Scientific Machine Learning

Christian Michelsen

Ph.D. student @ Troels Petersen

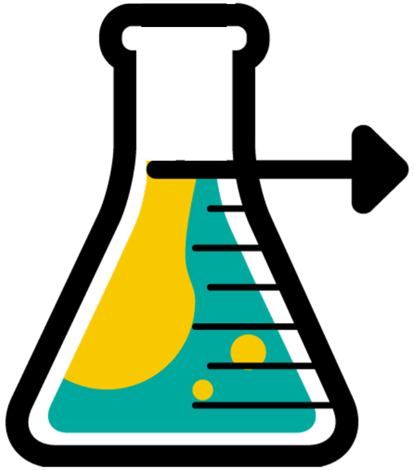
Chris Rackauckas

Director of Modeling and Simulation,
Julia Computing

Research Affiliate, Co-PI of Julia Lab,
Massachusetts Institute of Technology, Mathematics

Scientific Machine Learning is model-based data-efficient machine learning

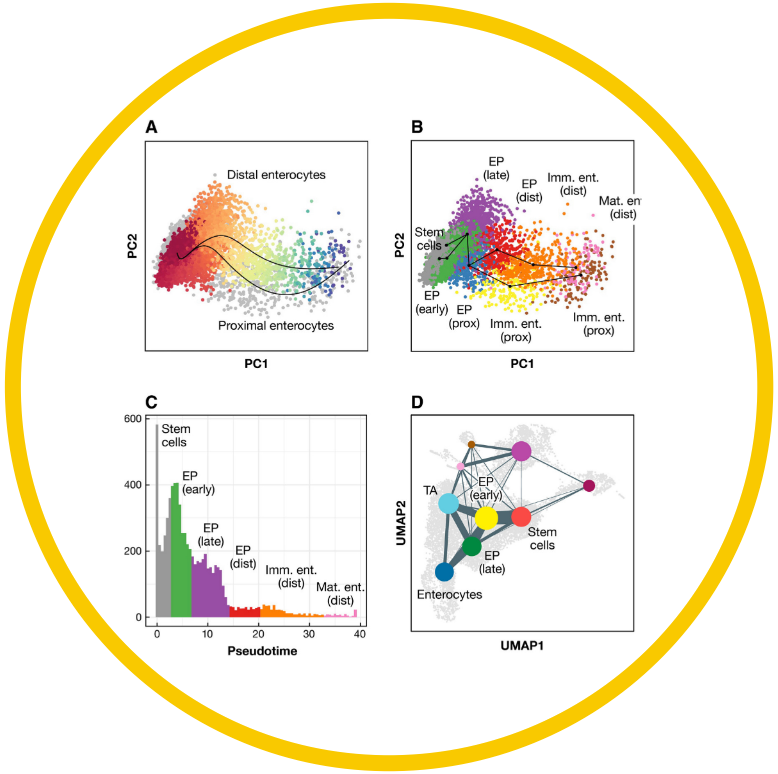
How do we simultaneously use both sources of knowledge?

