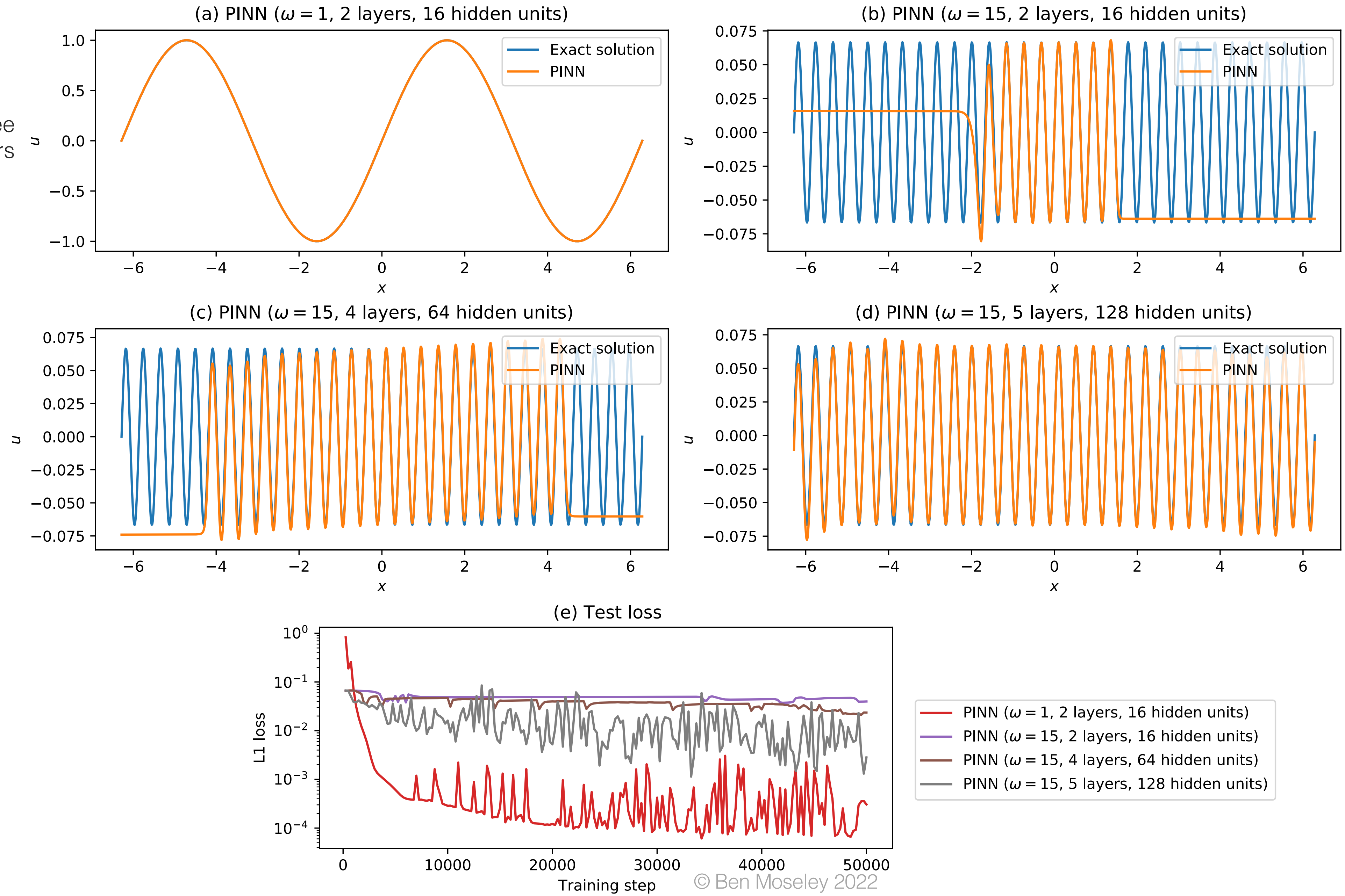


- NNs prioritise learning lower frequency functions first
- Under certain assumptions can be proved via neural tangent kernel theory

Rahaman, N., et al, On the spectral bias of neural networks. 36th International Conference on Machine Learning, ICML (2019)

A motivating problem

321 free parameters



$$\frac{du}{dx} = \cos(\omega x) \; ,$$
$$u(0) = 0 \; ,$$

66,433 free parameters

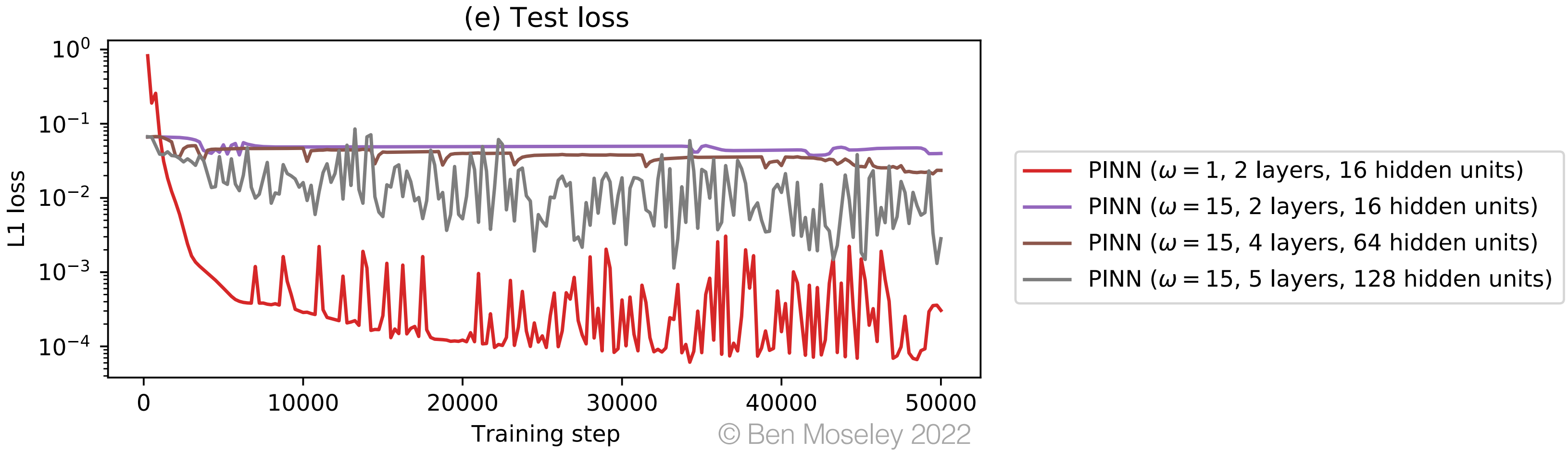
Key scaling issues

Multiple related issues when scaling PINNs to large domains:

