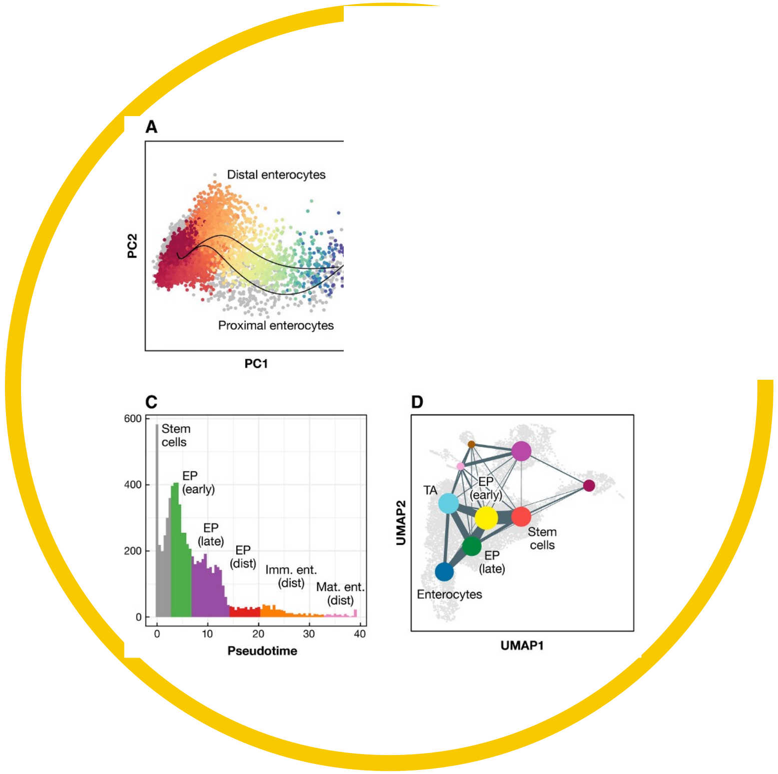
Scientific Machine Learning

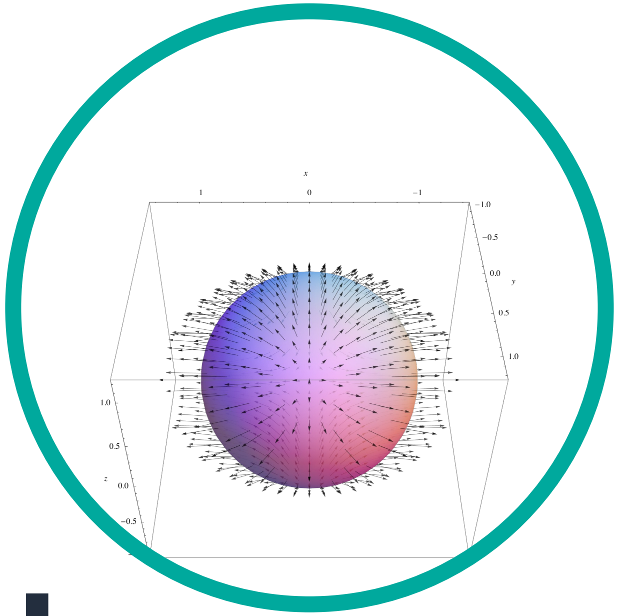
Christian Buus Michelsen @ TICRA

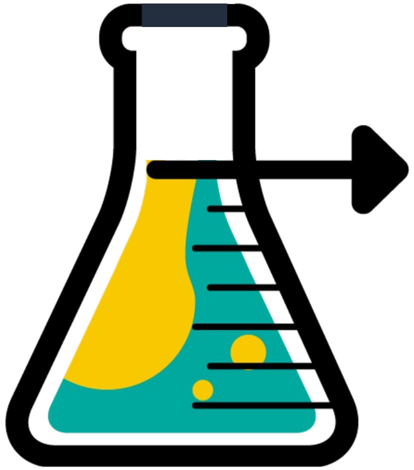
2025/06/04

TICRA

SciML: model-based, data-efficient machine learning





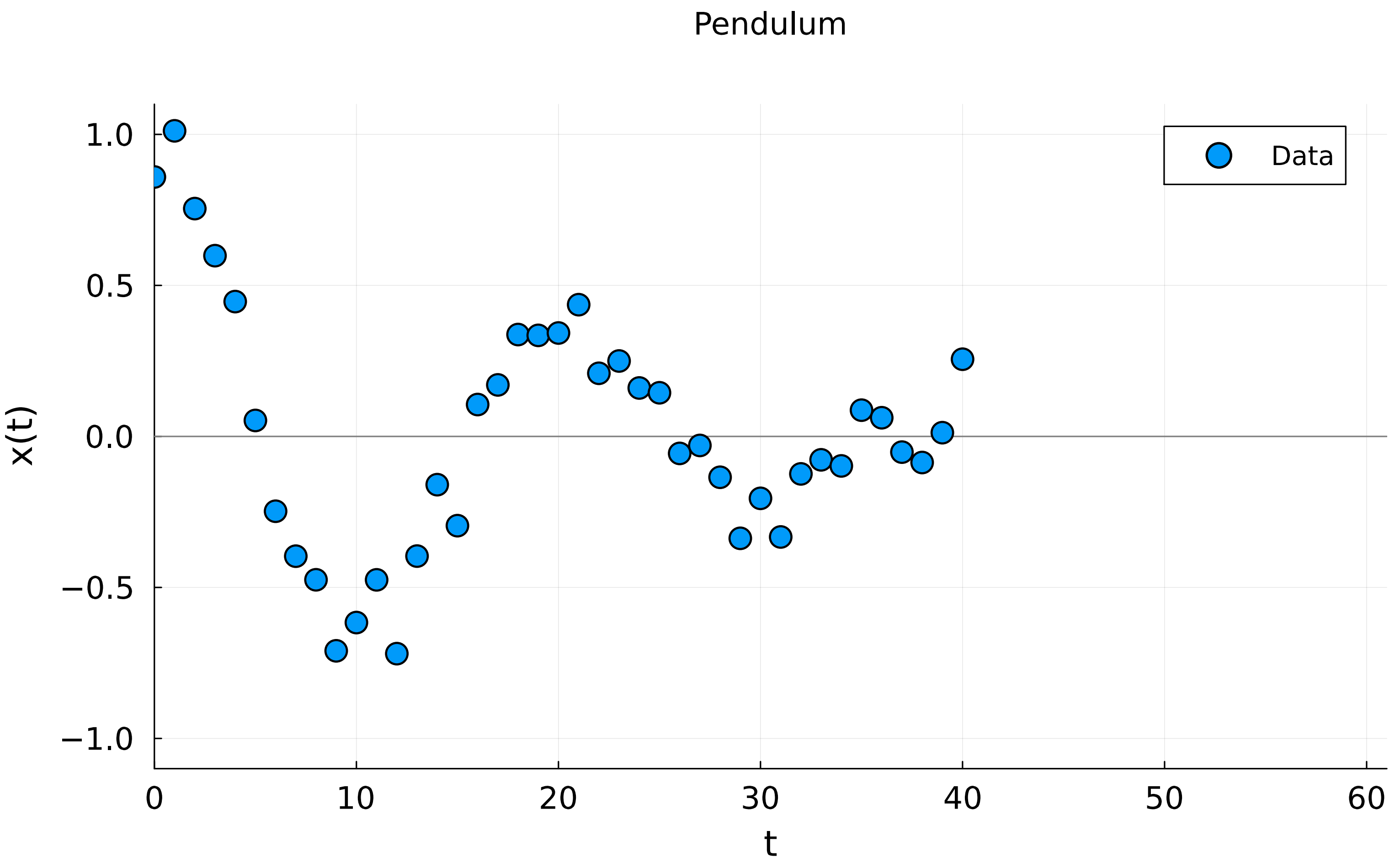


Good Predictions

TICRA

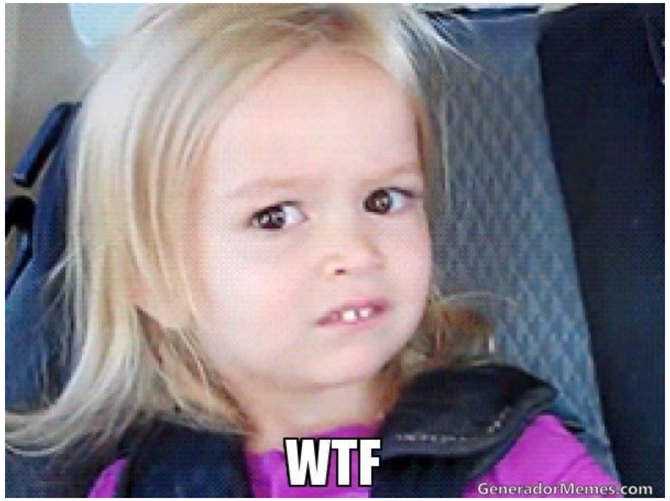
Simple Harmonic Oscillator

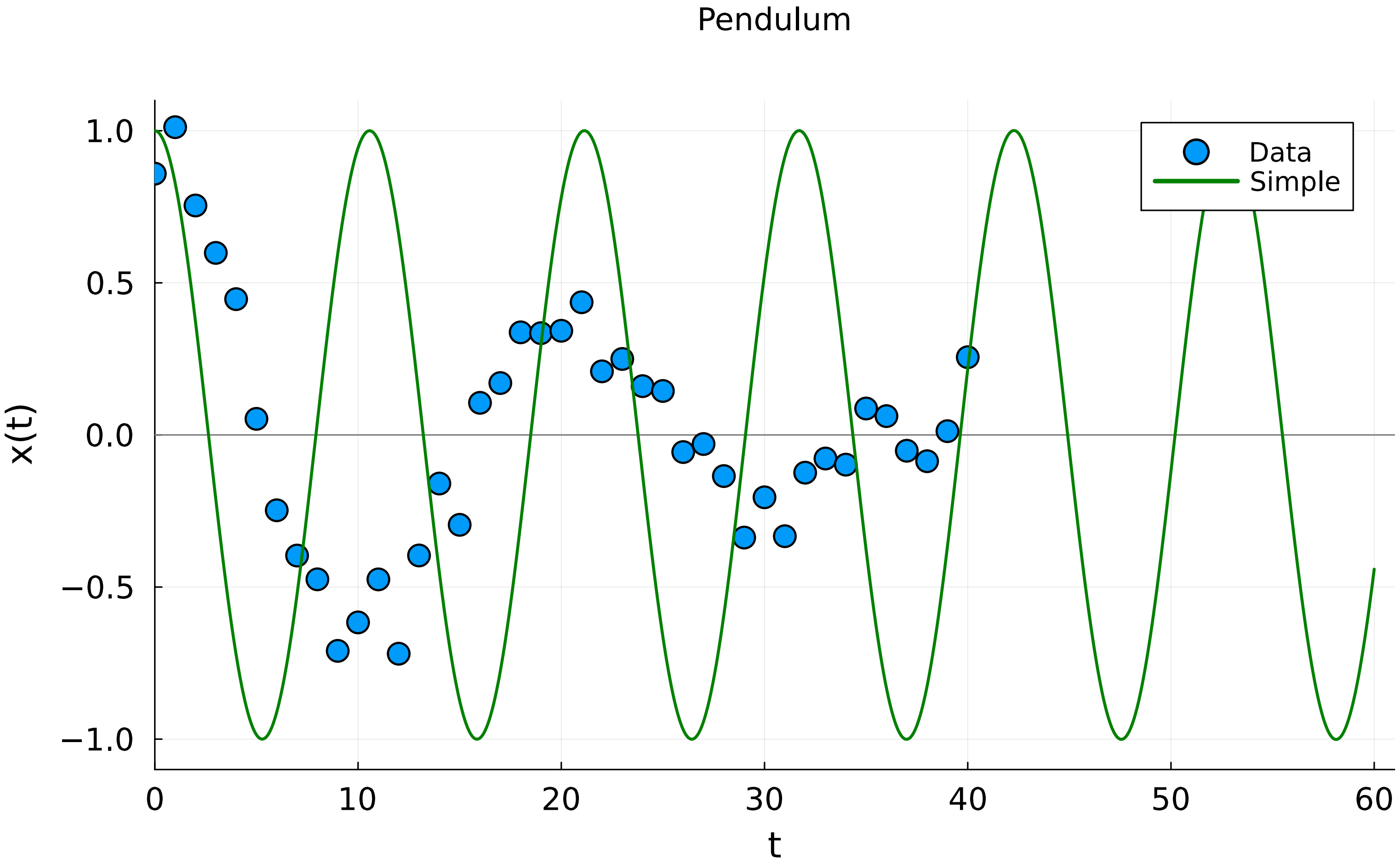
$$m\ddot{x} + kx = 0$$



Simple Harmonic Oscillator?