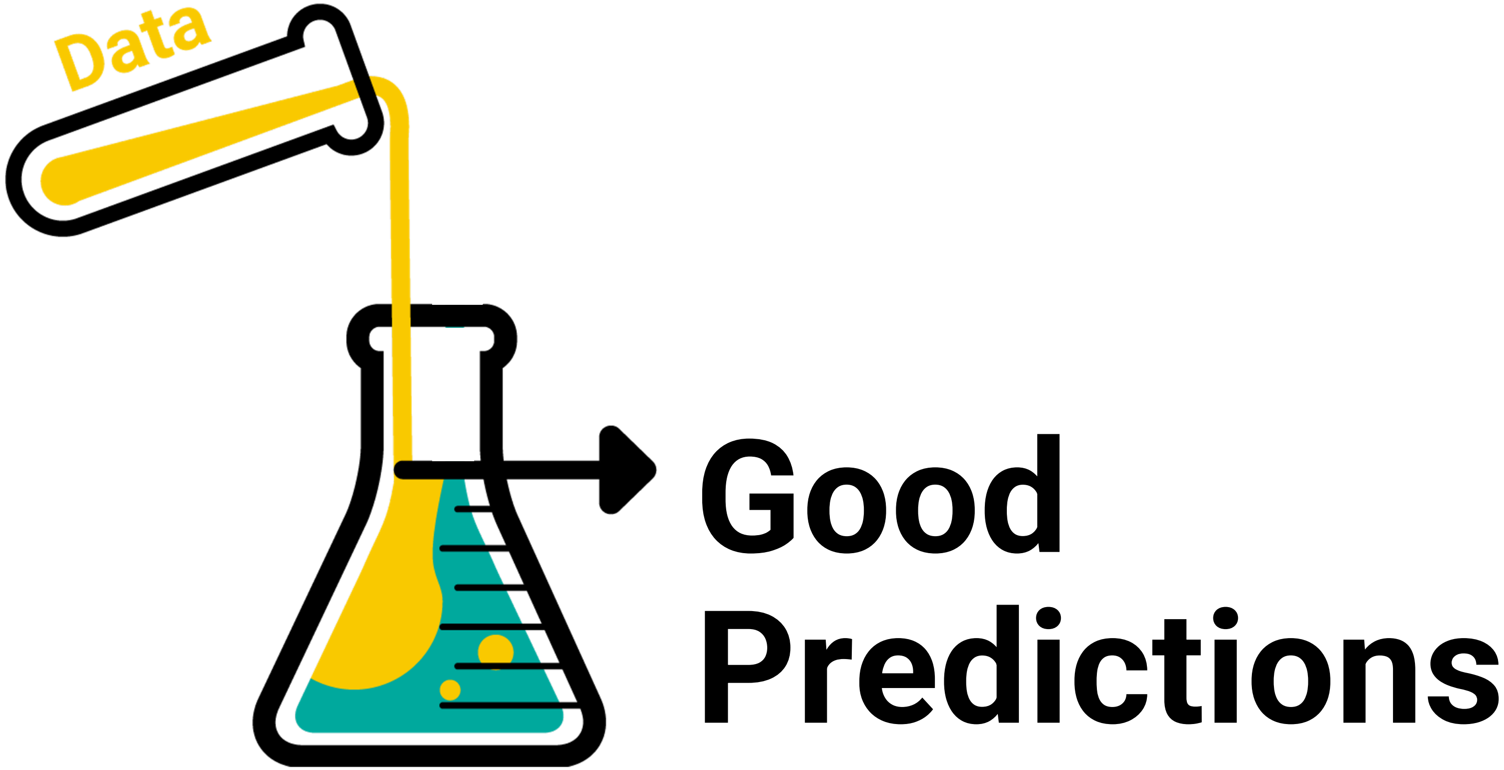


Good Predictions

Scientific Machine Learning is model-based data-efficient machine learning

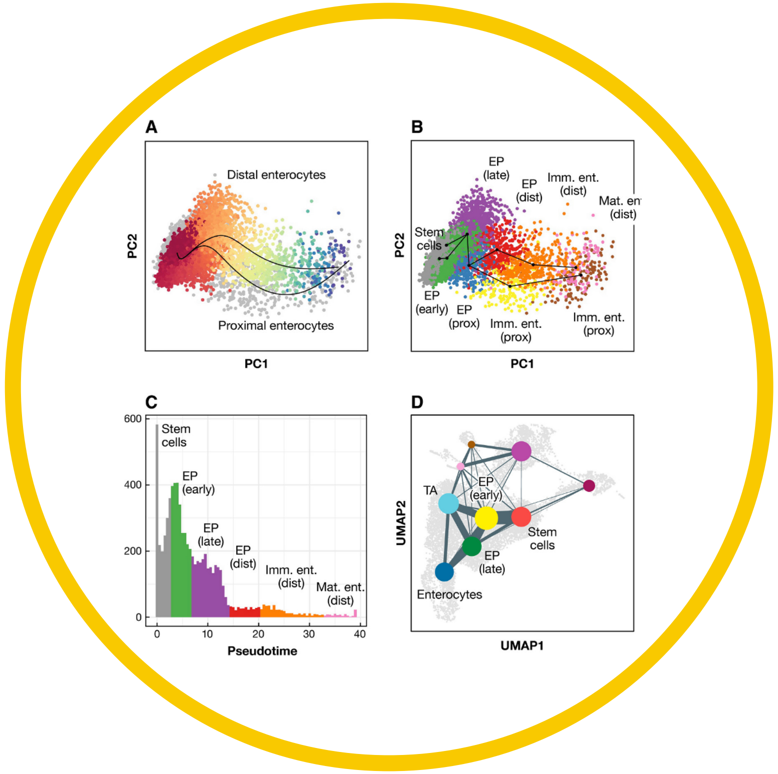
How do we simultaneously use both sources of knowledge?

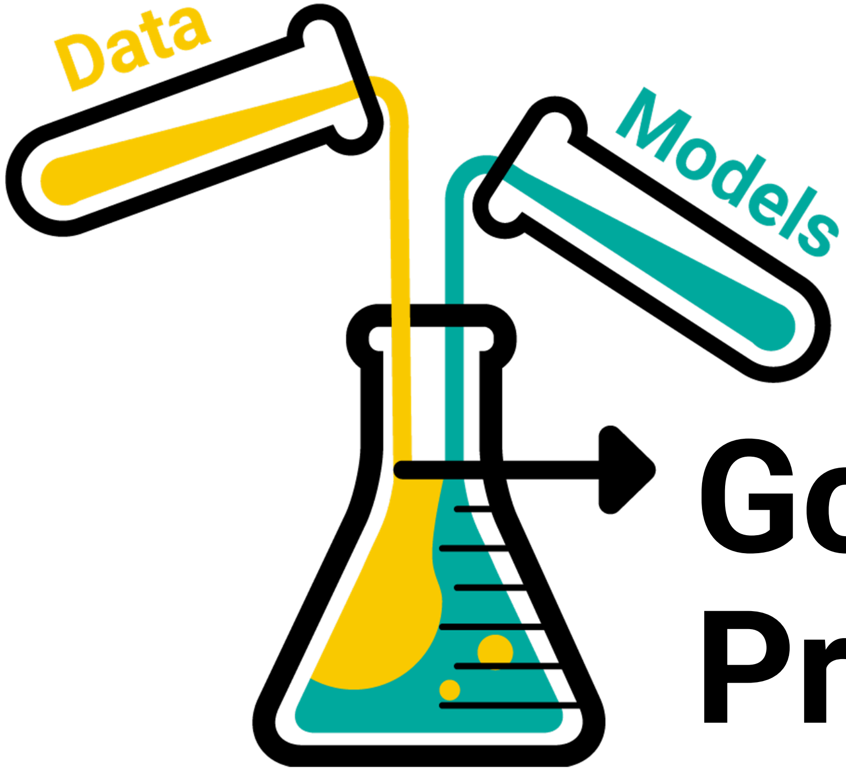




Scientific Machine Learning is model-based data-efficient machine learning

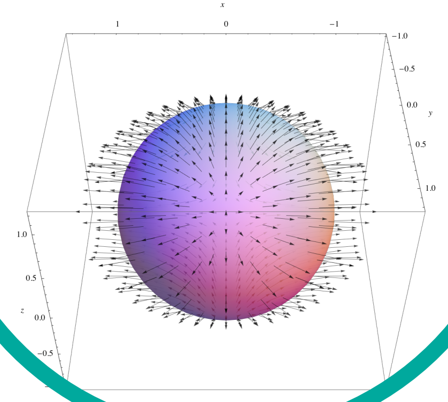
How do we simultaneously use both sources of knowledge?