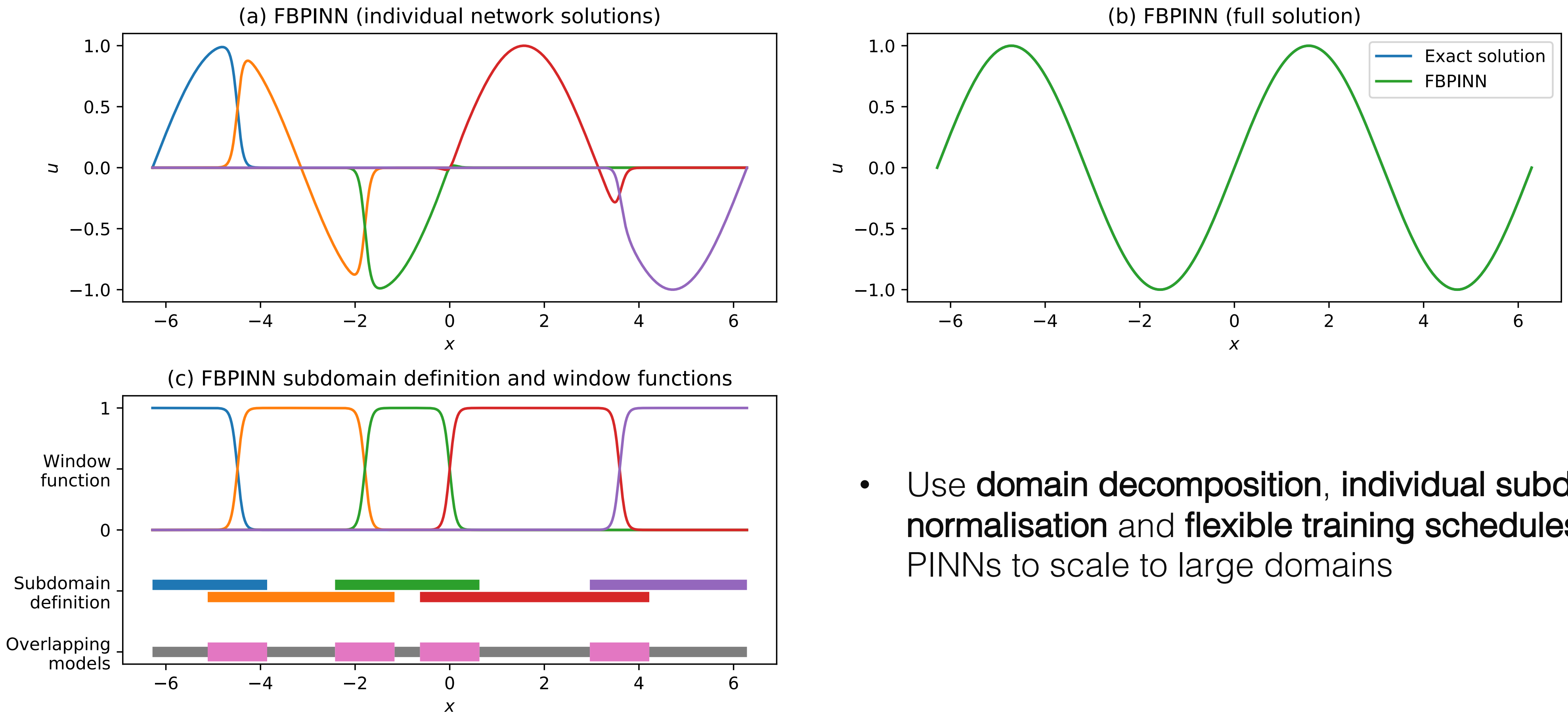
- As the domain increases:
- The complexity of the solution increases;
- Requiring a larger neural network (more free parameters);
- And more training points to sample the domain;
- -> Leading to a harder optimisation problem.

- As the frequency increases;
- The neural network takes longer to converge (spectral bias).

- As the size of the network, number of training points, and convergence time grows, the computational resources grows significantly



Finite basis physics-informed neural networks (FBPINNs)



- Use **domain decomposition**, **individual subdomain normalisation** and **flexible training schedules** to allow PINNs to scale to large domains

$$\hat{u}(x;\theta) = \mathcal{C}\left[\sum_{i}^{n} w_i(x) \cdot \mathrm{unnorm} \circ NN_i \circ \mathrm{norm}_i(x)\right]$$