$$m\ddot{x} + kx = 0$$



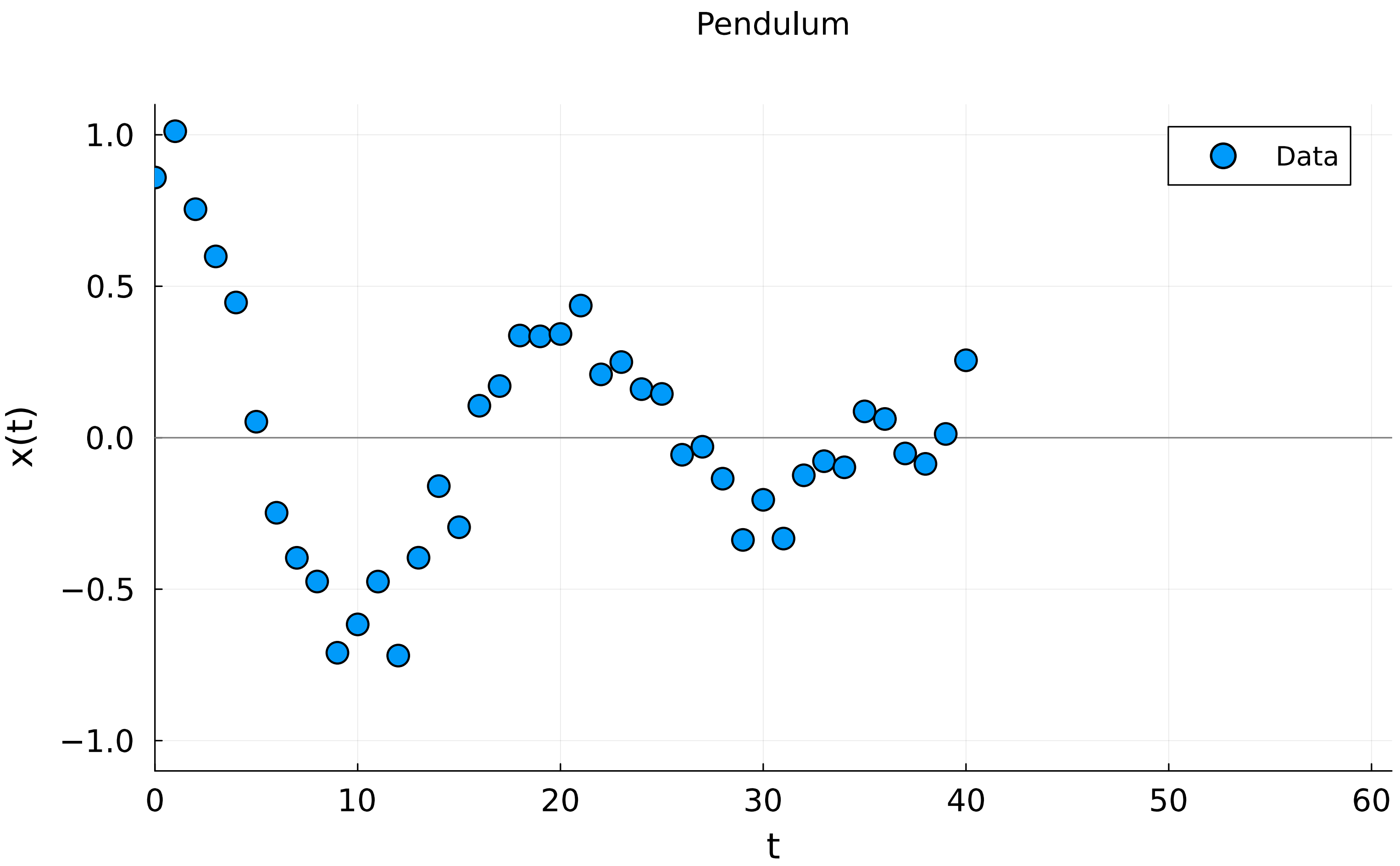


Neural networks

NN = Universal approximators

(Universal Approximation Theorem, G. Cybenko 1989)

$$x(t) = NN(t)$$



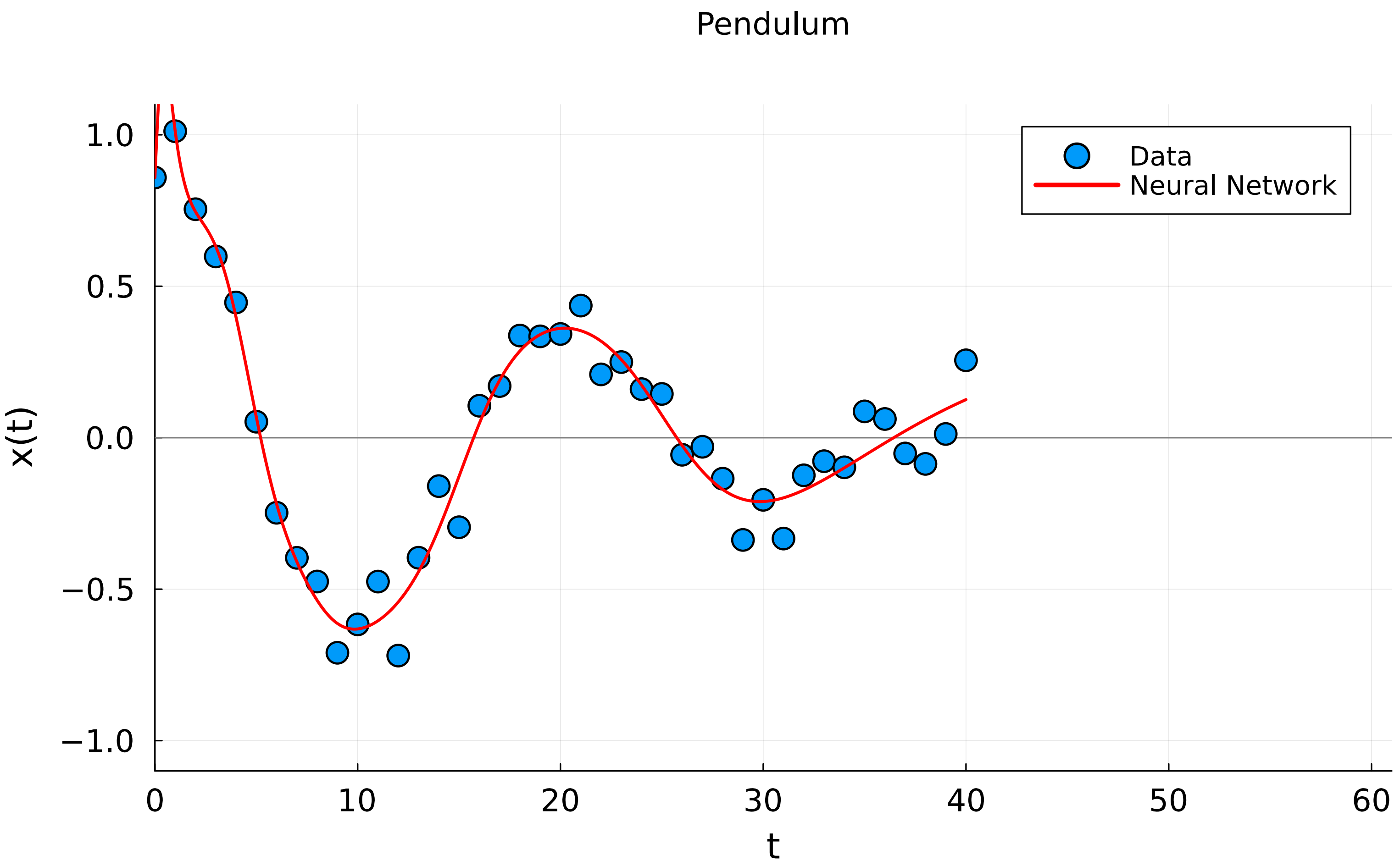
TICRA

Neural networks

NN = Universal approximators

(Universal Approximation Theorem, G. Cybenko 1989)

$$x(t) = NN(t)$$



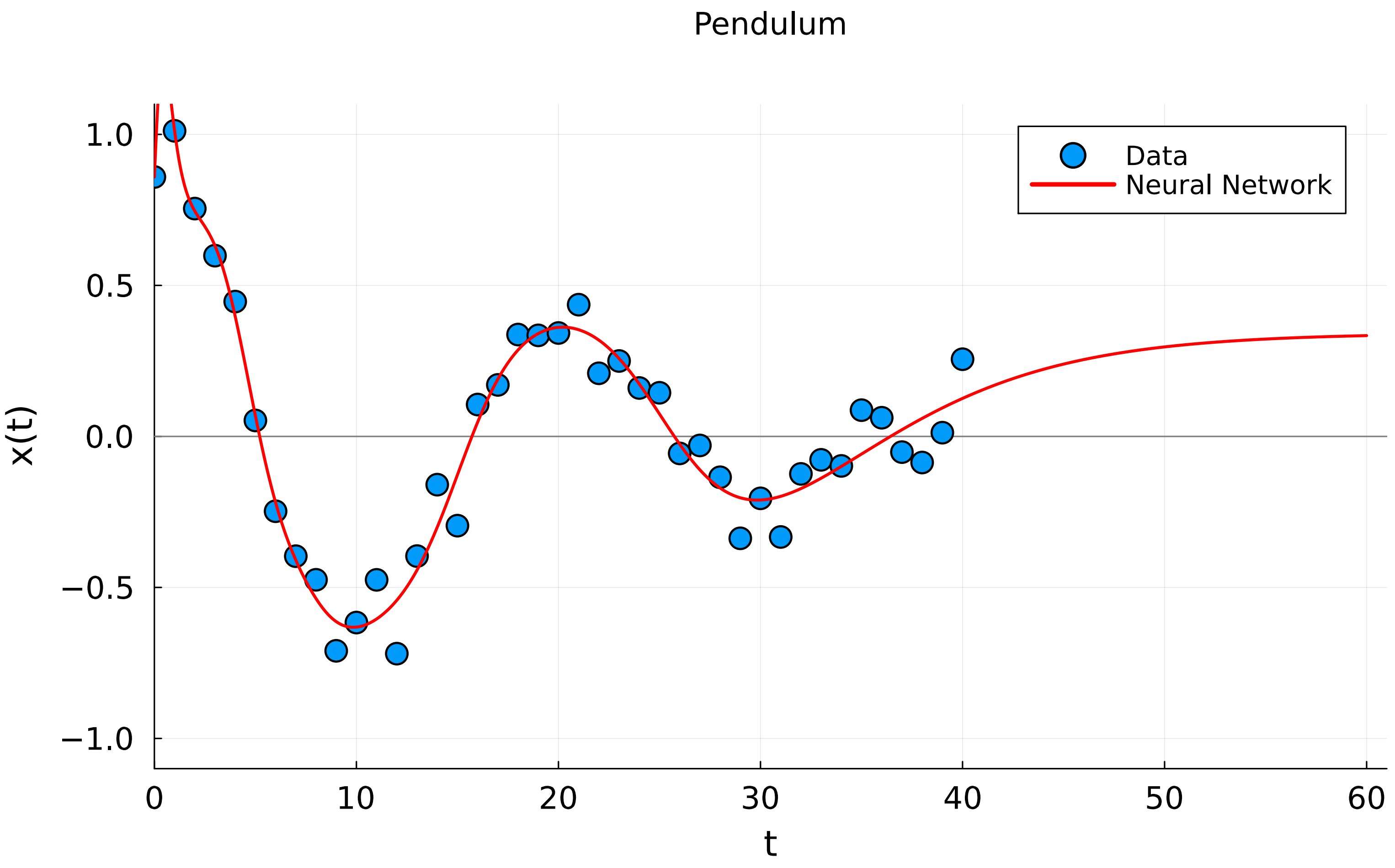
TICRA

Neural networks

NN = Universal approximators

(Universal Approximation Theorem, G. Cybenko 1989)