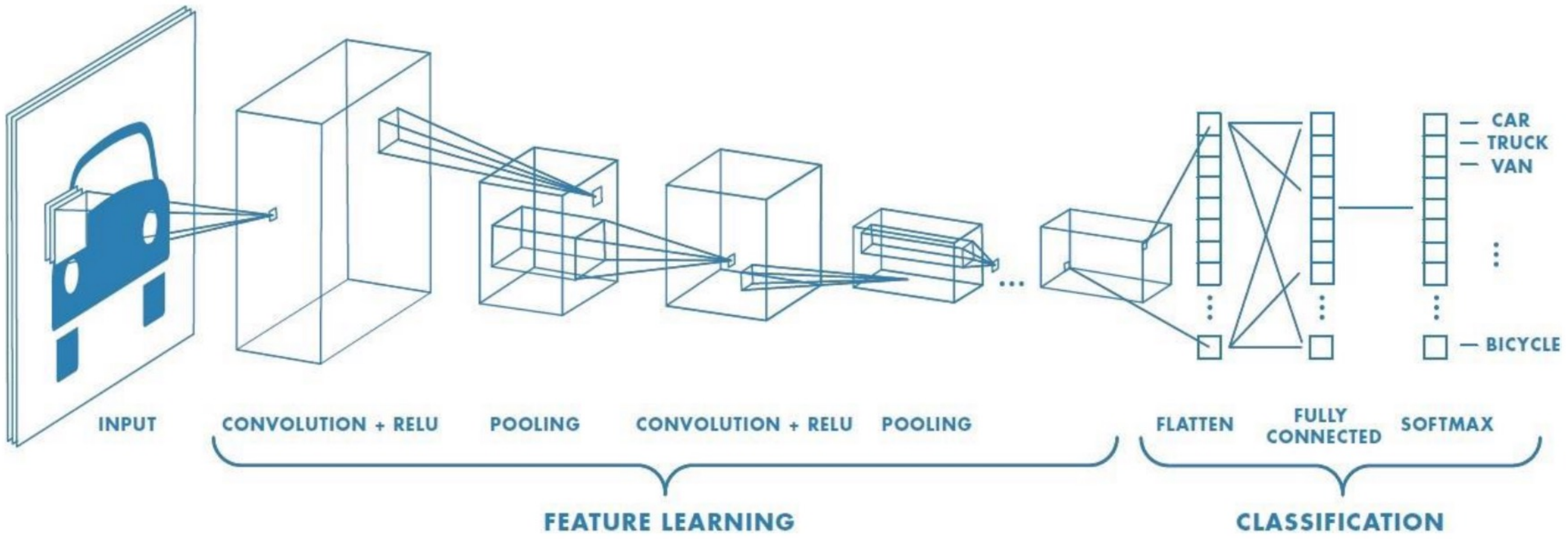
Good Predictions

$$\frac{\mathrm{d}x}{\mathrm{d}t} = x e^{t}$$



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



Lotka-Volterra

Predator–prey equations (1910-1920)

$$\frac{d}{dt}🐰 = \alpha 🐰$$

Exponential growth

Lotka-Volterra

Predator–prey equations (1910-1920)

$$\frac{d}{dt}🐰 = \alpha 🐰 \quad - \xi_1 🐰🐺$$

Exponential growth

Eaten by wolves

Lotka-Volterra

Predator–prey equations (1910-1920)

$$\frac{d}{dt}\text{🐰} = \alpha\,\text{🐰} \quad - \xi_1 \text{🐰}\text{🐺}$$

Exponential growth

Eaten by wolves

$$\frac{d}{dt}\text{🐺} = -\theta_1 \text{🐺}$$

Competition

Lotka-Volterra

Predator–prey equations (1910-1920)

$$\frac{d}{dt}🐰 = \alpha 🐰 - \xi_1 🐰 🐺$$

Exponential growth

Eaten by wolves

$$\frac{d}{dt}🐺 = -\theta_1 🐺 + \xi_2 🐰 🐺$$

Competition

More food

Lotka-Volterra

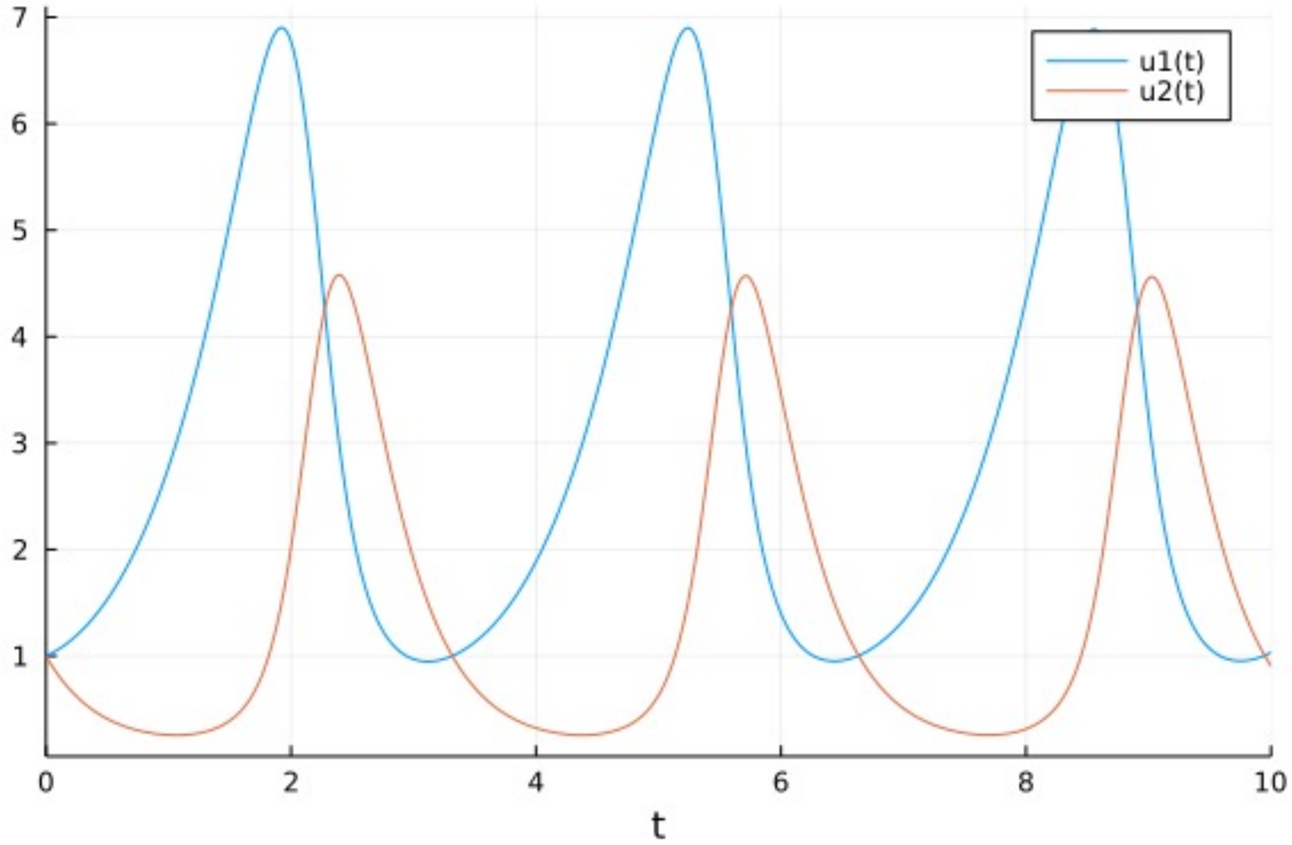
Predator–prey equations (1910-1920)

