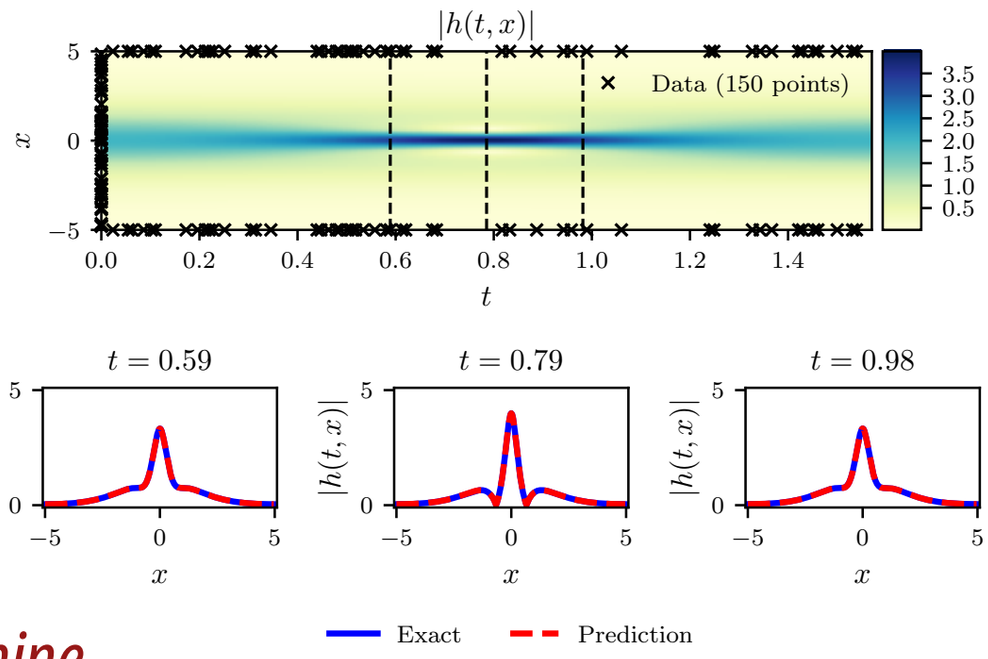
Data-driven discovery of nonlinear dynamical systems

Paris Perdikaris
Department of Mechanical Engineering and Applied Mechanics
University of Pennsylvania
email: pgp@seas.upenn.edu

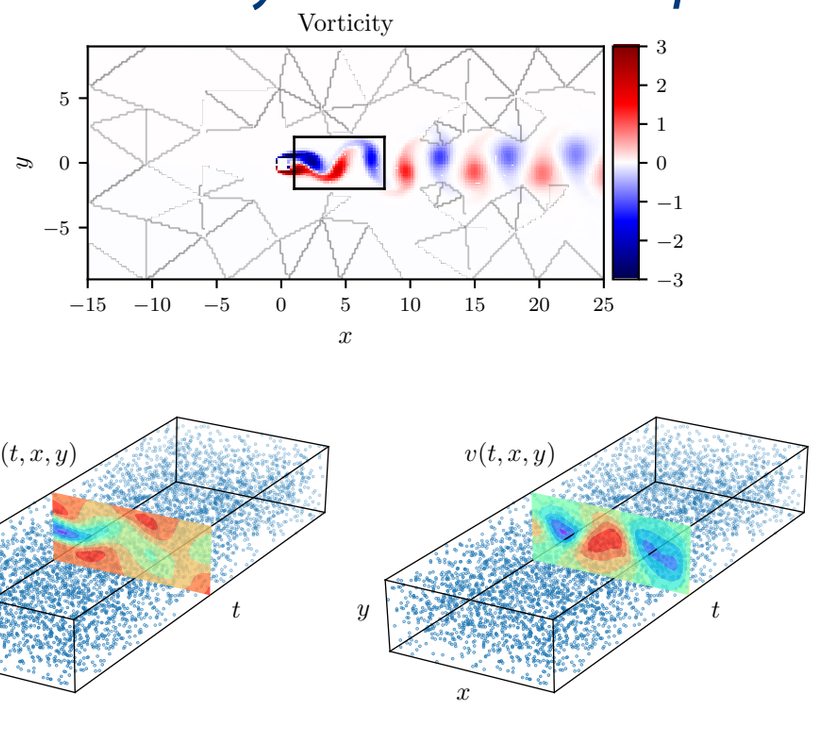
25ο Θερινό Σχολείο-Συνέδριο
"Δυναμικά Συστήματα και Πολυπλοκότητα"
ΕΚΕΦΕ "Δημόκριτος"
Αθήνα, 9-17 Ιουλίου 2018