$$x(t) = NN(t)$$

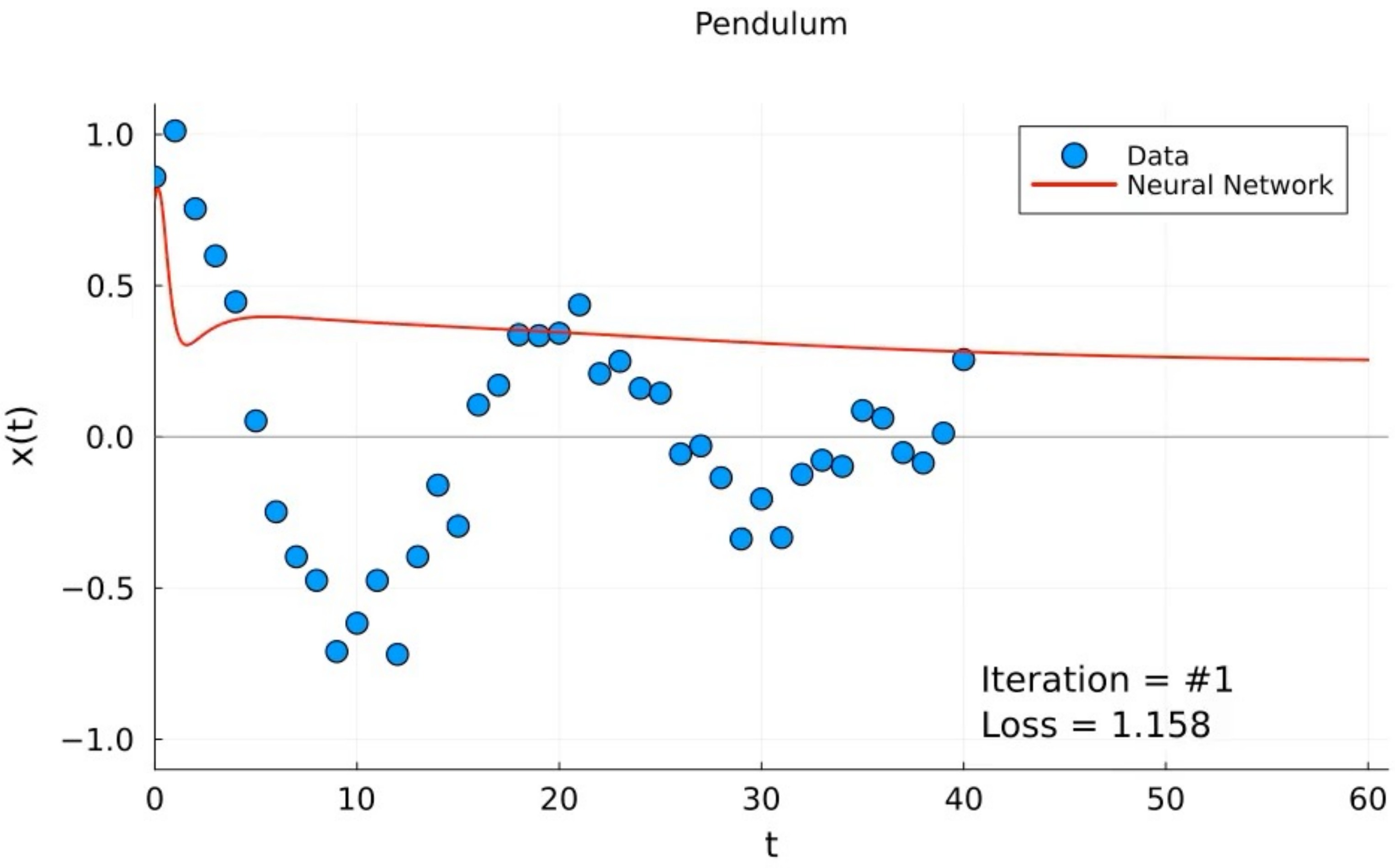


Neural networks

NN = Universal approximators

(Universal Approximation Theorem, G. Cybenko 1989)

$$x(t) = NN(t)$$

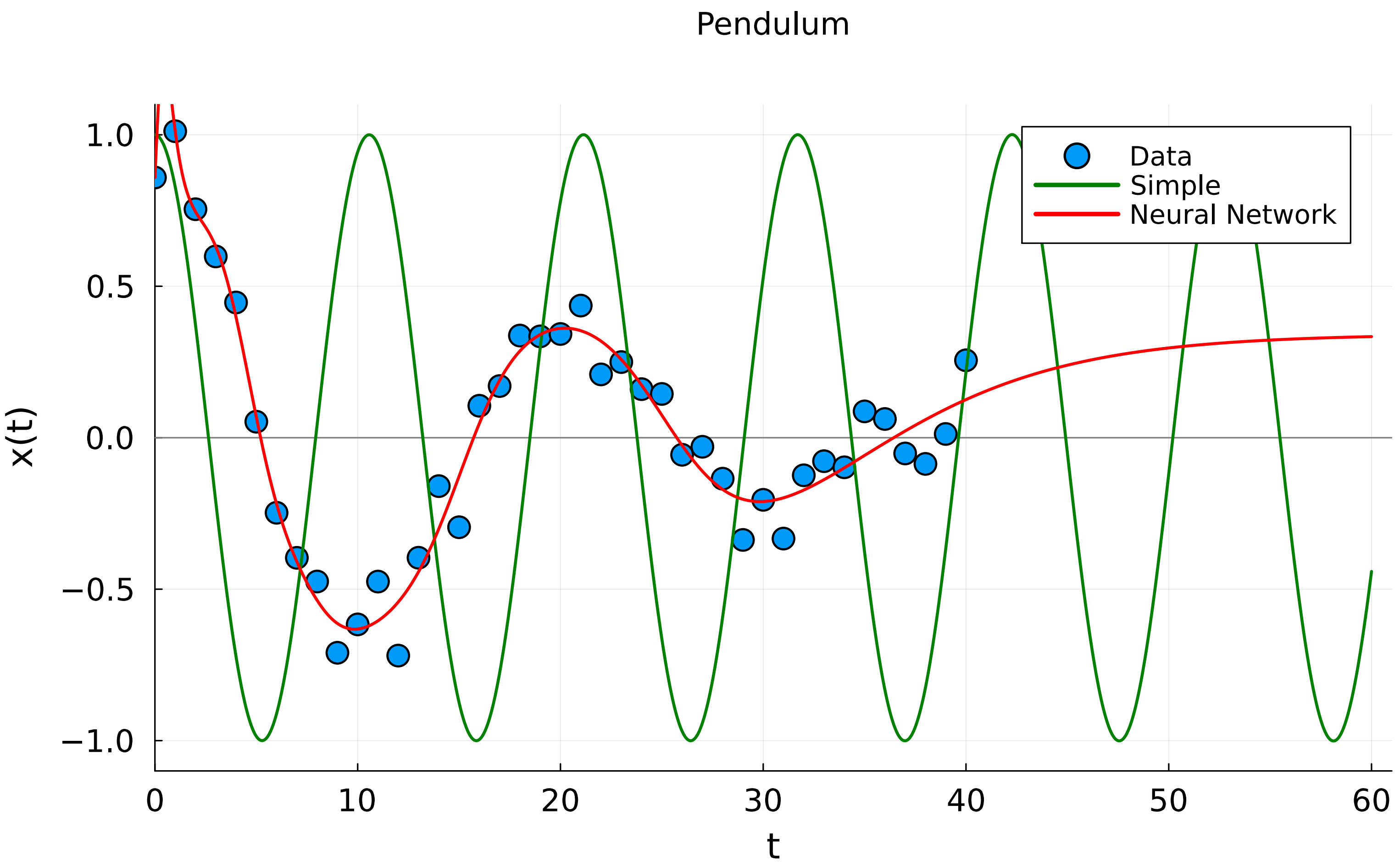


Neural networks

NN = Universal approximators

(Universal Approximation Theorem, G. Cybenko 1989)

$$x(t) = NN(t)$$

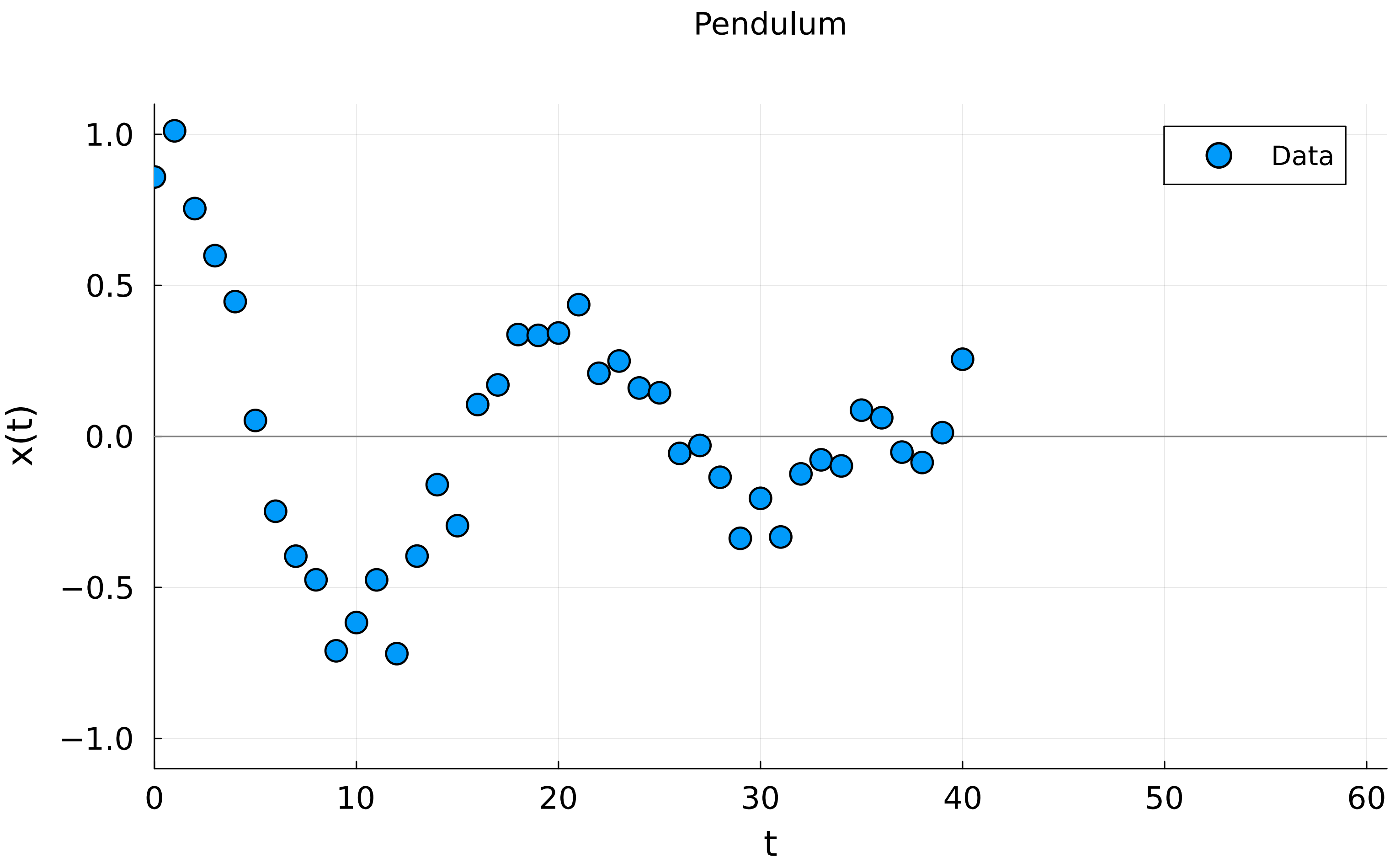


Harmonic Oscillator?