$$\frac{d}{dt}\text{🐰} = \alpha\,\text{🐰} - \xi_1\,\text{🐰}\,\text{🐺}$$

Exponential growth — Eaten by wolves

$$\frac{d}{dt}\text{🐺} = -\theta_1\,\text{🐺} + \xi_2\,\text{🐰}\,\text{🐺}$$

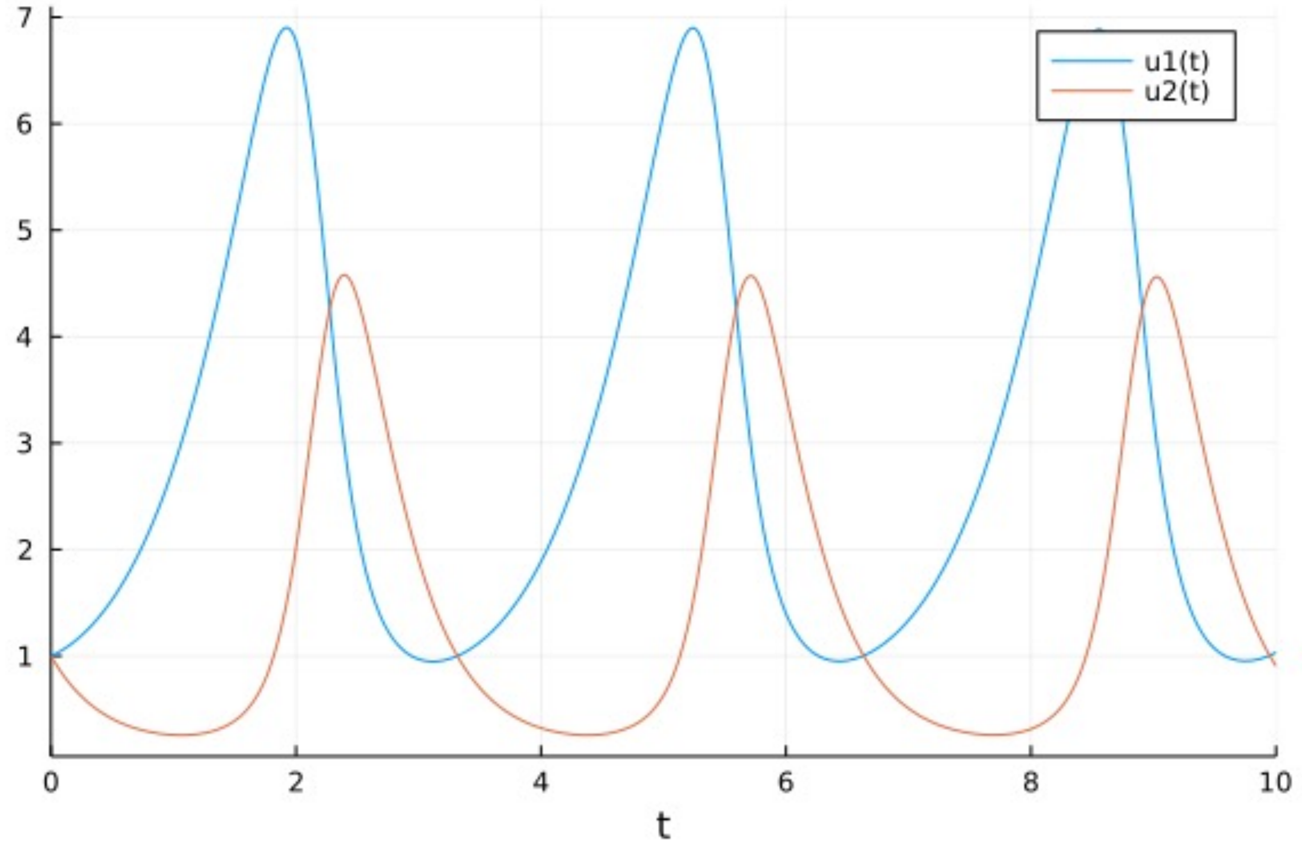
Competition — More food



Lotka-Volterra

Predator–prey equations (1910-1920)

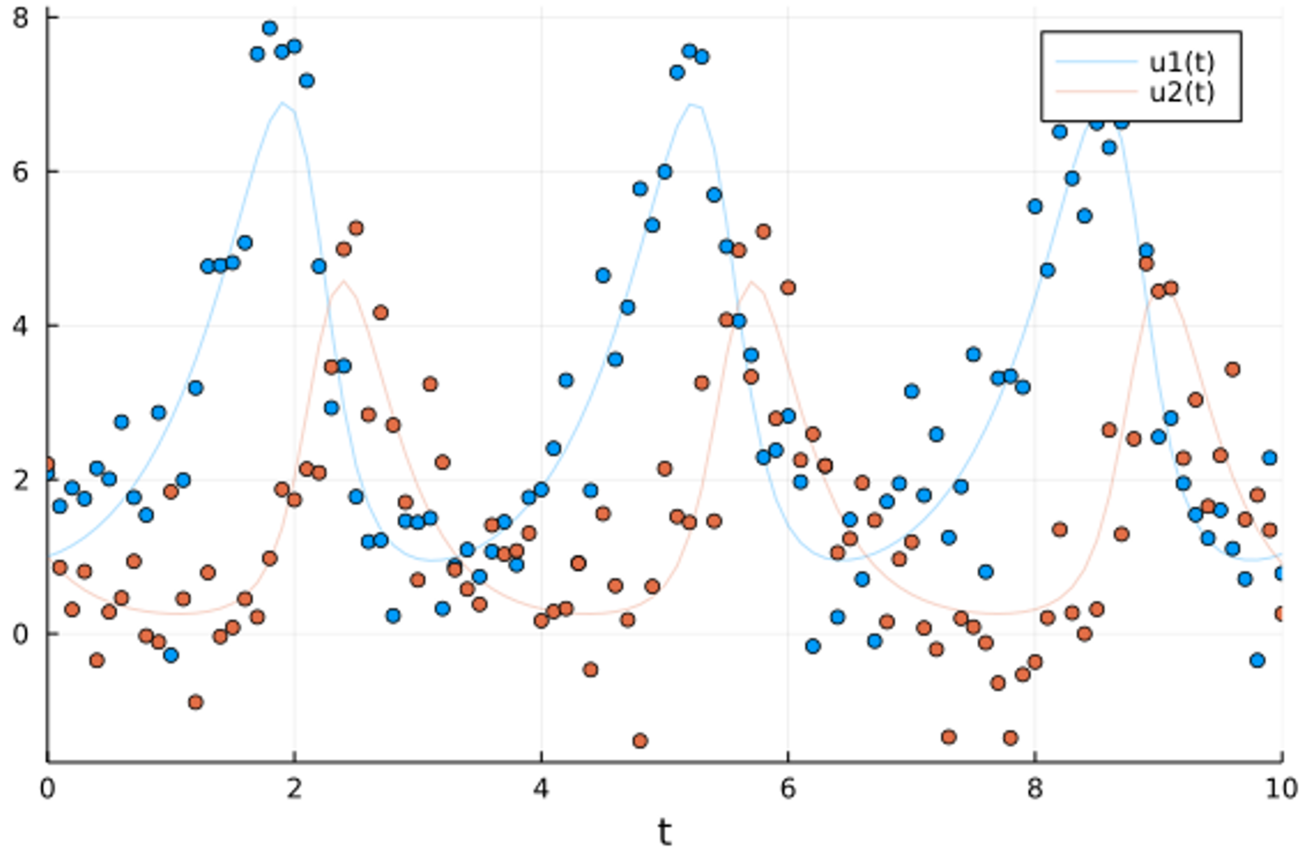
$$\dot{x} = \alpha x \quad - \xi_1 xy$$
$$\dot{y} = -\theta_1 y + \xi_2 xy$$