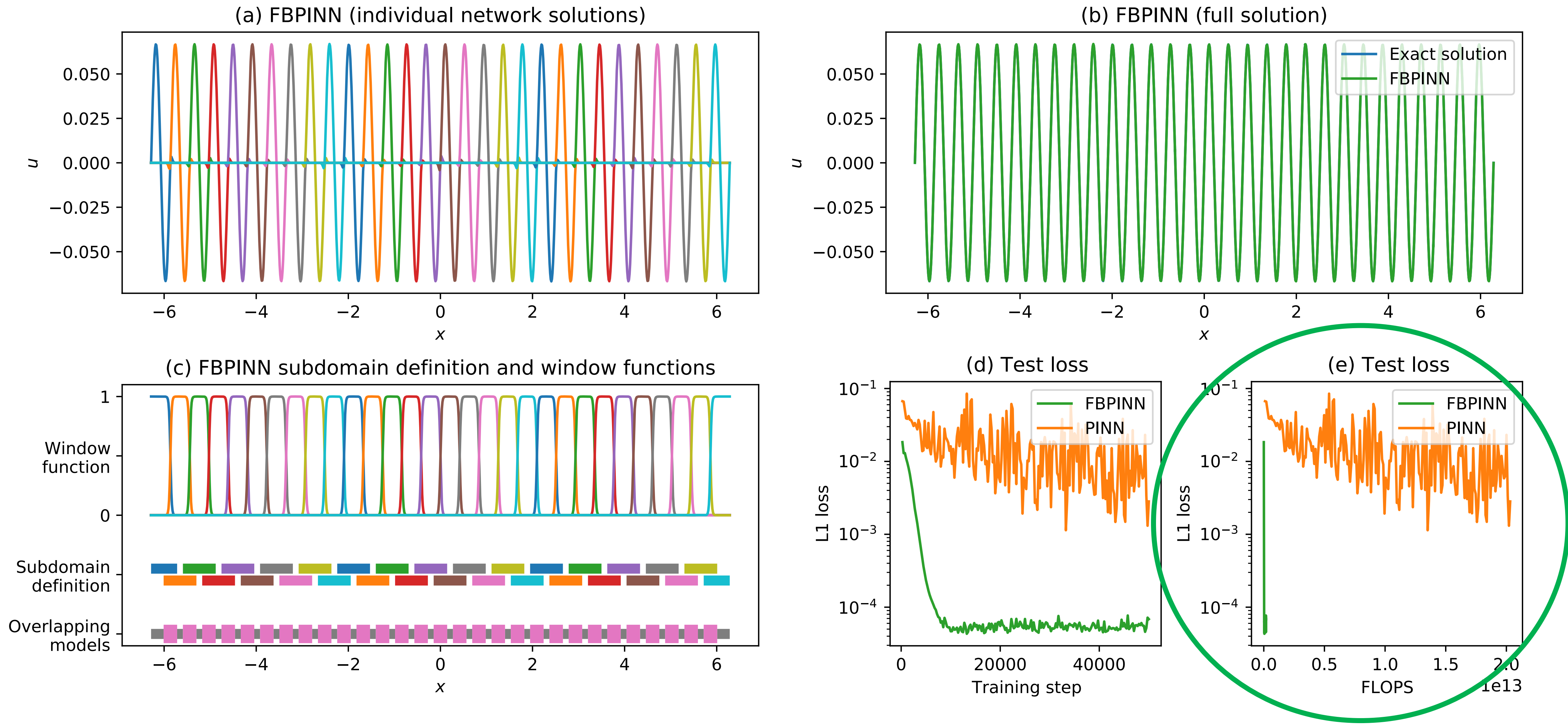
Window function

Subdomain network

Separate subdomain normalisation

https://arxiv.org/abs/2107.07871

$$\frac{du}{dx} = \cos(\omega x)\ , \quad u(0) = 0\ , \qquad u(x) = \frac{1}{\omega}\sin(\omega x) \qquad \hat{u}(x;\theta) = \tanh(\omega x)\overline{NN}(x;\theta) \qquad \omega = 15$$



- Subdomain network size: 2 layers, 16 hidden units

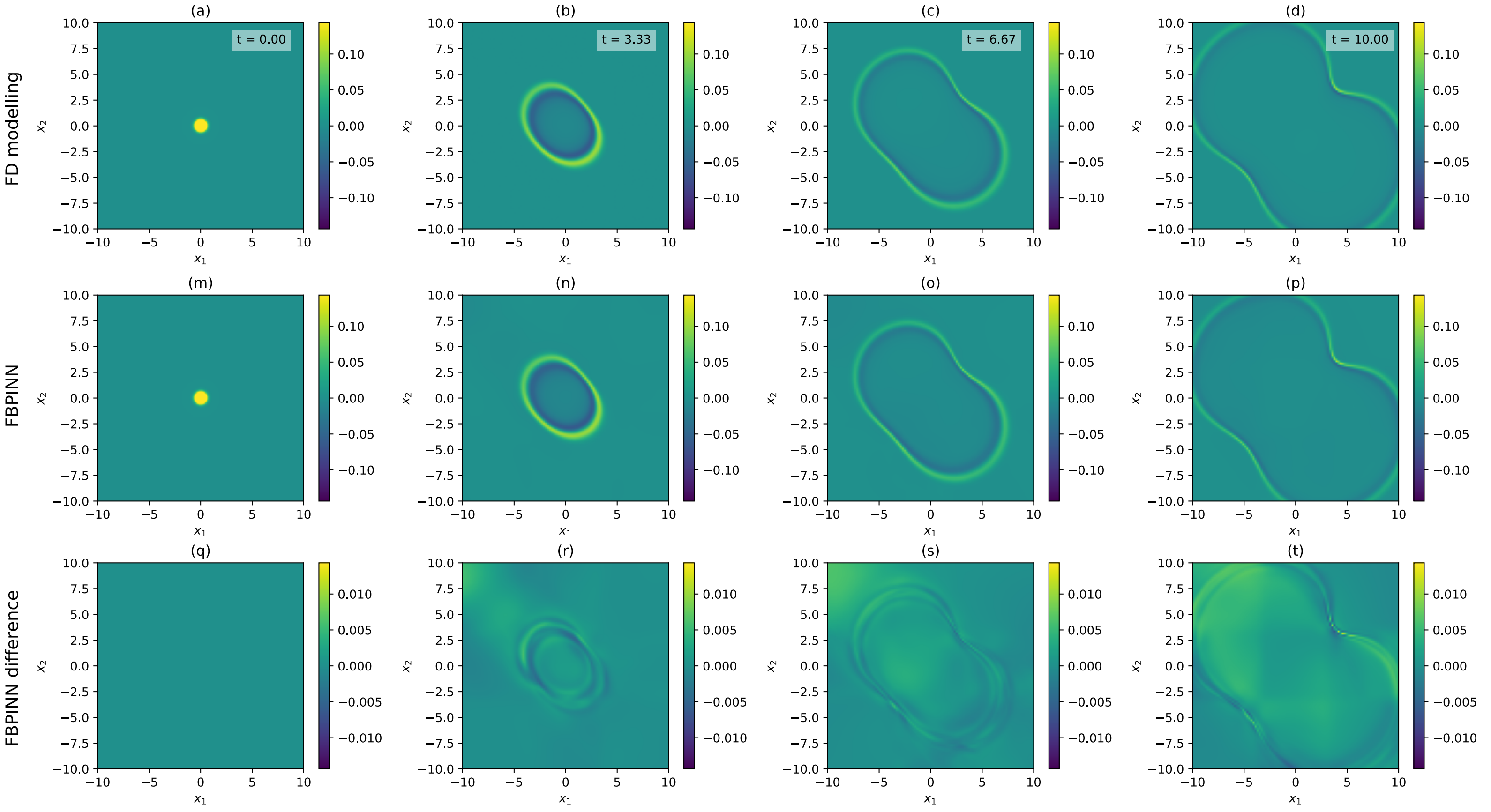
- (2+1)D time-dependent wave equation

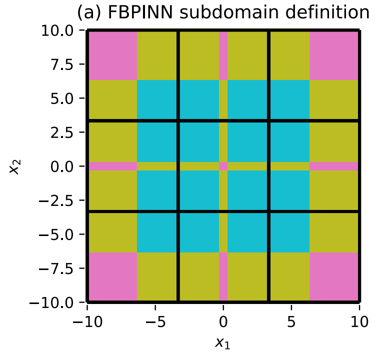
$$\left[\nabla^2 - \frac{1}{c(x)^2}\frac{\partial^2}{\partial t^2}\right] u(x,t) = 0\ ,$$