2^{nd} order ODE → 1^{st} order system:

$$\dot{x} = v$$

$$\dot{v} = -\frac{kx}{m}$$



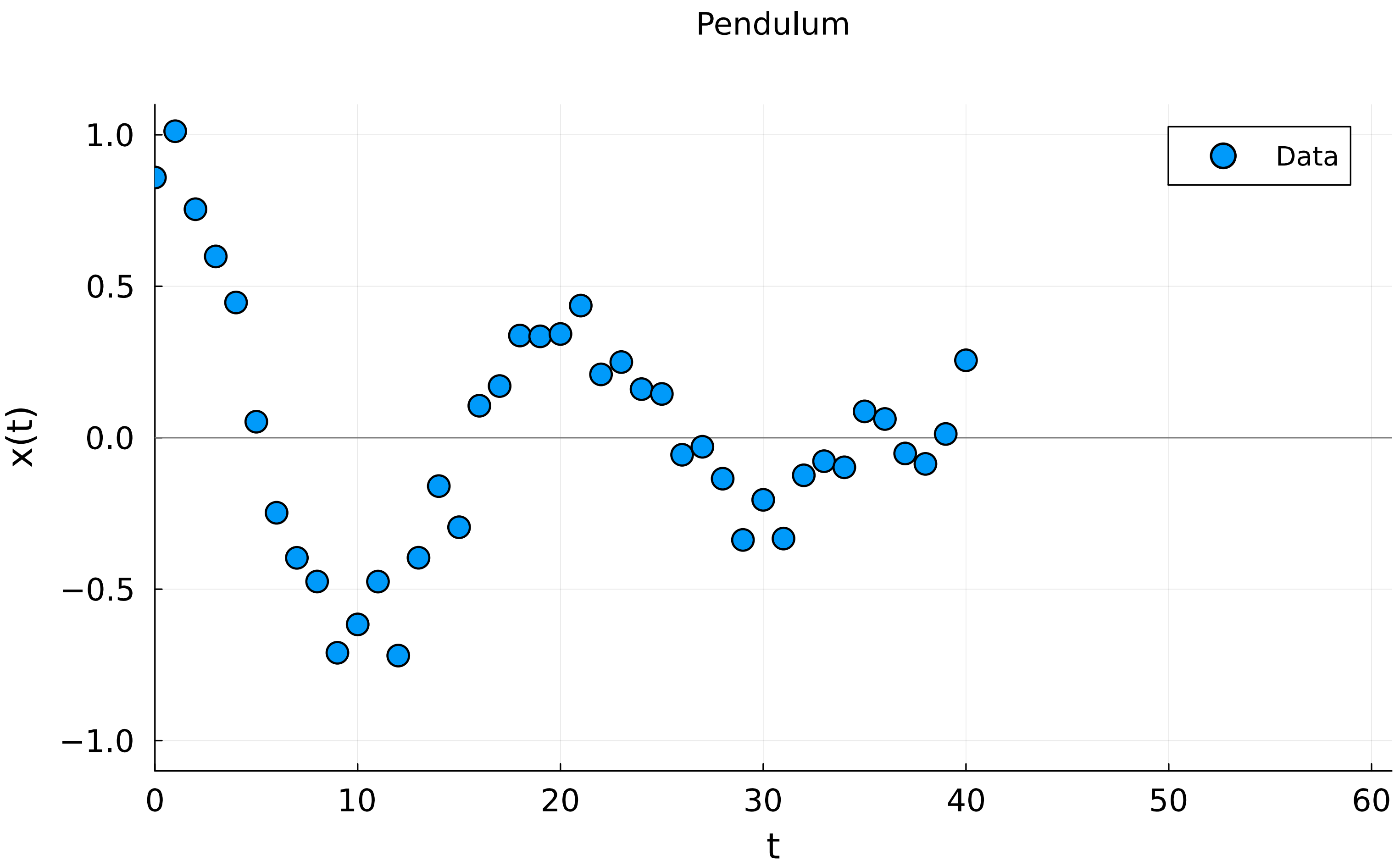
$$m\ddot{x} + kx = 0$$

Harmonic Oscillator?

2^{nd} order ODE → 1^{st} order system:

$$\dot{x} = v$$

$$\dot{v} = -\frac{kx + NN(v)}{m}$$



$$m\ddot{x} + kx = 0$$

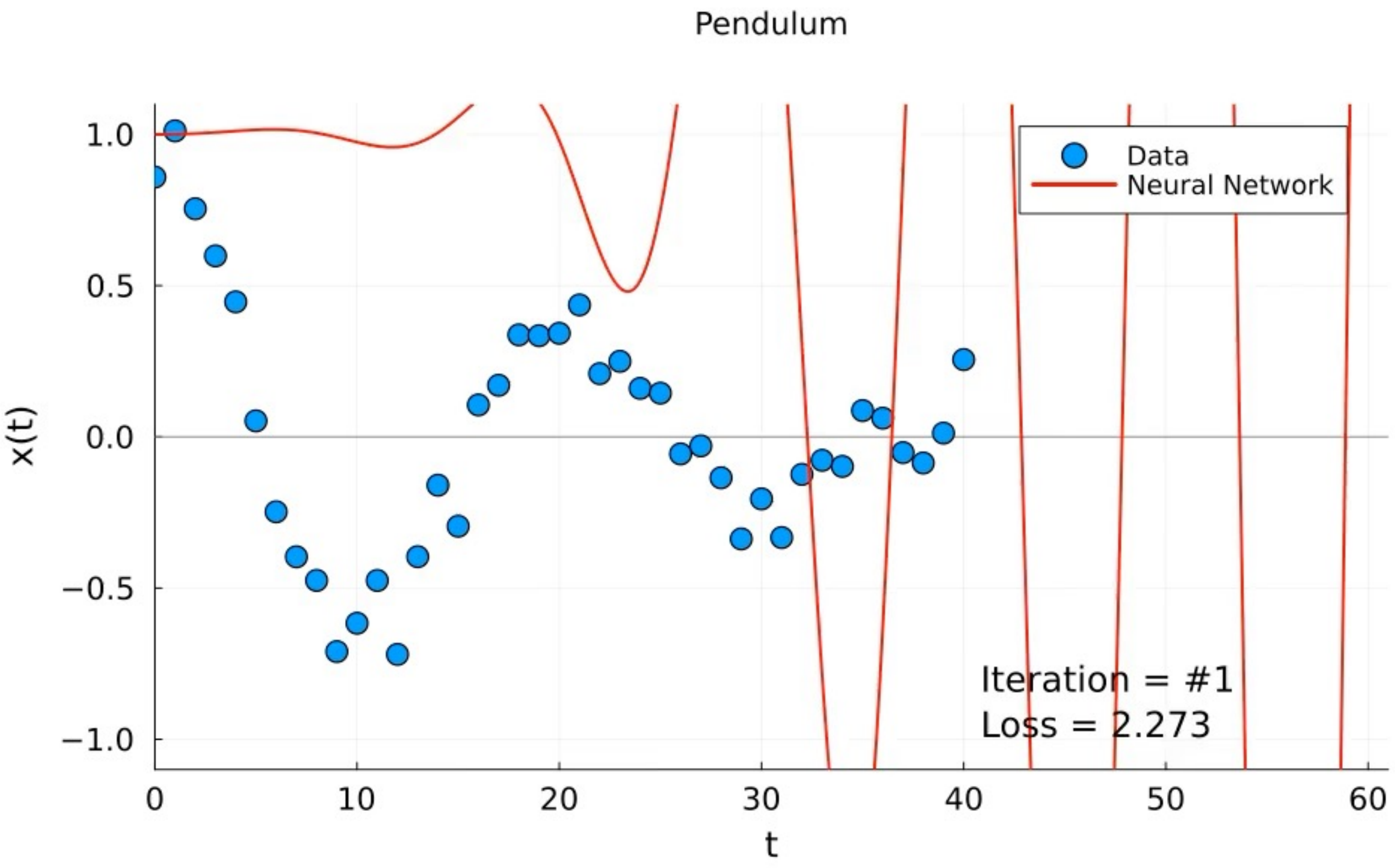
Harmonic Oscillator?

2nd order ODE → 1st order system:

$$\dot{x} = v$$

$$\dot{v} = -\frac{kx + NN(v)}{m}$$

$$m\ddot{x} + kx = 0$$



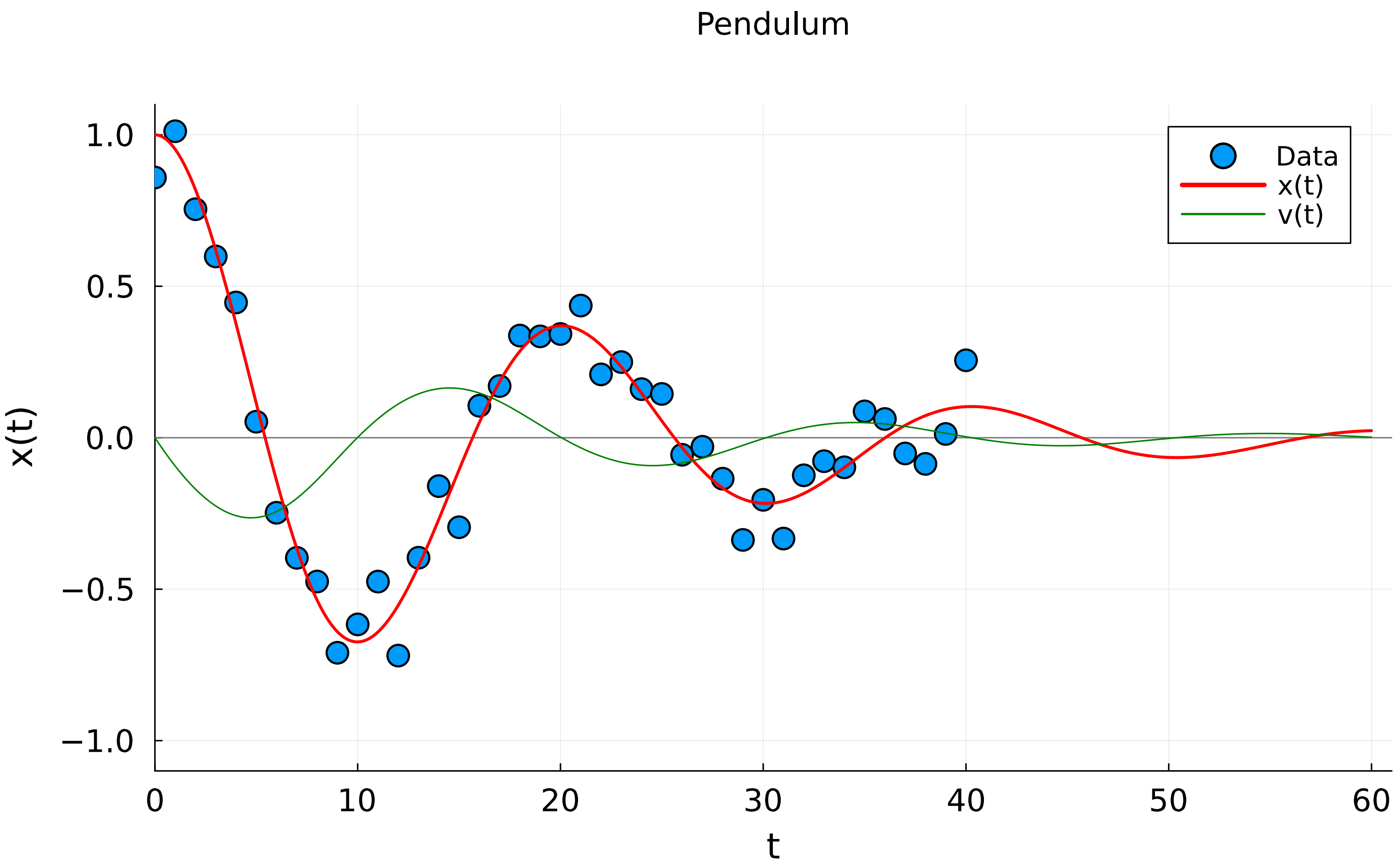
TICRA

Harmonic Oscillator?