Lotka-Volterra

Predator–prey equations (1910-1920)

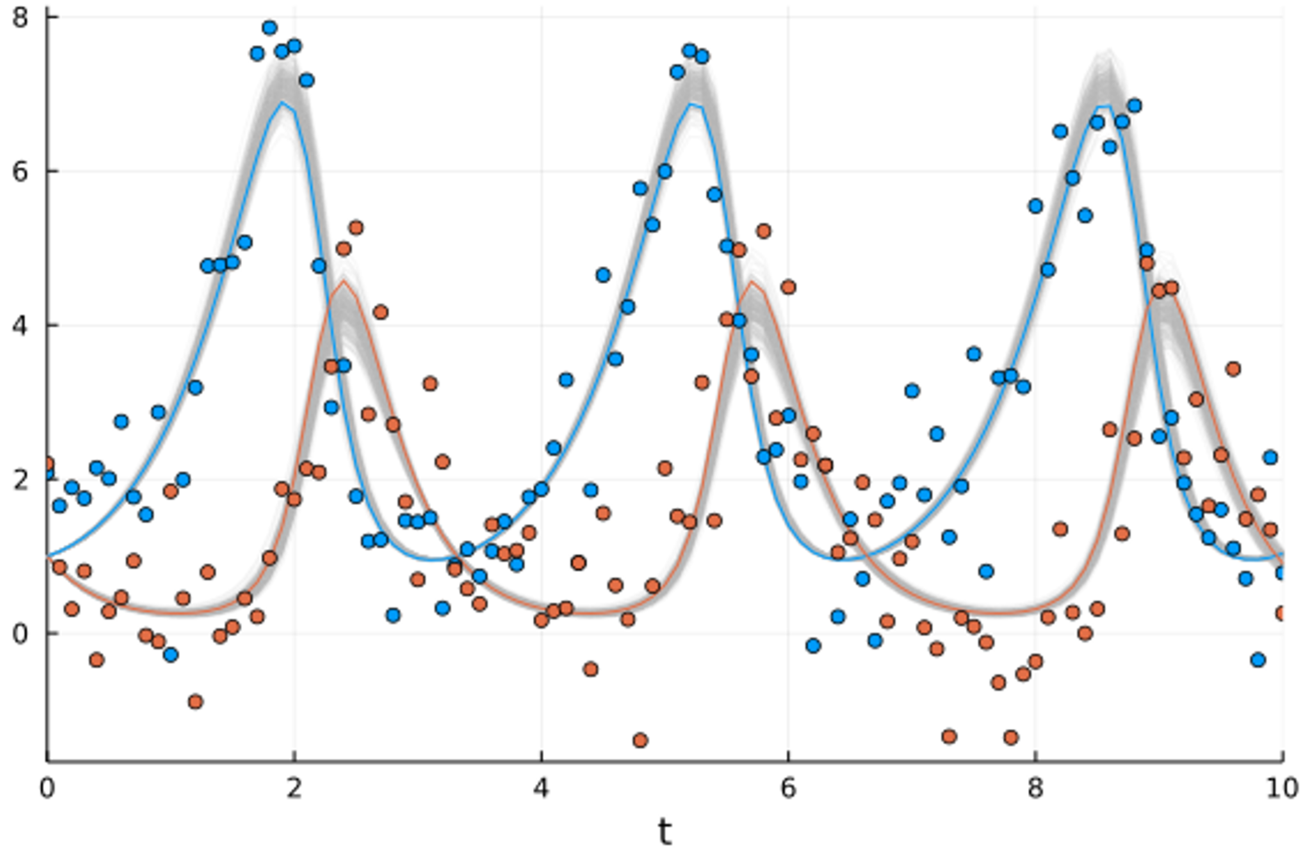
$$\dot{x} = \alpha x \quad - \xi_1 xy$$
$$\dot{y} = -\theta_1 y + \xi_2 xy$$



Lotka-Volterra

Predator–prey equations (1910-1920)

$$\dot{x} = \alpha x \quad - \xi_1 xy$$
$$\dot{y} = -\theta_1 y + \xi_2 xy$$



Turing.jl

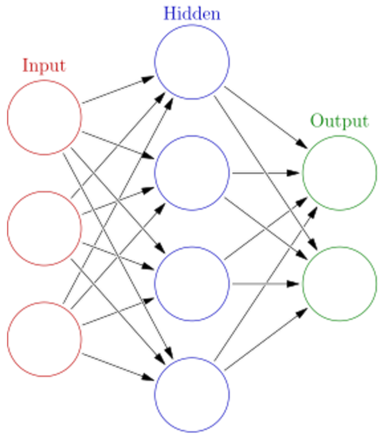
Unknown DEs

Predator–prey equations (1910-1920)

Function approximators?

$$\dot{x} = \alpha x - \xi_1 xy$$
$$\dot{y} = -\theta_1 y + \xi_2 xy$$

$$\dot{x} = \alpha x -$$
$$\dot{y} = -\theta_1 y +$$



Workflow

Model:

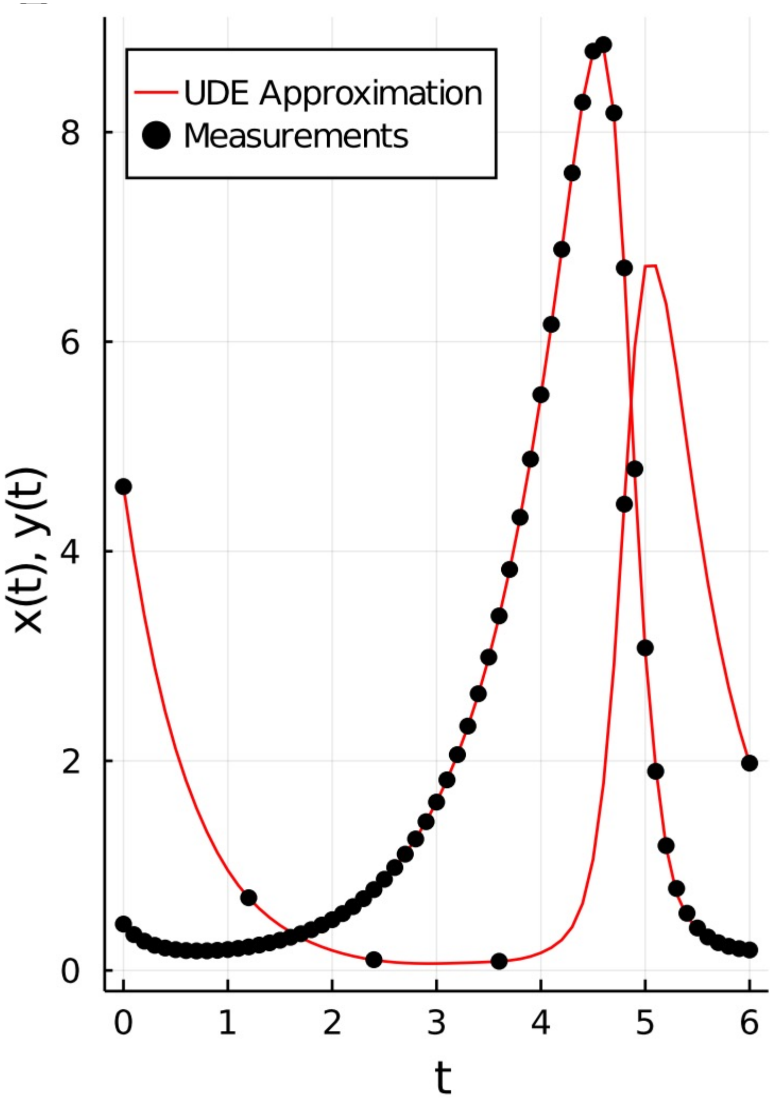
$$\dot{x} = \alpha x \quad + NN_1(\theta, x, y)$$
$$\dot{y} = -\theta_1 y + NN_2(\theta, x, y)$$