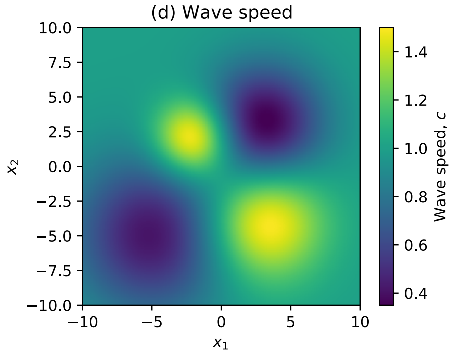
$$u(x,0) = e^{-\frac{1}{2}(||x-\mu||/\sigma)^2}\ ,$$

$$\frac{\partial u}{\partial t}(x,0) = 0\ ,$$

$$\hat{u}(x,t;\theta) = \phi(5(2-t/t_1))\, e^{-\frac{1}{2}(||x-\mu||/\sigma)^2} + \tanh^2(t/t_1)\overline{NN}(x,t;\theta)$$



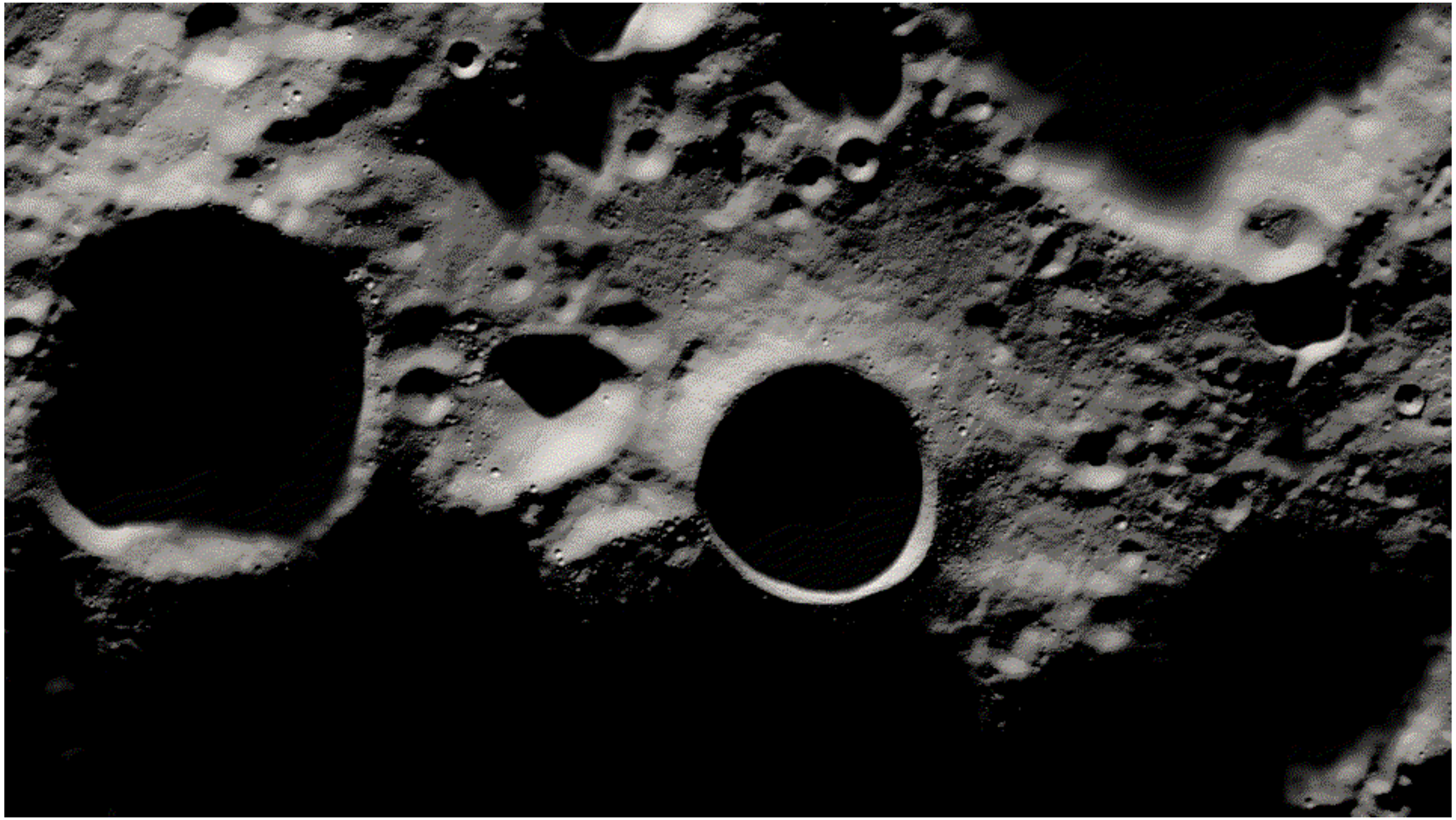




github.com/benmoseley/FBPINNs

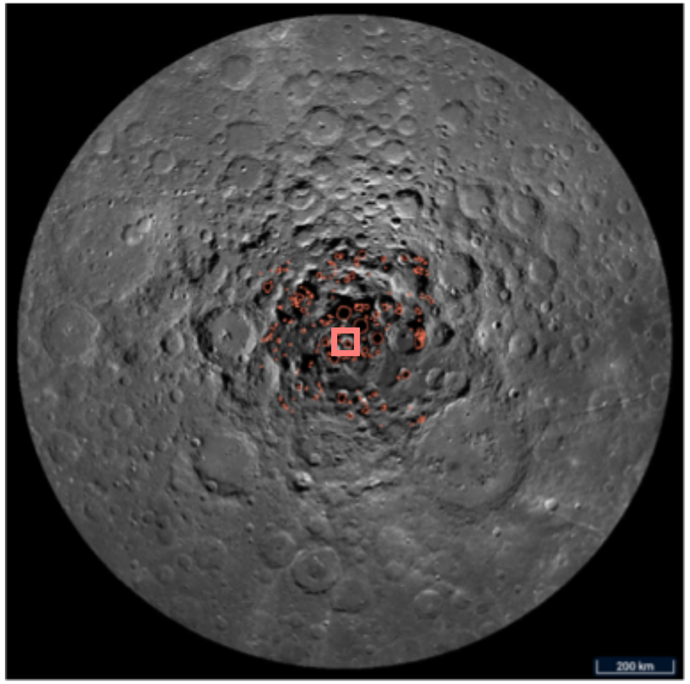
Peering into shadows on the Moon with SciML