2^{nd} order ODE → 1^{st} order system:

$$\dot{x} = v$$

$$\dot{v} = -\frac{kx + NN(v)}{m}$$



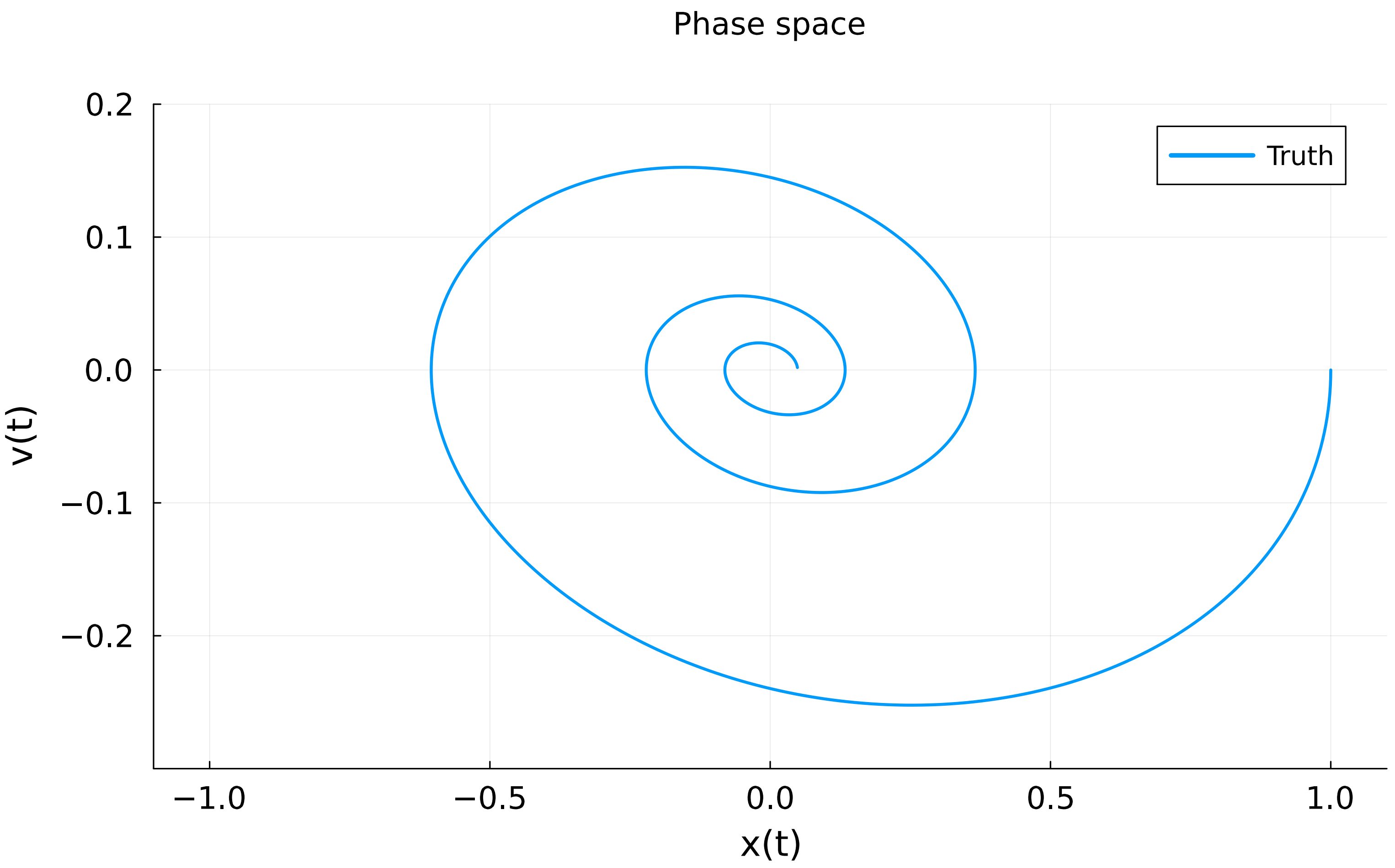
$$m\ddot{x} + kx = 0$$

Harmonic Oscillator?

2^{nd} order ODE → 1^{st} order system:

$$\dot{x} = v$$

$$\dot{v} = -\frac{kx + NN(v)}{m}$$



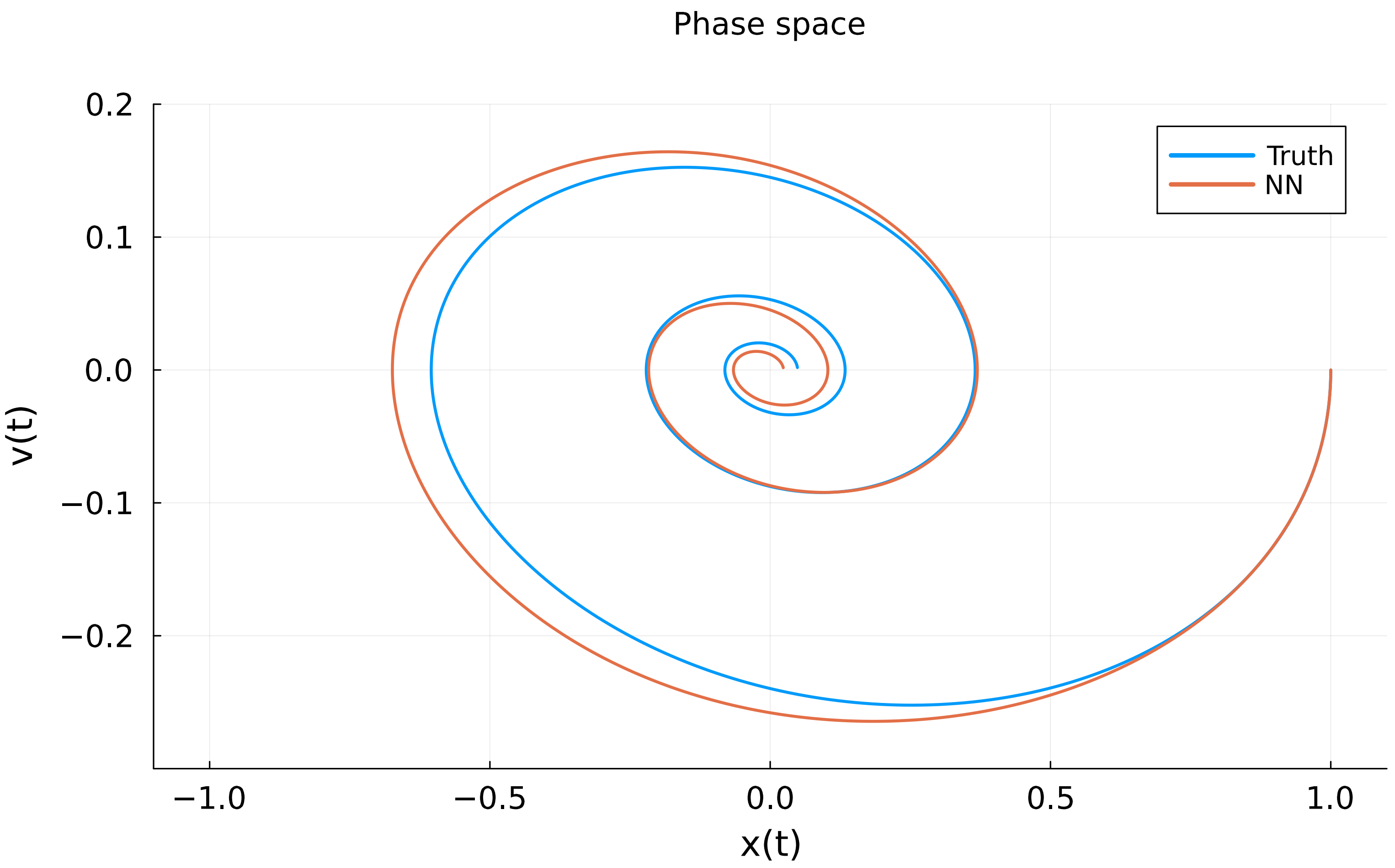
$$m\ddot{x} + kx = 0$$

Harmonic Oscillator?

2nd order ODE → 1st order system:

$$\dot{x} = v$$

$$\dot{v} = -\frac{kx + NN(v)}{m}$$



$$m\ddot{x} + kx = 0$$

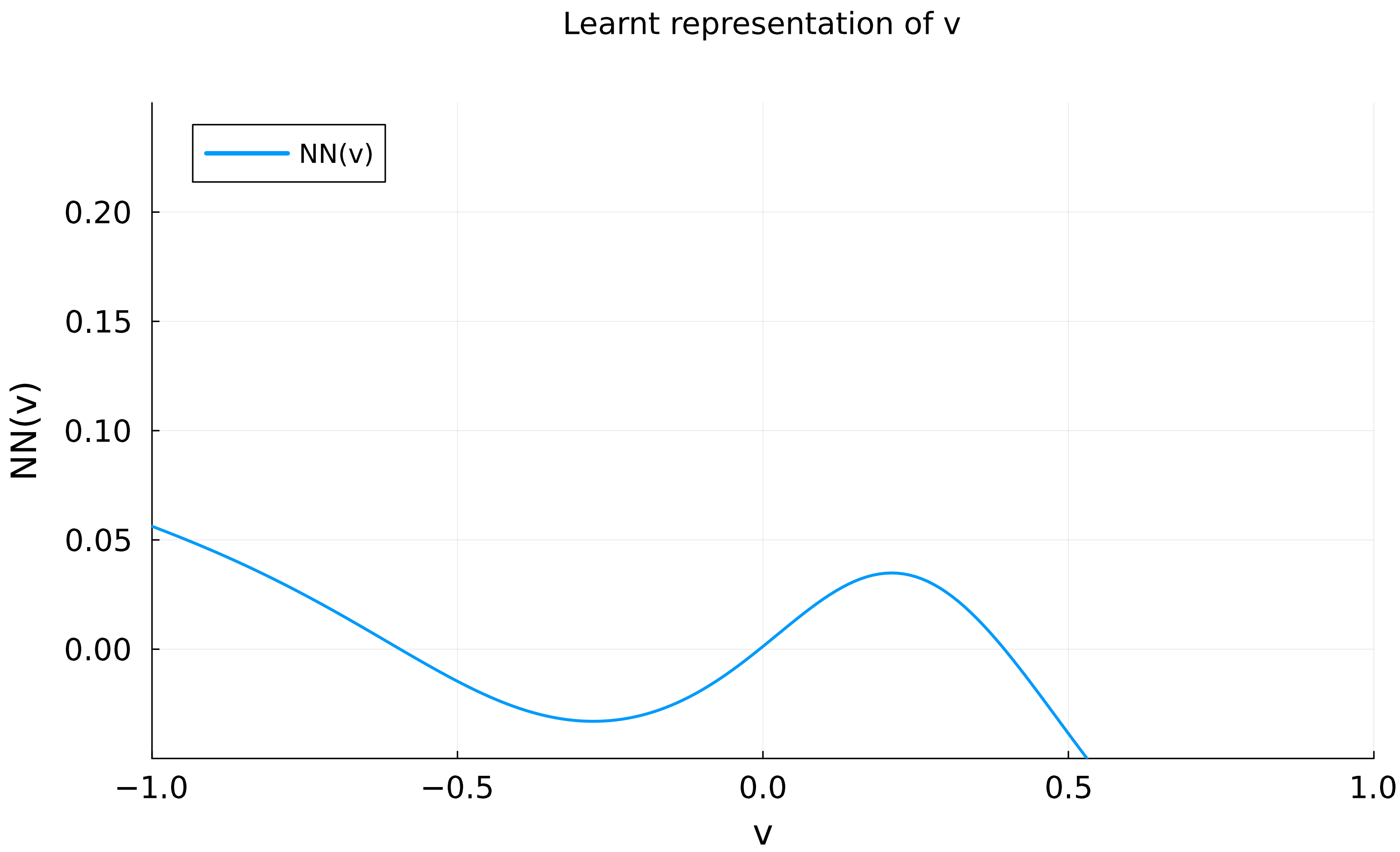
Harmonic Oscillator?

2nd order ODE → 1st order system:

$$\dot{x} = v$$

$$\dot{v} = -\frac{kx + NN(v)}{m}$$

$$m\ddot{x} + kx = 0$$



TICRA

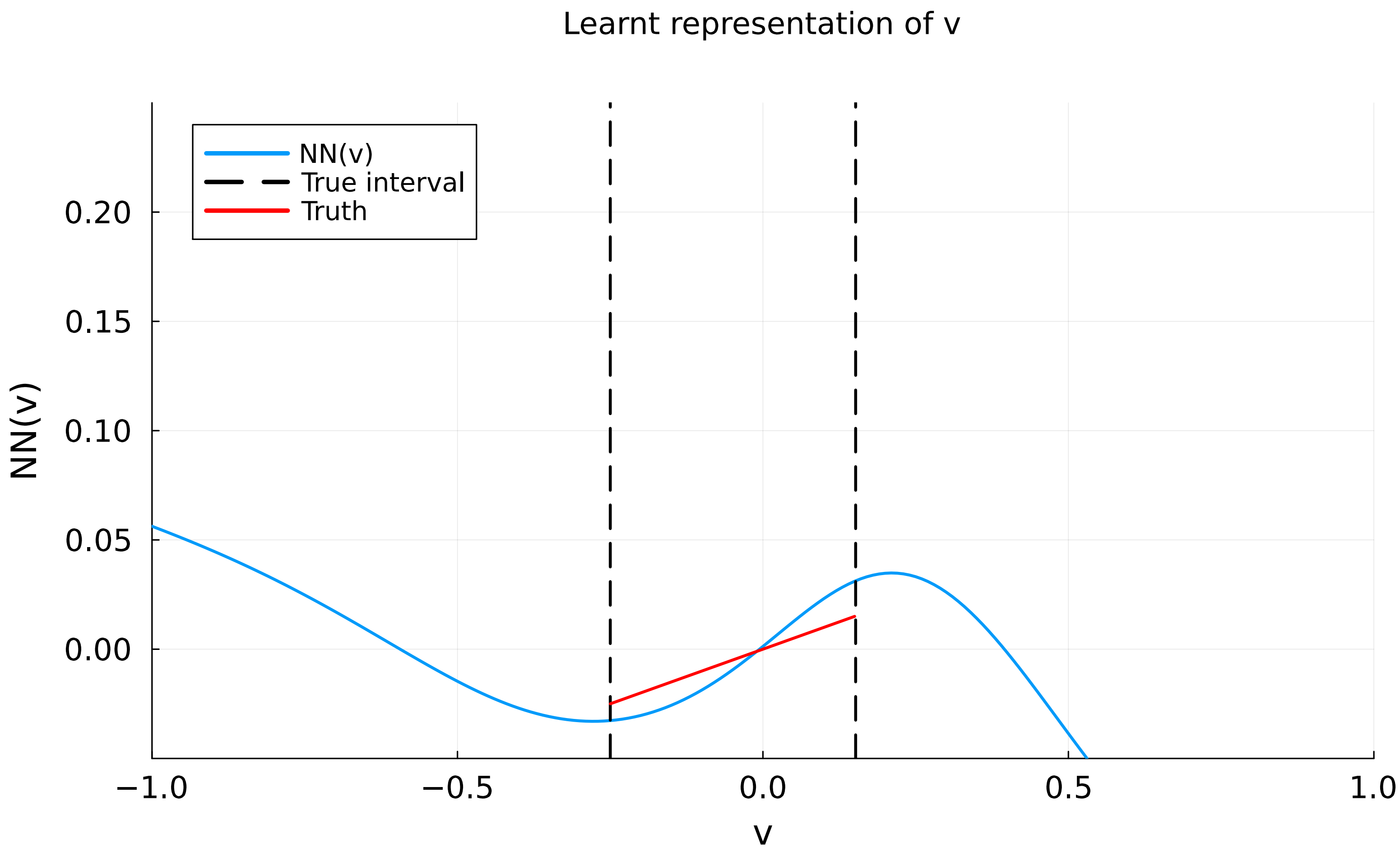
Damped Harmonic Oscillator!

2[nd] order ODE → 1[st] order system:

$$\dot{x} = v$$

$$\dot{v} = -\frac{kx + bv}{m}$$

$$m\ddot{x} + b\dot{x} + kx = 0$$



TICRA

Machine learning

```
function loss_nn(θ)
    y_pred = NN(x_train, θ)
    loss = MSE(y_pred, y_train)
    return loss
end
```

Scientific

```
function damped_harmonic(u, p, t)
    x, v = u          # states
    m, k, b = p       # parameters

    ∂x = v
    ∂v = -1 / m * (k * x + b * v)

    return [∂x, ∂v] # derivatives
end
```

Scientific Machine Learning

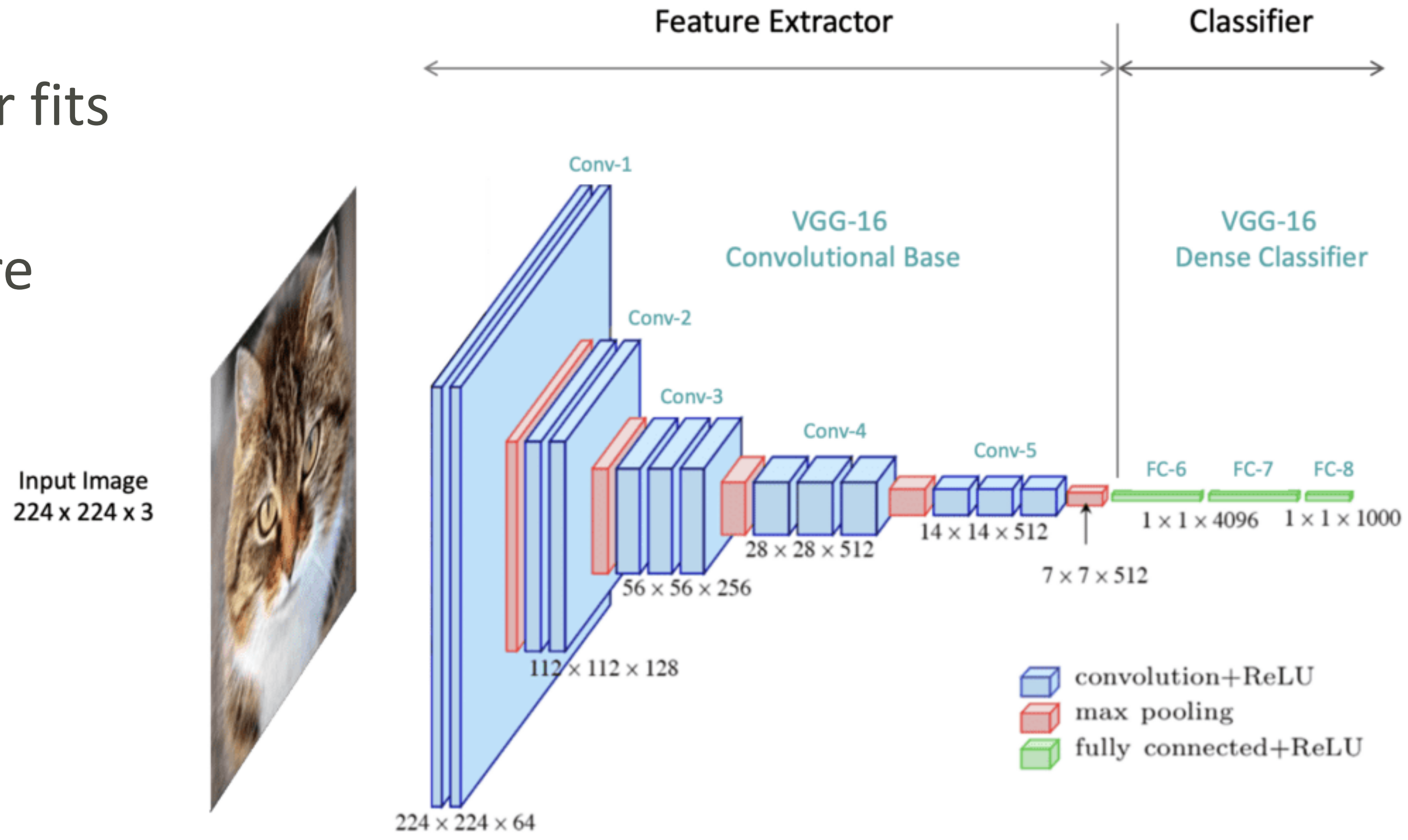
```
function sciml_harmonic(u, p, t)
    x, v = u          # states
    m, k, θ = p       # parameters

    ∂x = v
    ∂v = -1 / m * (k * x + NN(v, θ)

    return [∂x, ∂v] # derivatives
  end
```

Structure

- The major advances in machine learning were due to encoding more structure into the model
- More structure = faster and better fits from less data
- Convolutional Neural Networks are structure assumptions



VGG-16 CNN (ImageNet)

Extrapolation and generalization

- LIGO Black Hole dynamics from the gravitational wave data

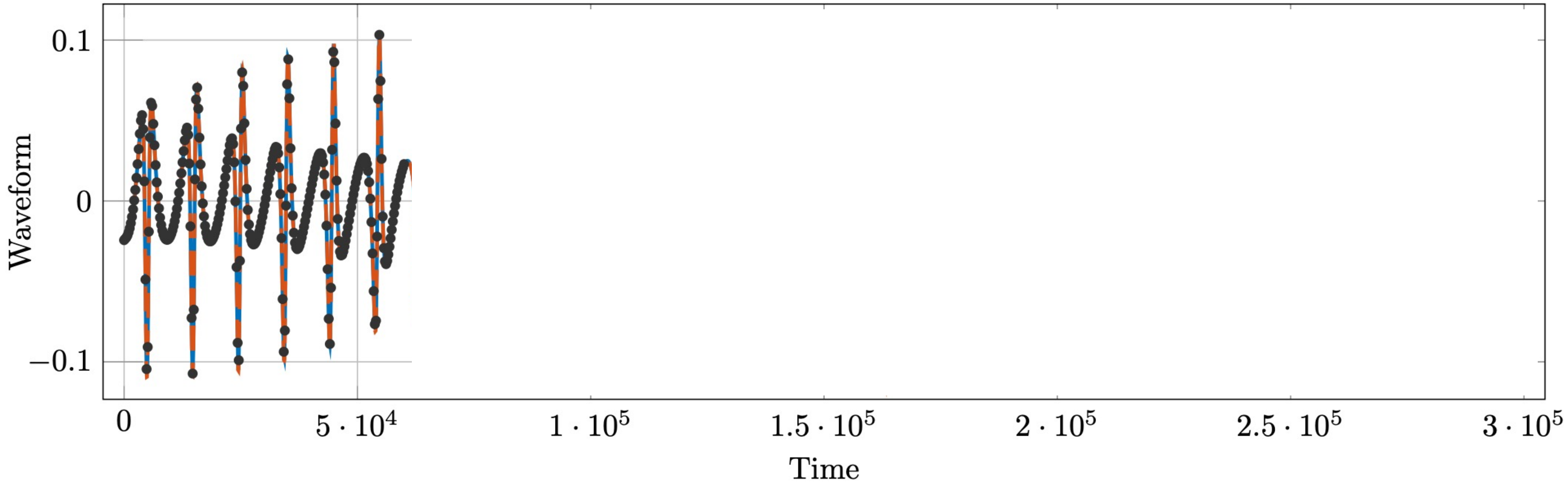
Upon denoting $\mathbf{x} = (\phi, \chi, p, e)$, we propose the following family of UDEs to describe the two-body relativistic dynamics:

$$\dot{\phi} = \frac{(1 + e\cos(\chi))^2}{Mp^{3/2}}\left(1 + \mathcal{F}_1(\cos(\chi), p, e)\right), \tag{5a}$$

$$\dot{\chi} = \frac{(1 + e\cos(\chi))^2}{Mp^{3/2}}\left(1 + \mathcal{F}_2(\cos(\chi), p, e)\right), \tag{5b}$$

$$\dot{p} = \mathcal{F}_3(p, e), \tag{5c}$$

$$\dot{e} = \mathcal{F}_4(p, e), \tag{5d}$$



TICRA

👶 Founded in 1971

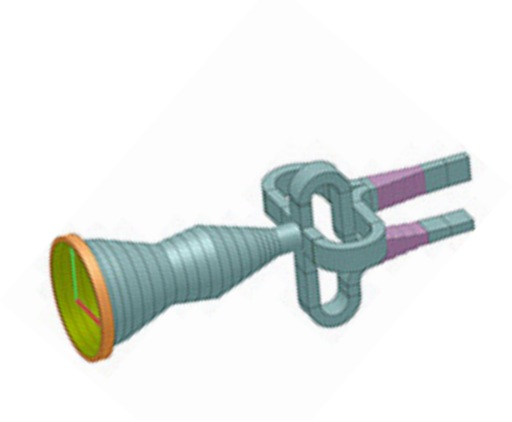
📍 Copenhagen, Denmark

⚡ Electromagnetic radiation

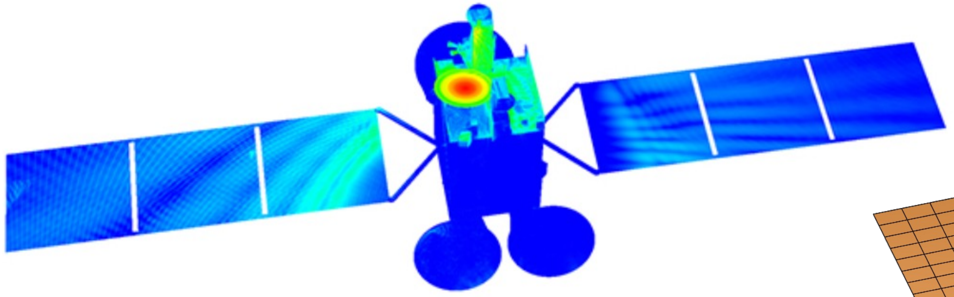
📡 Flagship product: TICRA Tools

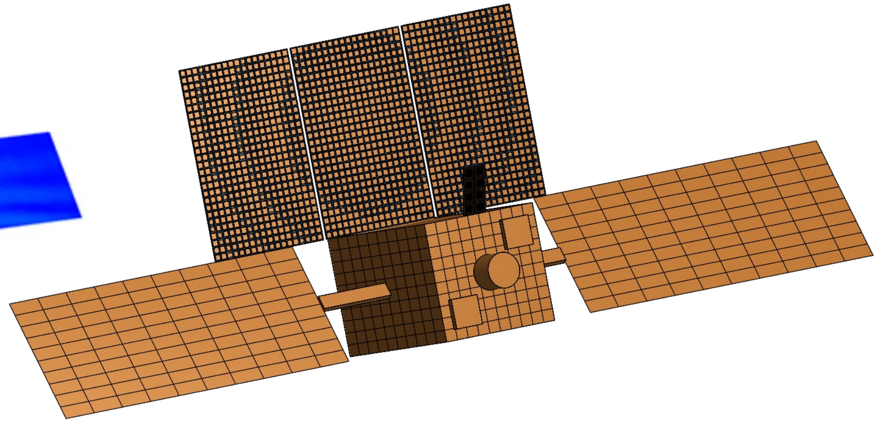
🚀 Long partnership with the European Space Agency (ESA), spacecraft manufacturers, and satellite operators