Train on data:

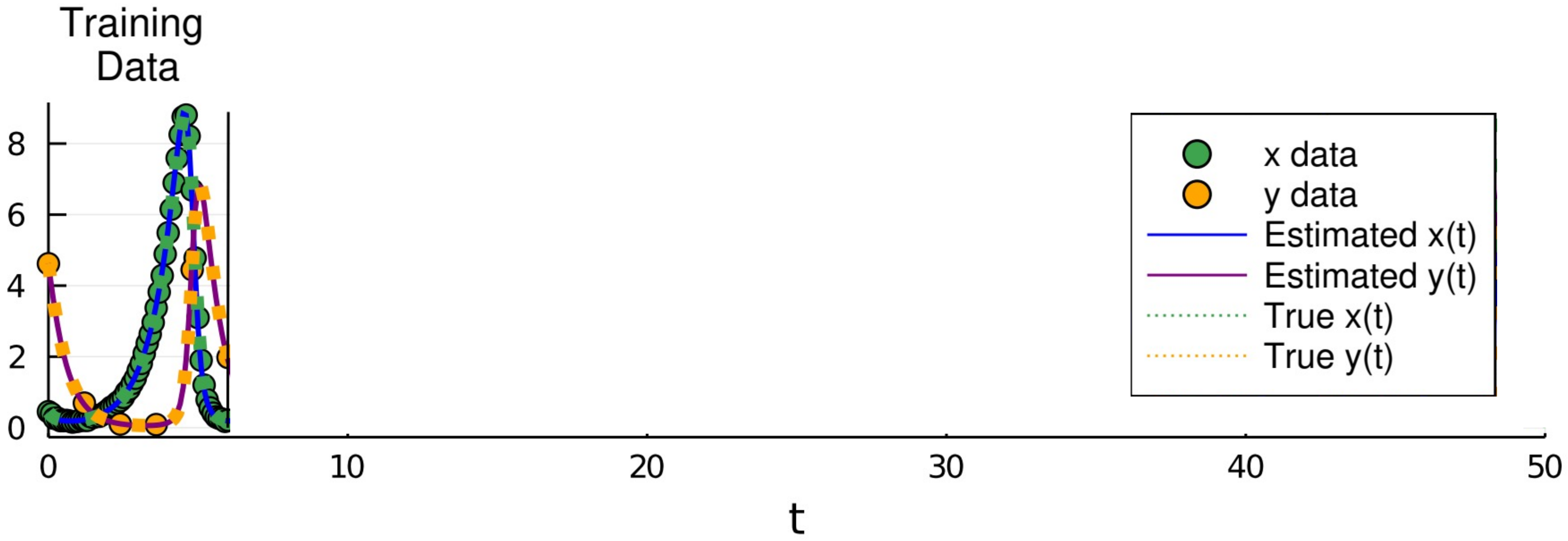
Symbolic regression: $\Rightarrow \begin{bmatrix} -\xi_1 xy \\ \xi_2 xy \end{bmatrix}$

Solution:

$$\dot{x} = \alpha x \quad - \xi_1 xy$$
$$\dot{y} = -\theta_1 y + \xi_2 xy$$



Extrapolation



Discovery of Unknown Physics?

1. Identify known parts of a model, build a UODE
2. Train a neural network to capture the missing mechanisms
3. Sparse identify the missing terms to mechanistic terms
4. Verify the mechanisms are scientifically plausible
5. Extrapolate, do asymptotic analysis, predict bifurcation, etc.
6. Get more data to verify the new terms

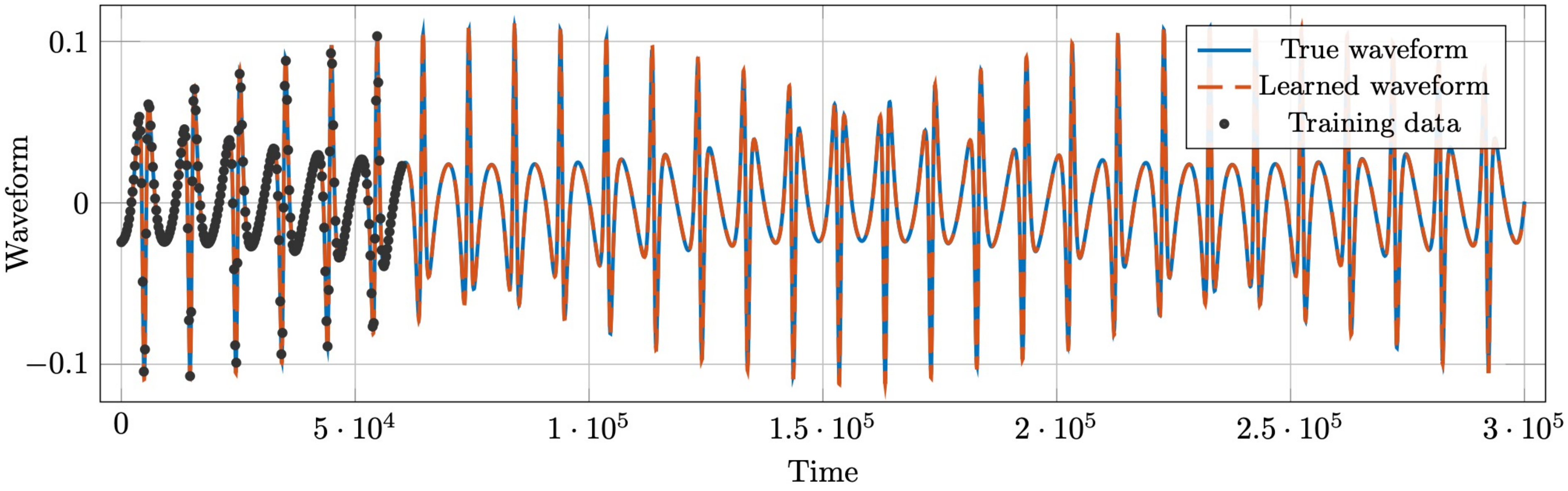
UODEs show accurate extrapolation and generalization