Shackleton crater, lunar south pole

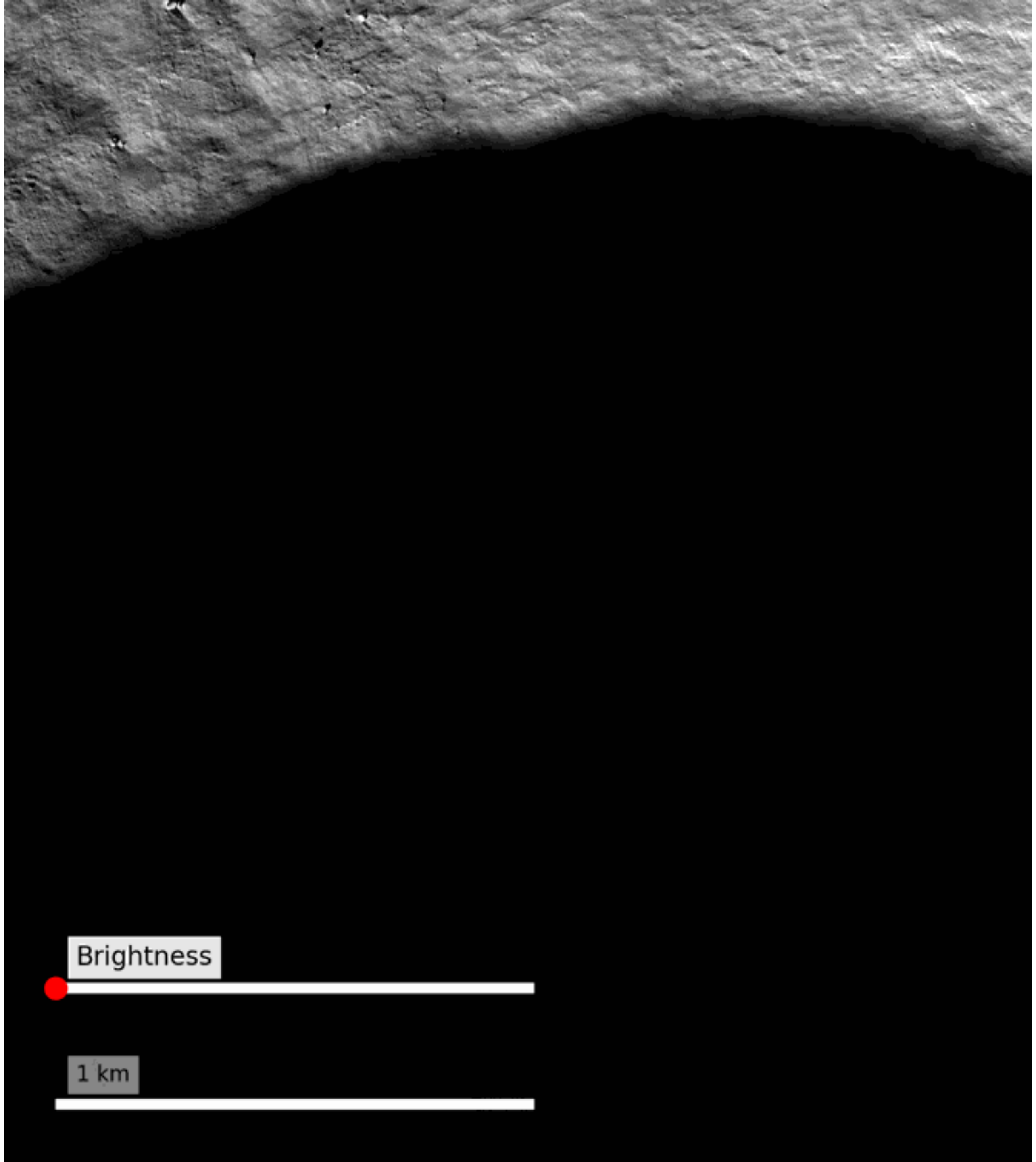


21 km

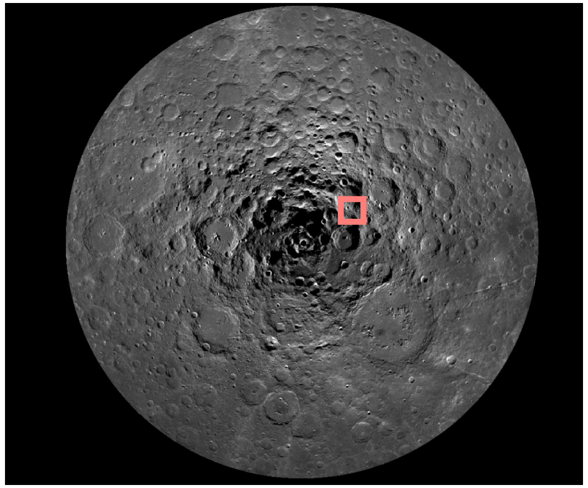
Lunar south pole



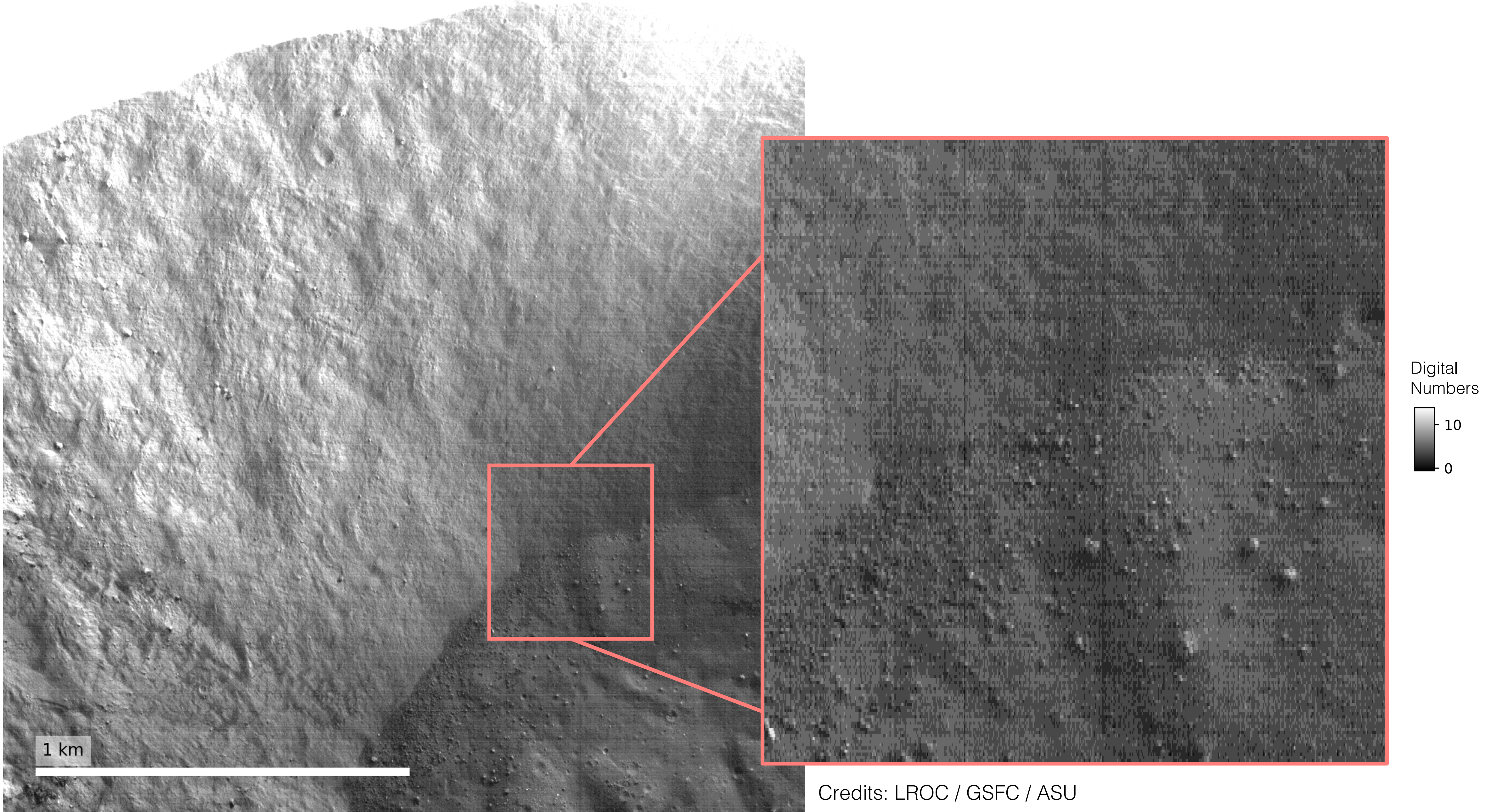
Credits: NASA Scientific Visualization Studio / QuickMap / LROC