🤓 50 employees
≥ 80% with MSc
~ 60% with Ph.D.



esa
European Space Agency





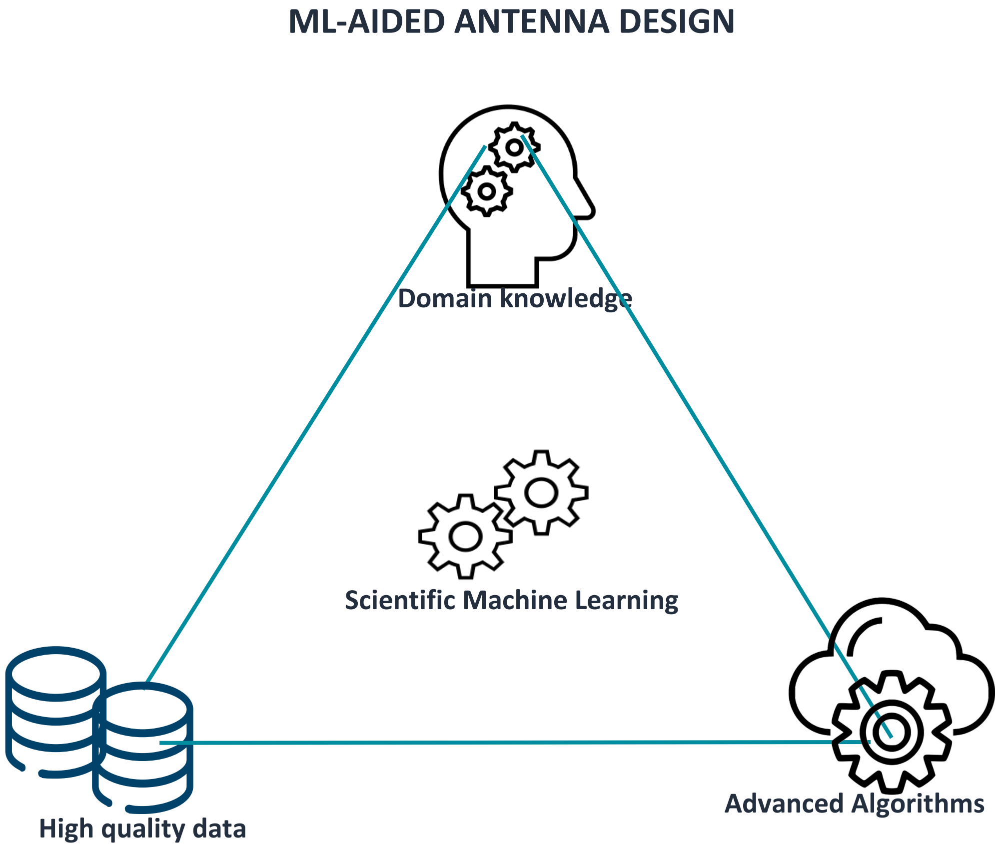
TICRA Core Customers

- Space agencies
- Satellite operators
- Satellite, payload and antenna manufacturers

Design Philosophy – Scientific Machine Learning

Core assets:

- Tailored, state-of-the art simulation and optimisation tools for antenna design
- Solutions to antenna design task that competitors cannot currently solve



Incorporating physics into ML models

2025/06/04

TICRA

Why incorporate physics?

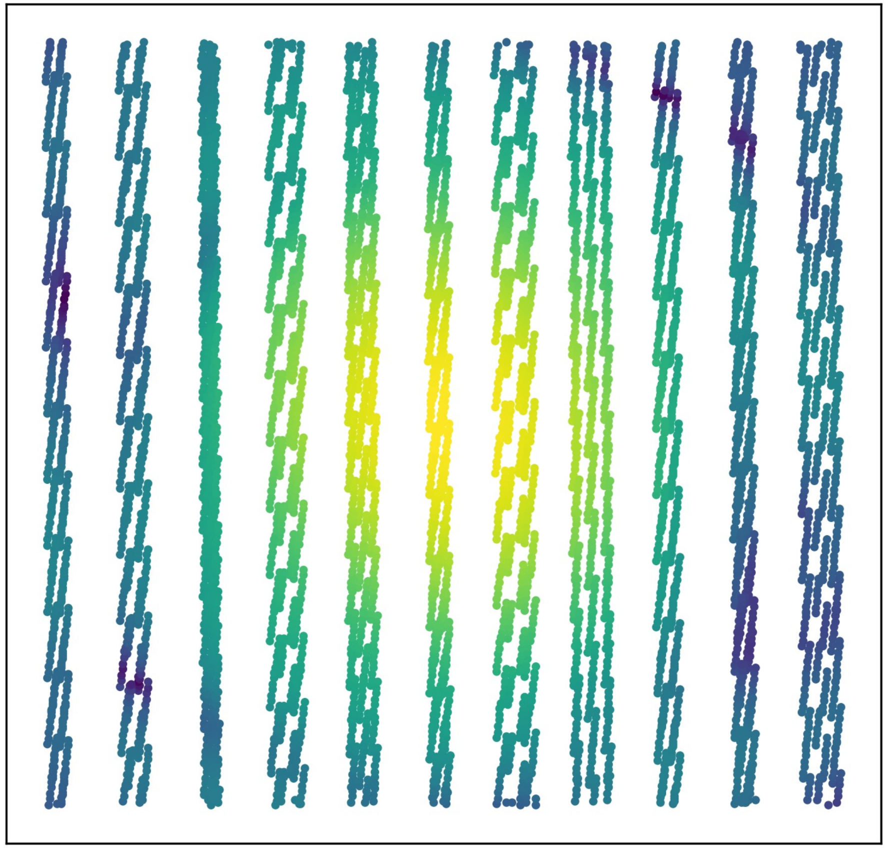
- Traditional methods for solving SciML problems can be slow
  - ML surrogate models could alleviate computational bottlenecks

- SciML data has rich structure that we can take advantage of
  - Data is generated by physics and therefore well-defined (unlike e.g. predicting human behaviour)
  - The physics is often theoretically well-understood

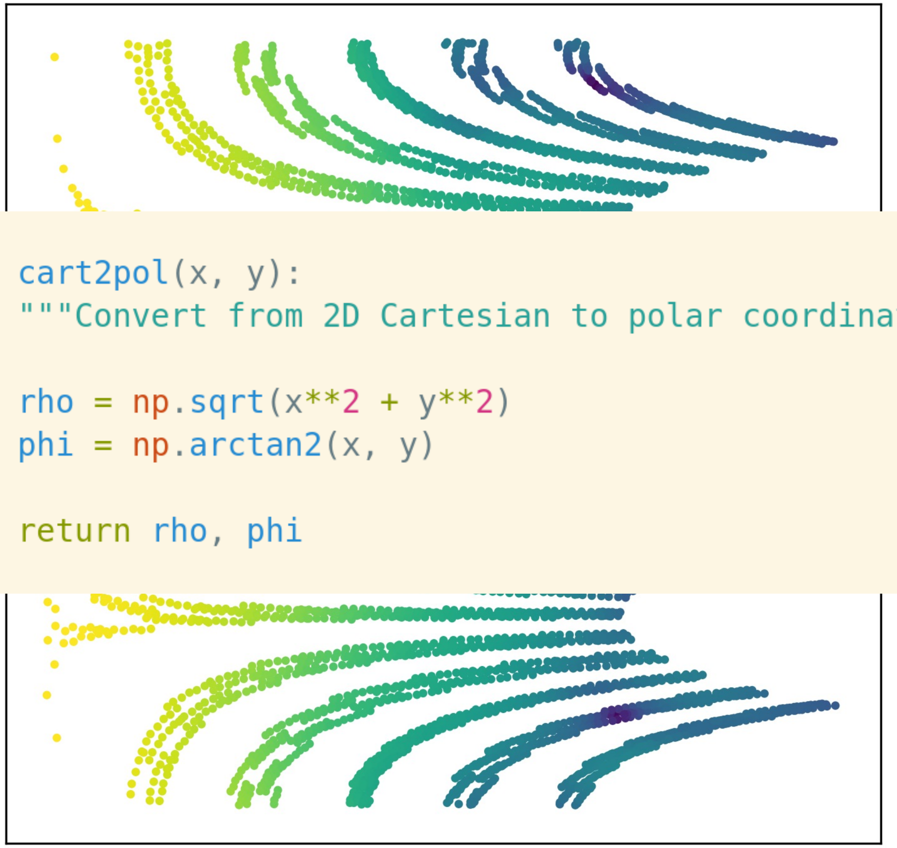
- Incorporating physics makes prediction task easier
  - Shrinks space of possible solutions
  - Acts as a regularizer
  - Can allow for extrapolation, instead of just interpolation
  - Can make models more interpretable and/or trustworthy

Basic example

Original



Transformed



```python
def cart2pol(x, y):
    """Convert from 2D Cartesian to polar coordinates."""

    rho = np.sqrt(x**2 + y**2)
    phi = np.arctan2(x, y)

    return rho, phi
```

PINNs (Physics-informed NN)

Enforcing physics through soft constraints

2025/06/04

TICRA

Enforcing physics through soft constraints: PINNs

- Introduced in 2019

ELSEVIER

Journal of Computational Physics

Volume 378, 1 February 2019, Pages 686-707

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi [a], P. Perdikaris [b], G.E. Karniadakis [a]

- Idea:
  - Given a PDE, e.g. $(\nabla^2 + k^2)\mathbf{E}(\mathbf{x}) = 0$, approximate its solution with a NN
  - Train it to minimize $$\underbrace{|\mathbf{E}^{\text{pred}}(\mathbf{x}_b) - \mathbf{E}^{\text{true}}(\mathbf{x}_b)|^2}_{\text{data loss}} + \underbrace{|(\nabla^2 + k^2)\mathbf{E}^{\text{pred}}(\mathbf{x}_c)|^2}_{\text{PDE loss}}$$

Enforcing physics through soft constraints: PINNs

>14k citations, making it the most cited numerical methods paper of the 21st century

Still, lots of problems:

- Training instabilities, especially in high-frequency domain
- Only works for small networks (typically less than 0.5M parameters)
- No accuracy guarantees, as opposed to traditional numerical methods
- Uses expensive second order optimization methods (L-BFGS), which has implications for activations functions
- etc.