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

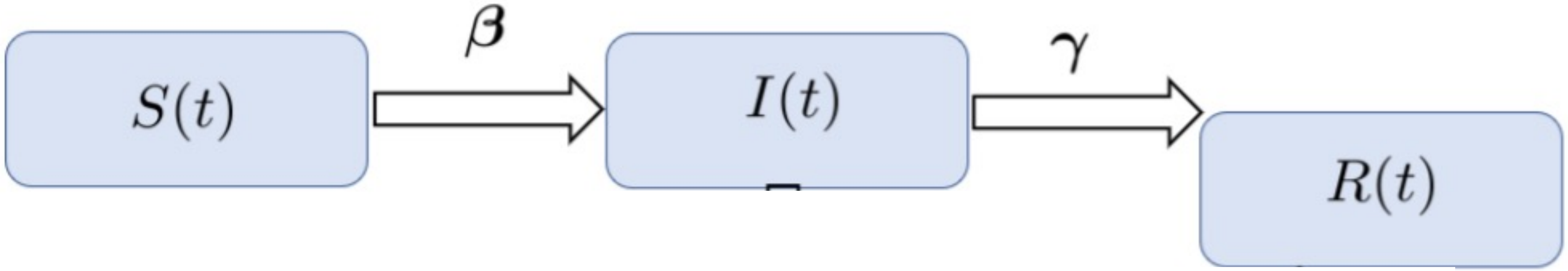
$$\dot{\phi} = \frac{(1 + e\cos(\chi))^2}{Mp^{3/2}}\left(1 + \mathcal{F}_1(\cos(\chi), p, e)\right),$$
$$\dot{\chi} = \frac{(1 + e\cos(\chi))^2}{Mp^{3/2}}\left(1 + \mathcal{F}_2(\cos(\chi), p, e)\right),$$
$$\dot{p} = \mathcal{F}_3(p, e),$$
$$\dot{e} = \mathcal{F}_4(p, e),$$

Demonstrated with the LIGO Black Hole dynamics from the gravitational wave data!

(arXiv:2102.12695)



QSIR Predicts Quarantine Measure Evolution

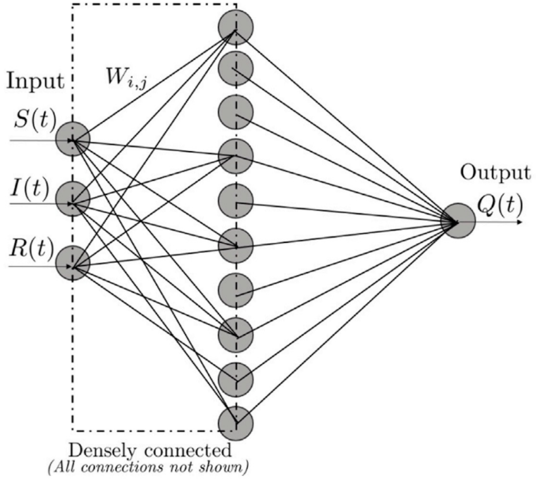


QSIR Predicts Quarantine Measure Evolution

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{\beta\, S(t)\, I(t)}{N}$$

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\beta\, S(t)\, I(t)}{N} - \gamma I(t)$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \gamma I(t).$$



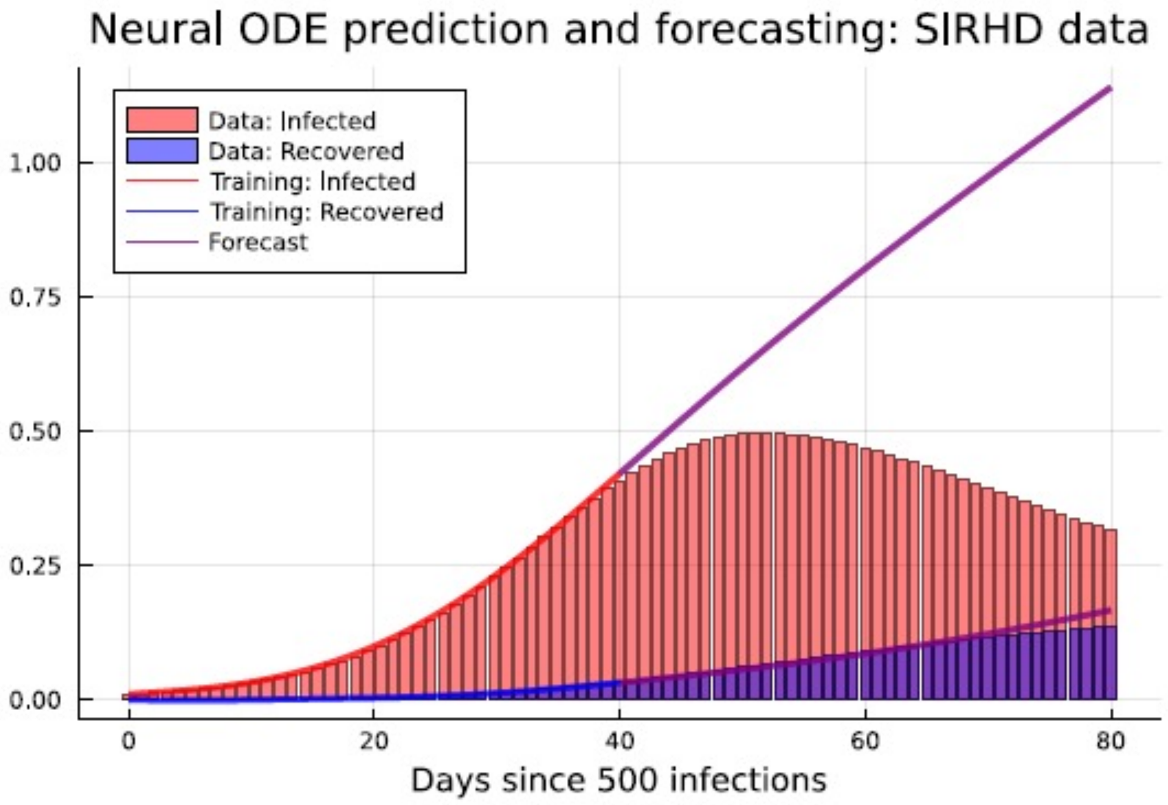
$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{\beta\, S(t)\, I(t)}{N}$$