Wapowski crater

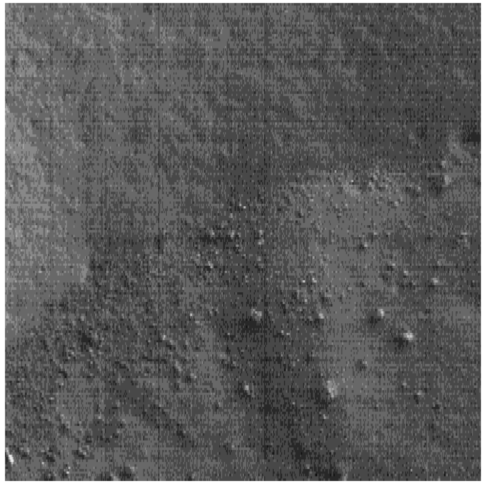


Credits: LROC / GSFC / ASU



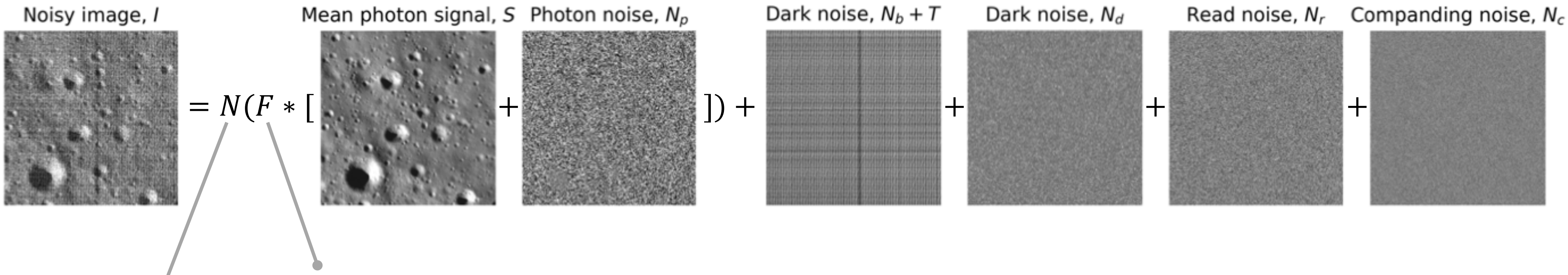
Credits: LROC / GSFC / ASU

Lunar Reconnaissance Orbiter Camera

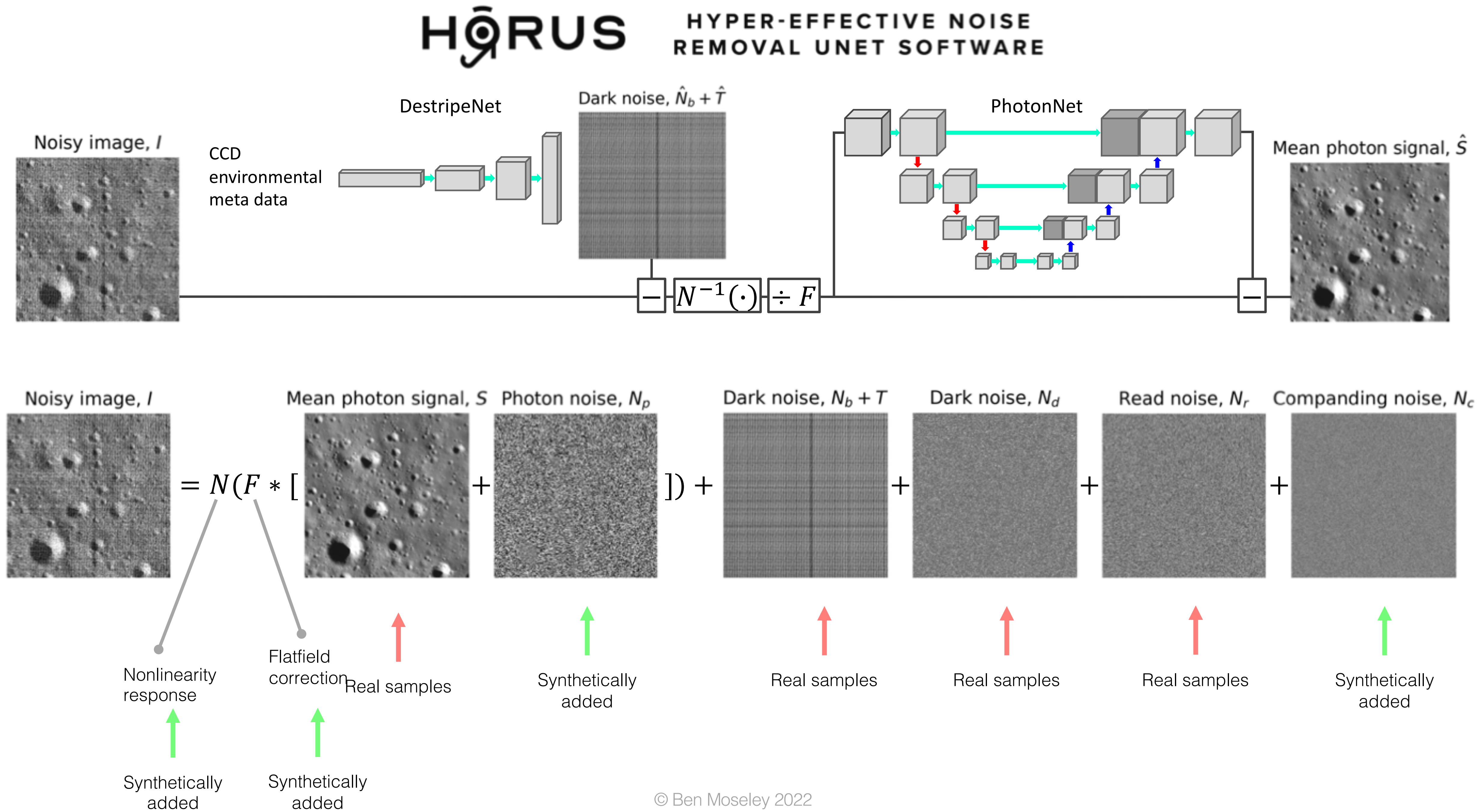


Smartphone

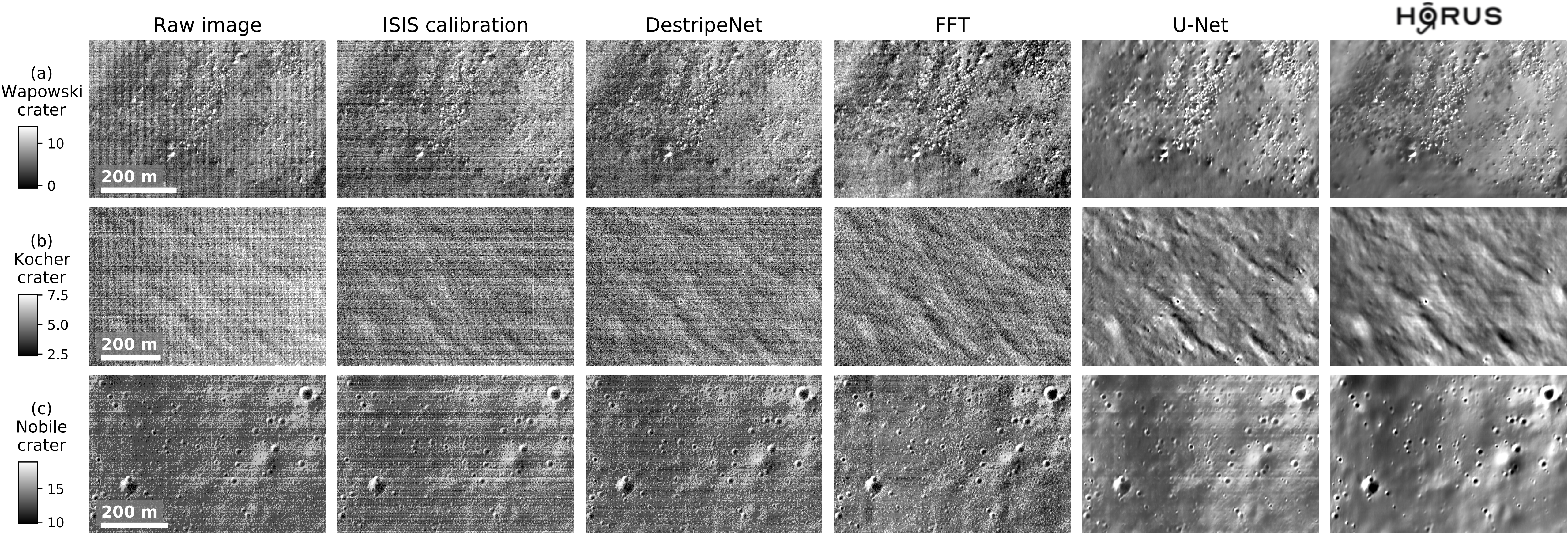
Physical noise model



Building on:
Humm, D. C. et al (2016) Flight Calibration of the LROC Narrow Angle Camera. Space Science Reviews
Robinson, M. S. et al (2010) Lunar reconnaissance orbiter camera (LROC) instrument overview. Space Science Reviews