Characterizing possible failure modes in physics-informed neural networks

Aditi S. Krishnapriyan[*,1,2], **Amir Gholami**[*,2], **Shandian Zhe**[3], **Robert M. Kirby**[3], **Michael W. Mahoney**[2,4]

Satisfying physical laws

Enforcing physics through hard constraints

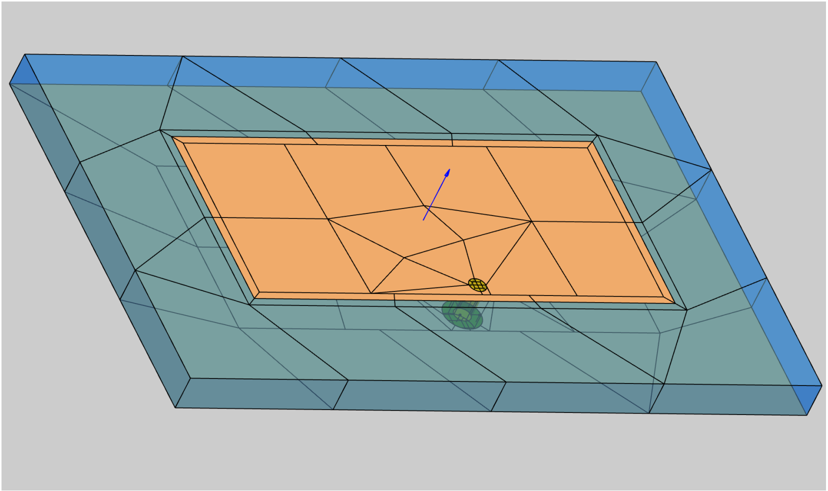
2025/06/04

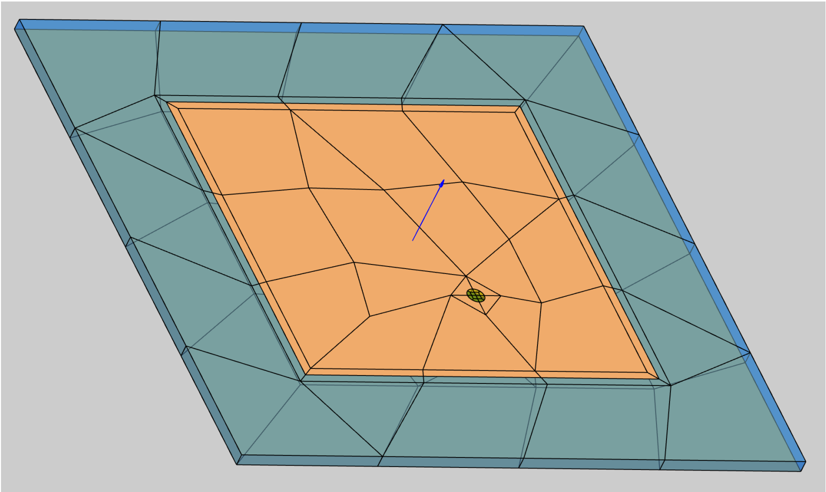
Enforcing physics through hard constraints: Via problem formulation

- Suppose you were designing an antenna, trying out different designs

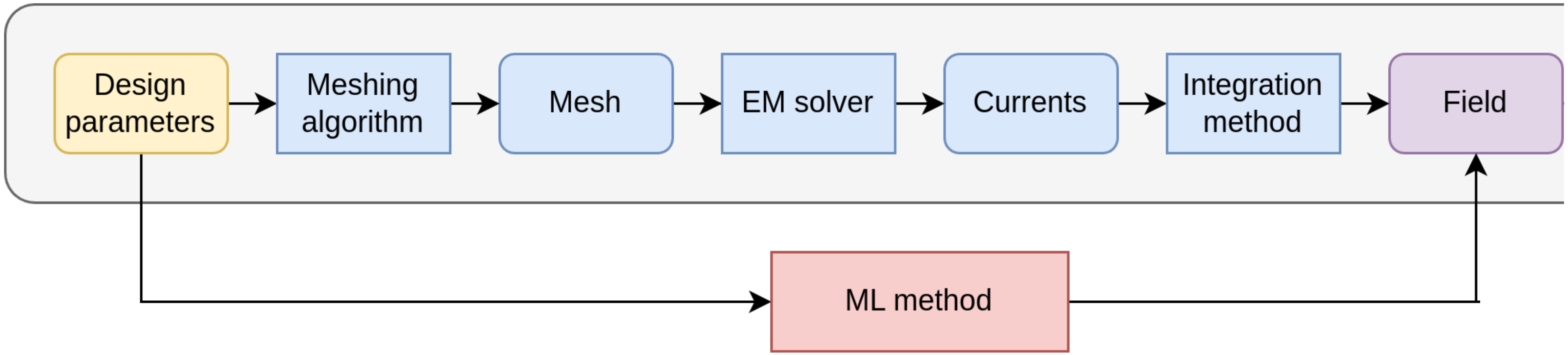
- Every change requires you to wait *minutes* before you can see if it got better

- Train a surrogate model to predict the antenna's EM field from the design parameters!





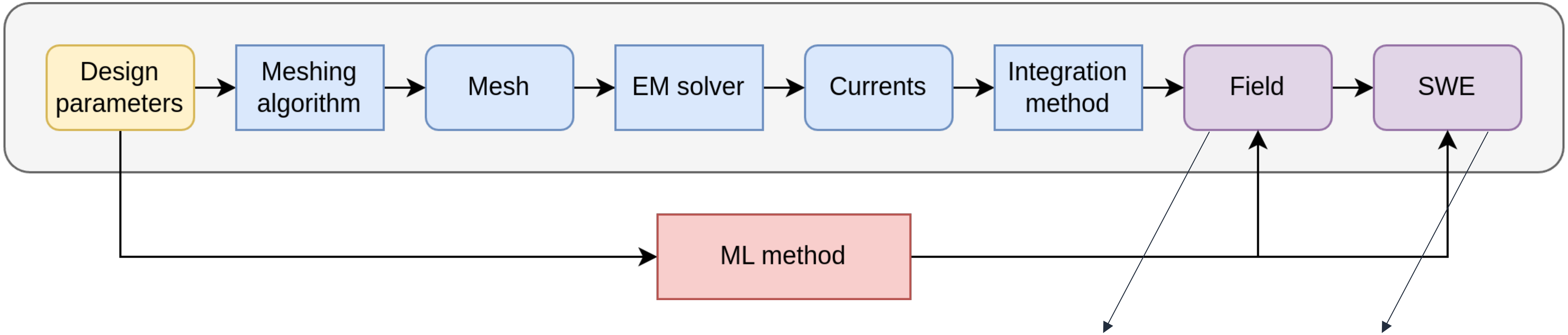
Enforcing physics through hard constraints: Via problem formulation



From design parameters, predict

- EM field in grid (3264 predicted variables)

Enforcing physics through hard constraints: Via problem formulation

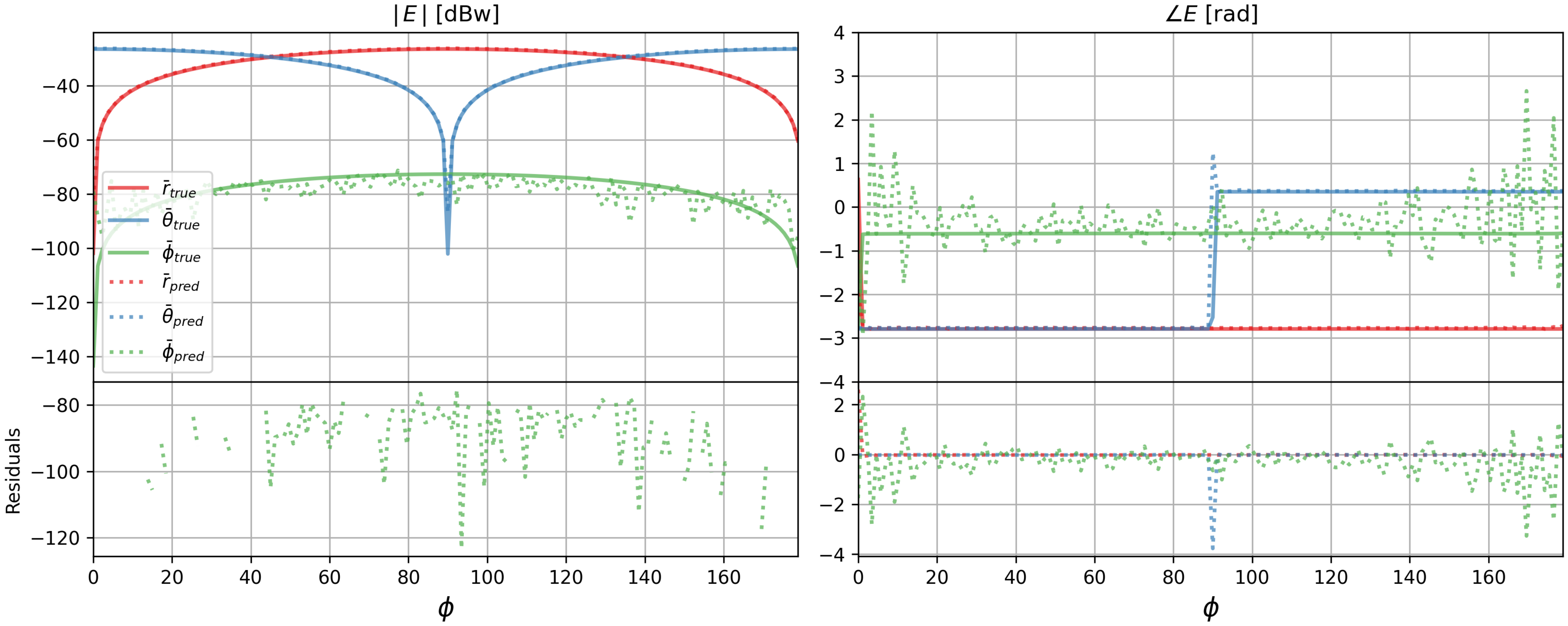


$$\bar{E}(r,\theta,\phi) = k\sqrt{\zeta}\sum_{smn} Q_{smn}\bar{F}_{smn}(r,\theta,\phi)$$

From design parameters, predict

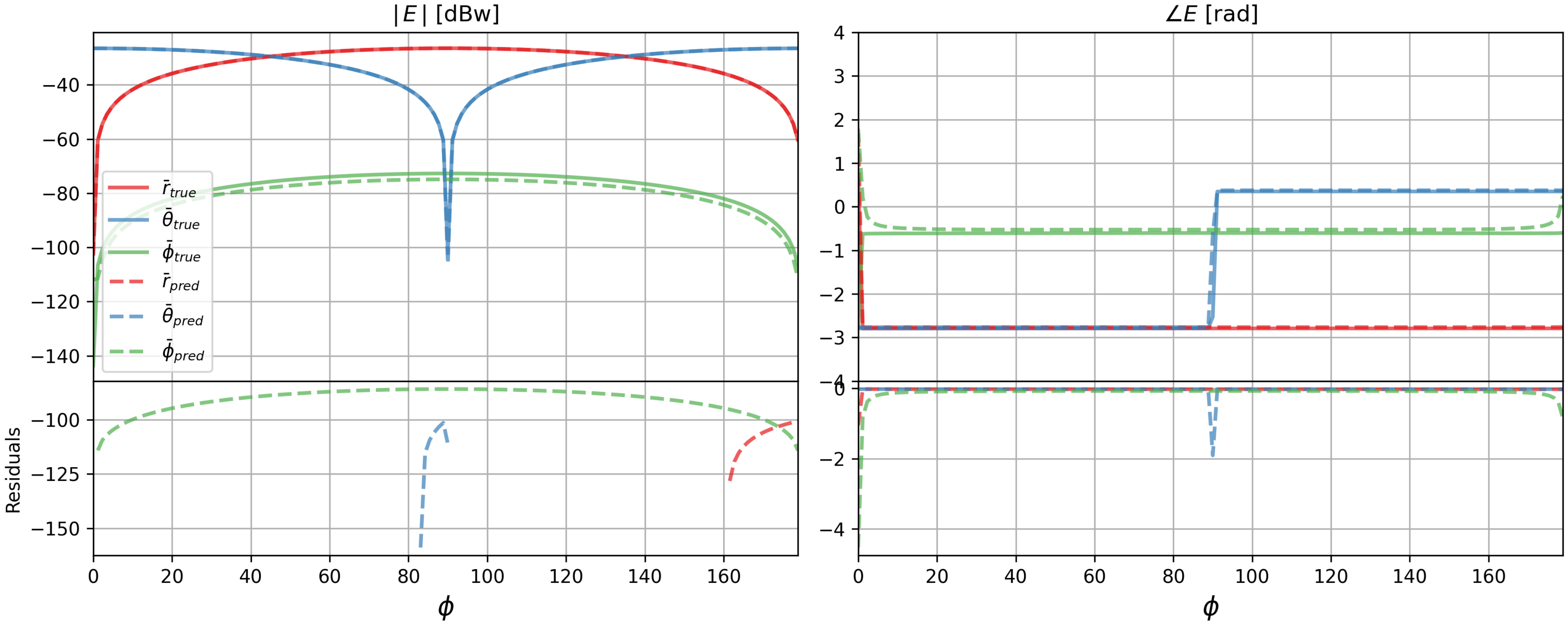
- EM field in grid (3264 predicted variables)
- Expansion (SWE) coefficients encoding the EM field (96 predicted variables)
  - Compressed representation
  - Guarantees that the predicted field is a solution to Maxwell's equations (!!)

Enforcing physics through hard constraints: Via problem formulation



Good performance, but quite jittery/unphysical

Enforcing physics through hard constraints: Via problem formulation



Still good performance, but much more physically correct
(this stuff matters to domain experts!)

Thank you!

Christian Buus Michelsen
Senior ML Engineer @ TICRA
Scientific Machine Learning 2025-06-04
https://github.com/ChristianMichelsen/SciMLPresentation