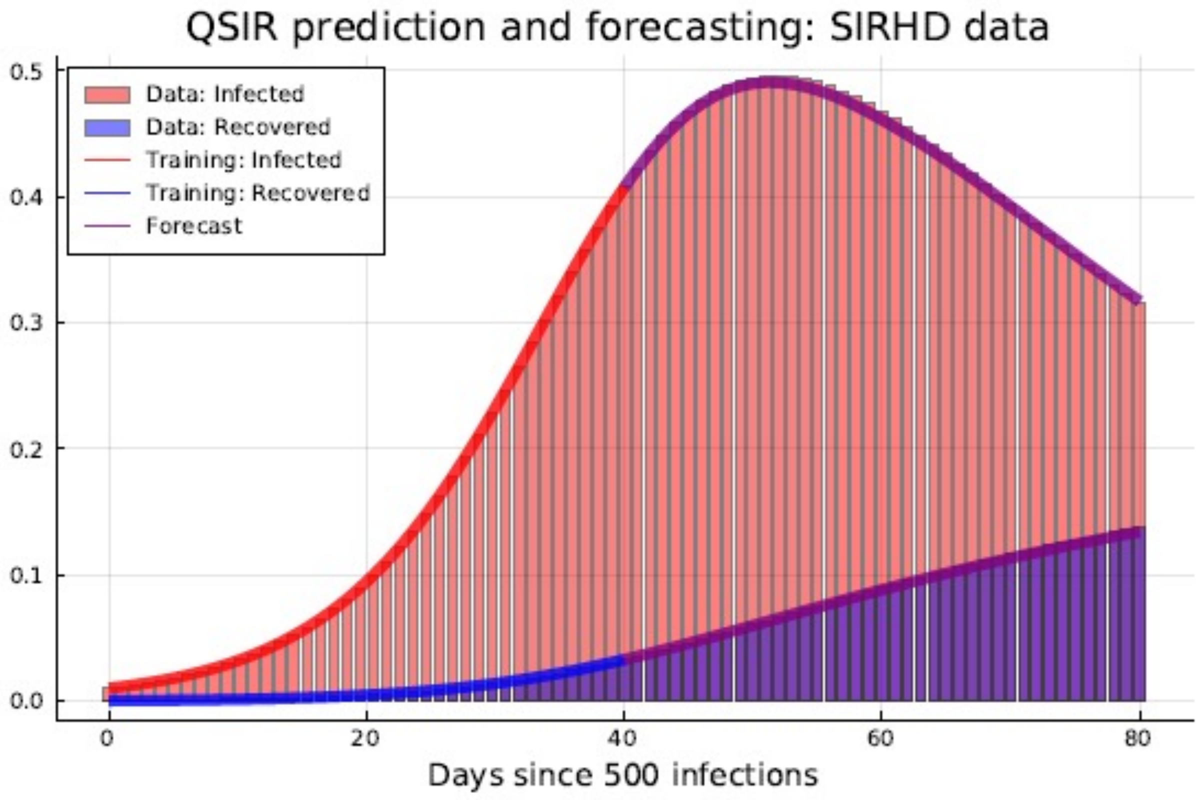
$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\beta\, S(t)\, I(t)}{N} - \left(\gamma + Q(t)\right) I(t) =$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \gamma I(t) + \delta T(t)$$

$$\frac{\mathrm{d}T(t)}{\mathrm{d}t} = Q(t)\, I(t)$$

QSIR Predicts Quarantine Measure Evolution

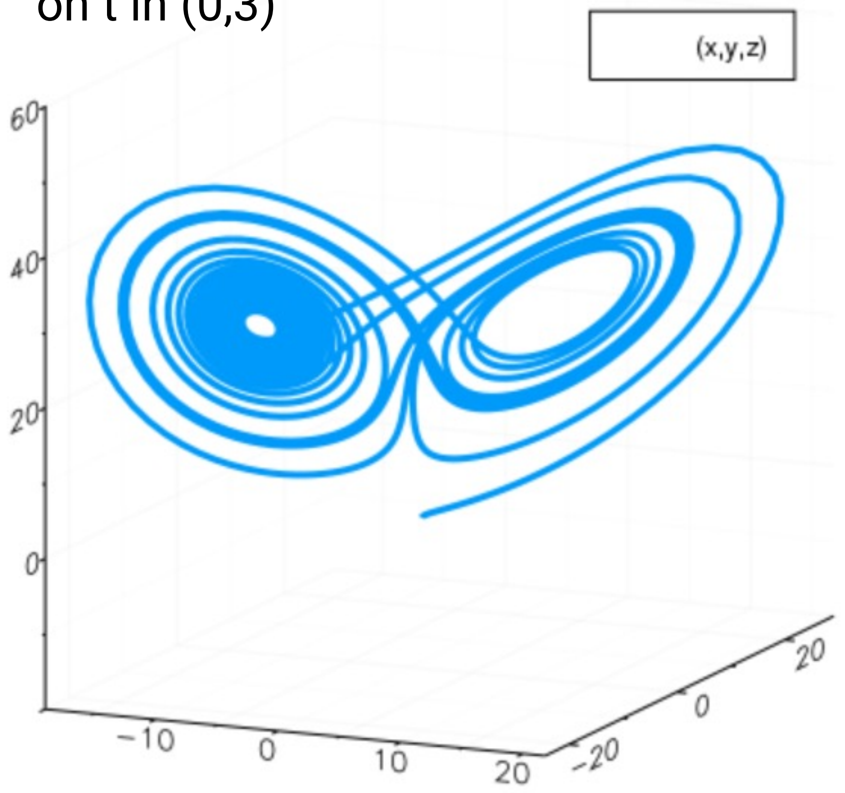




A machine learning aided global diagnostic and comparative tool to assess effect of quarantine control in Covid-19 spread." *Cell Patterns* (2020)

Keeping Neural Networks Small Keeps Speed For Inverse Problems

Problem: parameter estimation of Lorenz equation from data on t in (0,3)



DeepXDE (TensorFlow Physics-Informed NN)

```
Best model at step 57000:
  train loss: 5.91e-03
  test loss: 5.86e-03
  test metric: []

'train' took 362.351454 s
```

DiffEqFlux.jl (Julia UDEs)

```
opt = Opt(:LN_BOBYQA, 3)
lower_bounds!(opt,[9.0,20.0,2.0])
upper_bounds!(opt,[11.0,30.0,3.0])
min_objective!(opt, obj_short.cost_function2)
xtol_rel!(opt,1e-12)
maxeval!(opt, 10000)
@time (minf,minx,ret) = NLopt.optimize(opt,LocIniPar) # 0.1 seconds
```

```
0.032699 seconds (148.87 k allocations: 14.175 MiB)
(2.7636309213683456e-18, [10.0, 28.0, 2.66], :XTOL_REACHED)
```