HORUS
HYPER-EFFECTIVE NOISE REMOVAL UNET SOFTWARE
Noisy image, I
CCD environmental meta data
DestripeNet
Dark noise, $\hat{N}_b + \hat{T}$
PhotonNet
Mean photon signal, $\hat{S}$
$-$
$N^{-1}(\cdot)$
$\div F$
$-$
Noisy image, I
Mean photon signal, S
Photon noise, N_p
Dark noise, $N_b + T$
Dark noise, N_d
Read noise, N_r
Companding noise, N_c
$= N(F * [\quad + \quad]) + \quad + \quad + \quad +$
Nonlinearity response
Flatfield correction
Real samples
Synthetically added
Real samples
Real samples
Real samples
Synthetically added
Synthetically added
Synthetically added

Comparison to other methods



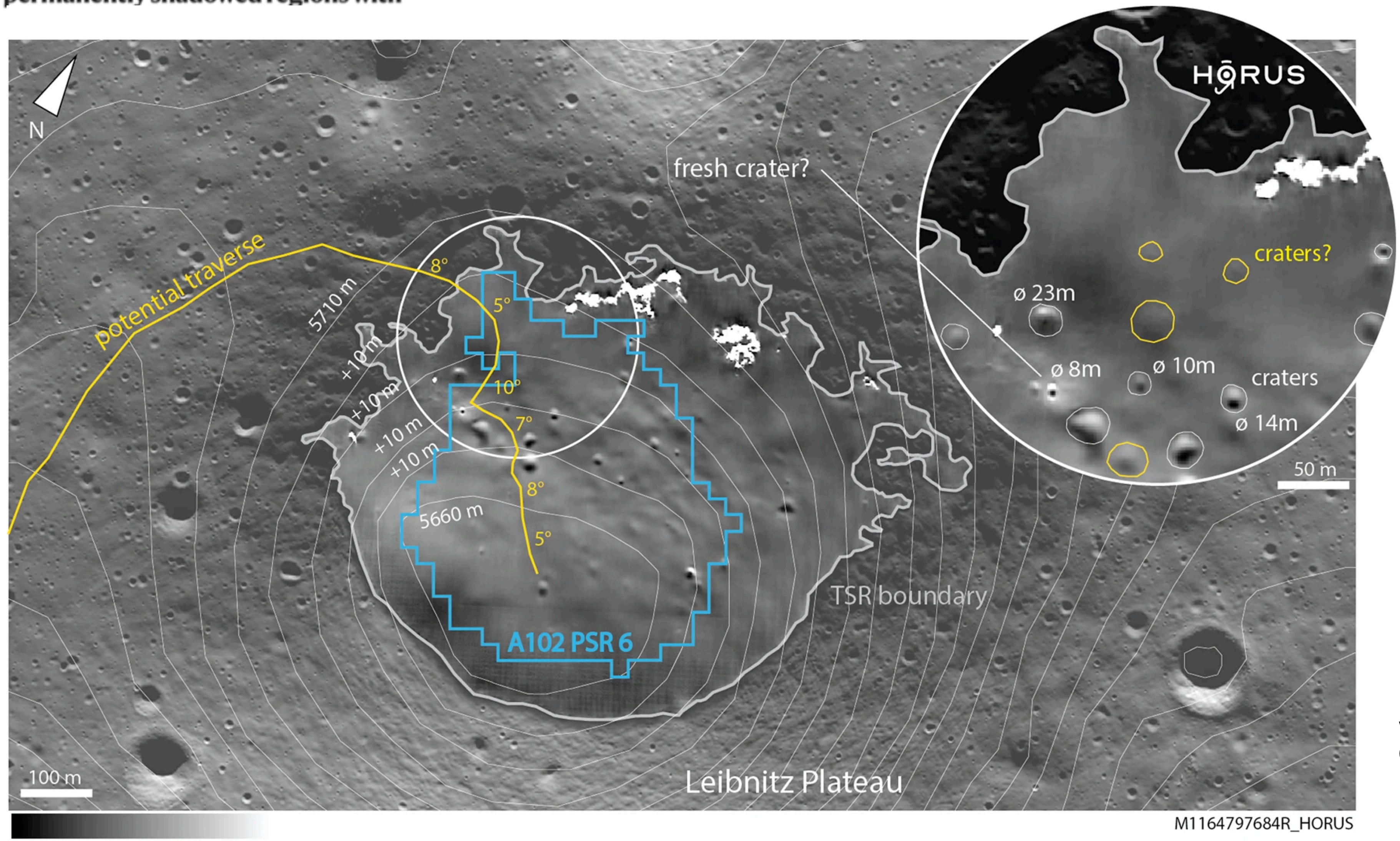
nature communications

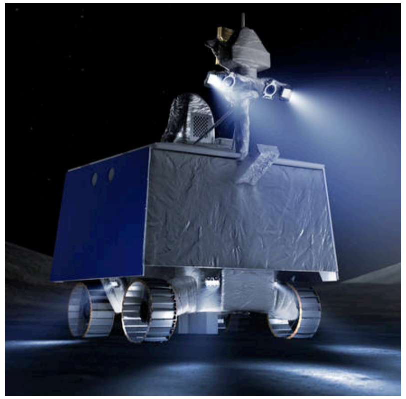
Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > nature communications > articles > article

Article | Open Access | Published: 23 September 2021

Peering into lunar permanently shadowed regions with deep learning





VIPER Lunar rover
Credits: NASA Ames/Daniel Rutter

Takeaways

- SciML is a blossoming field of research
- There are a plethora of different SciML techniques…
- …which range in the way scientific constraints are added, and their intended scientific task
- Scaling SciML techniques to more complex, multi-scale, multi-physics problems remains an exciting field of research!
- Check out my blog/ GitHub for more!

  benmoseley.blog github/benmoseley