Overview

Data-driven solution of differential equations



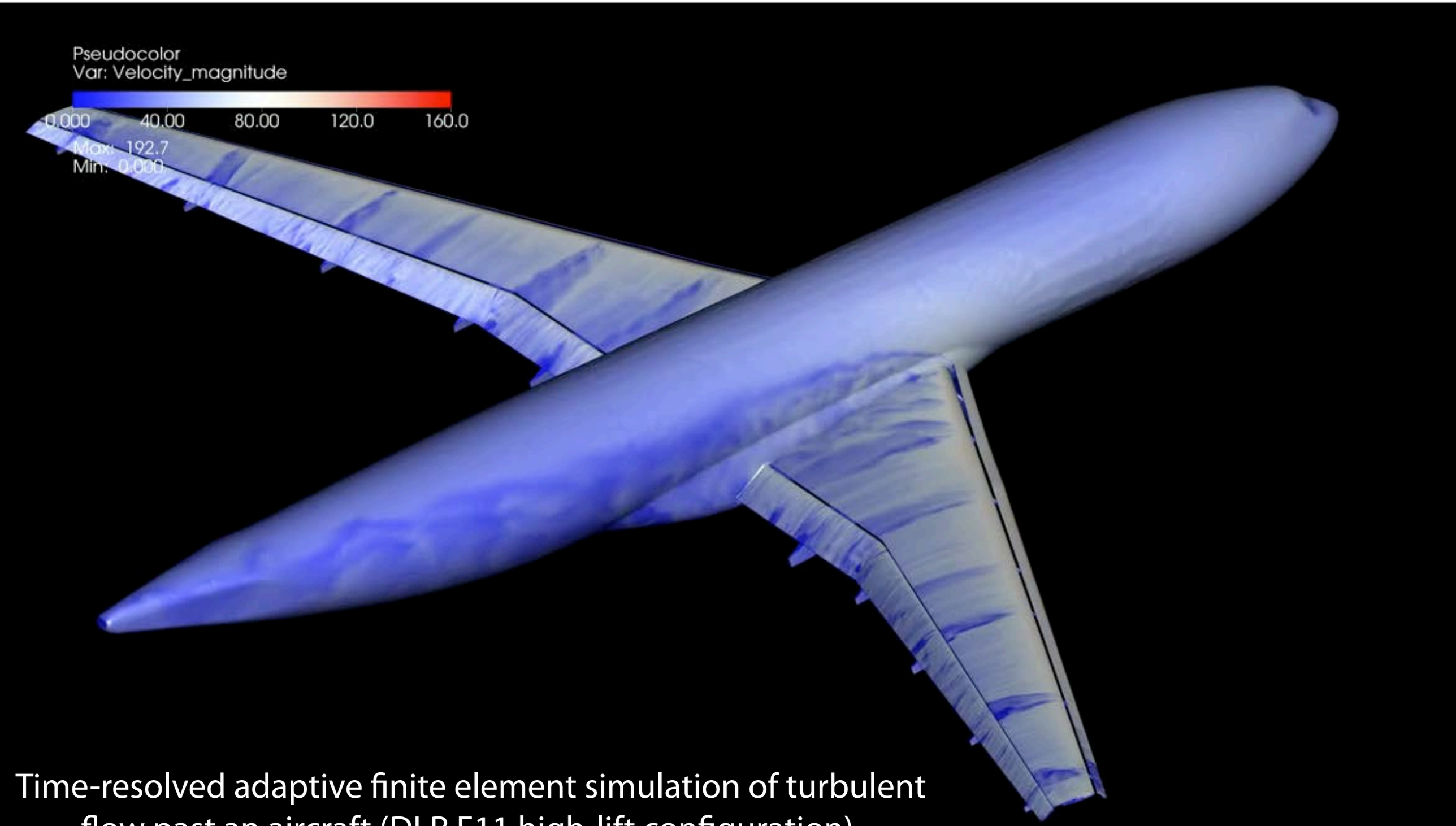
Data-driven discovery of differential equations



Machine learning

. . . some old ideas revisited with modern computational tools:

- *Psichogios, D. C., & Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. AIChE Journal, 38(10), 1499-1511.*
- *Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1997). Artificial neural networks for solving ordinary and partial differential equations. arXiv preprint physics/9705023.*
- *Rico-Martinez, R., Anderson, J. S., & Kevrekidis, I. G. (1994, September). Continuous-time nonlinear signal processing: a neural network based approach for gray box identification. In Neural Networks for Signal Processing [1994] IV. Proceedings of the 1994 IEEE Workshop (pp. 596-605). IEEE.*



Time-resolved adaptive finite element simulation of turbulent flow past an aircraft (DLR F11 high-lift configuration). Simulation is by CTL (http://www.csc.kth.se/ctl)


LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA
MOTION FLOW
LANE LINES
LANE LINES
ROAD FLOW
IN-PATH OBJECTS
ROAD LIGHTS
OBJECTS
ROAD SIGNS

Physics-informed neural networks

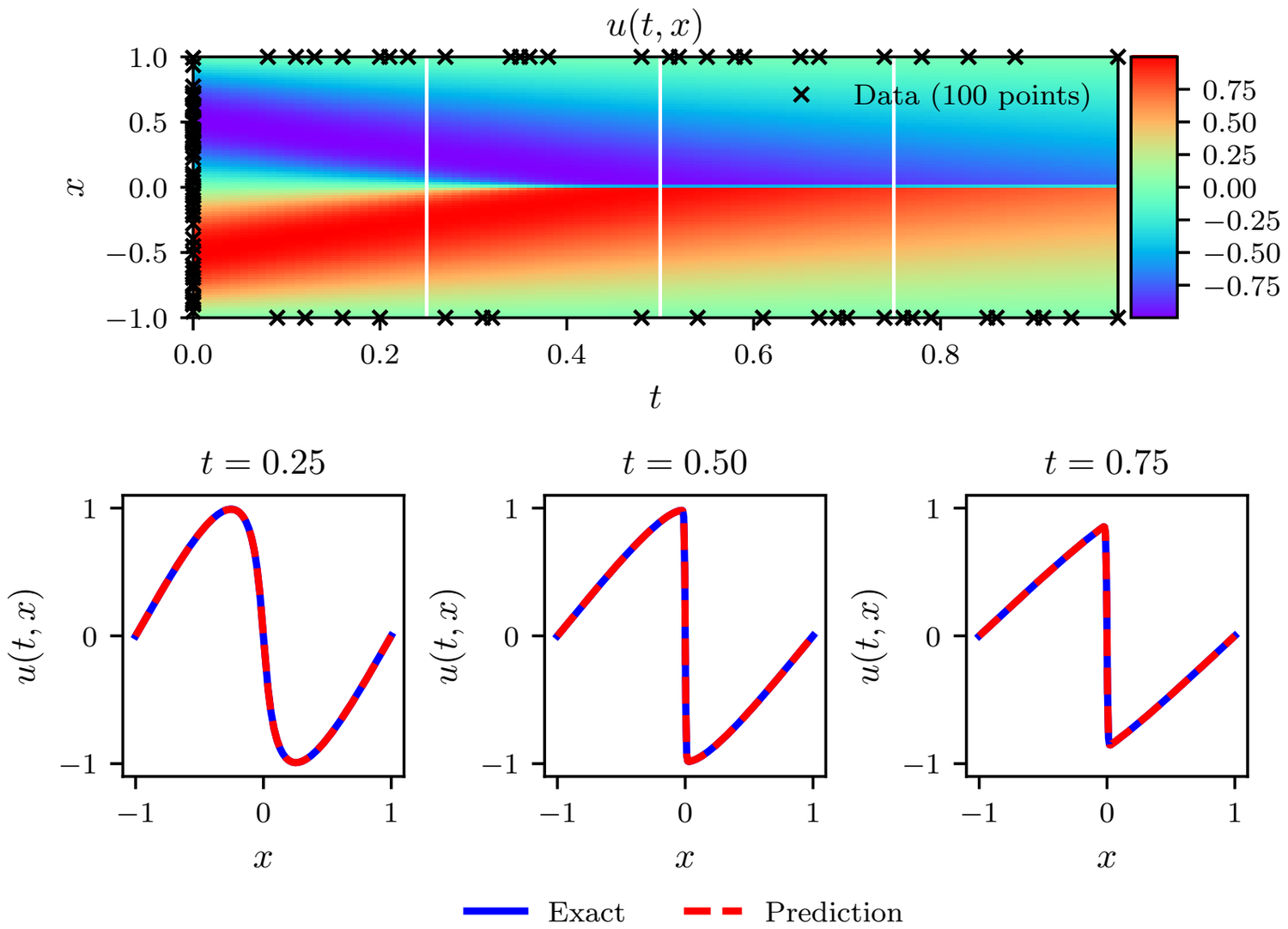


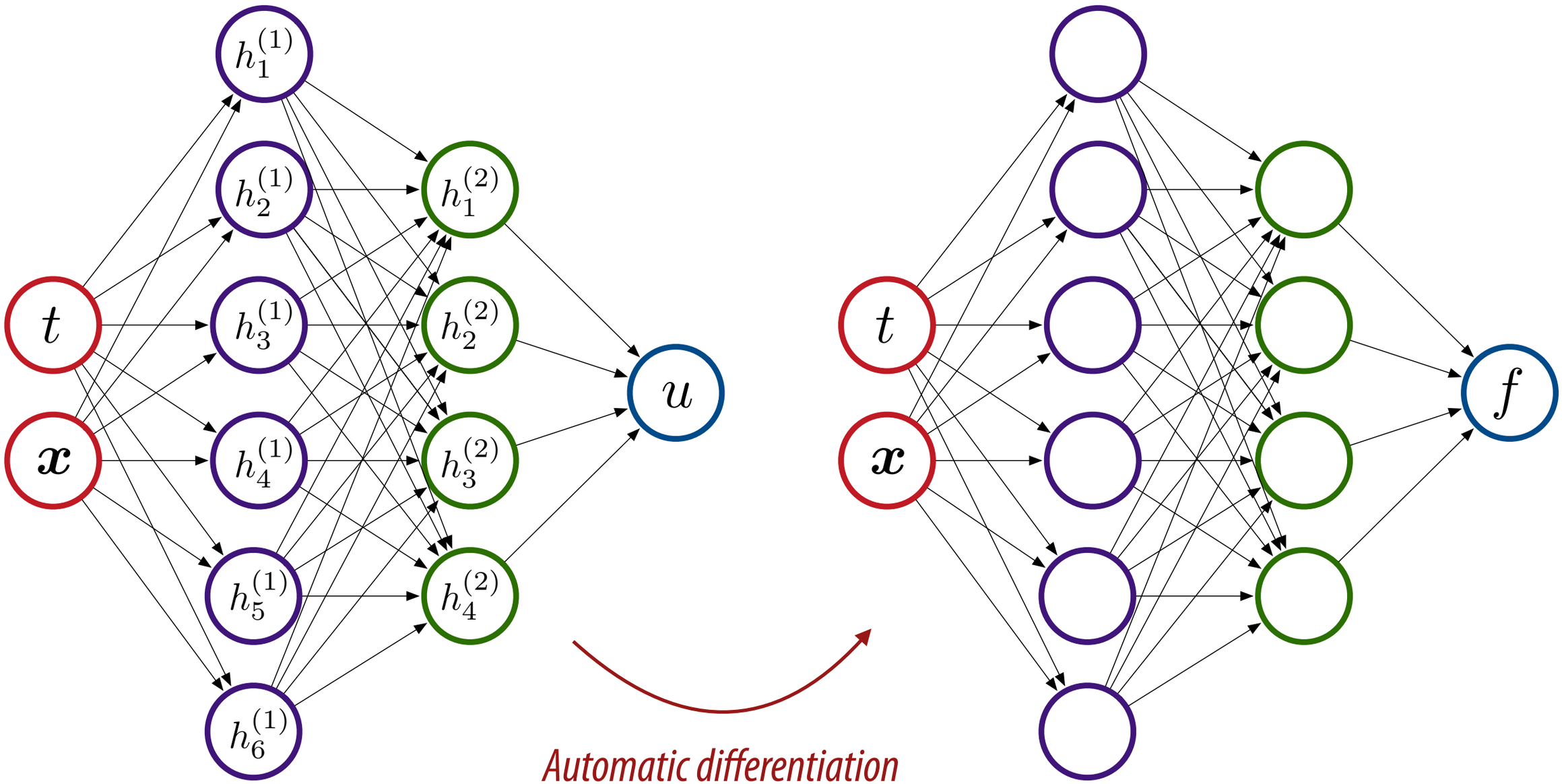
Figure 1: *Burgers' equation: Top:* Predicted solution $u(t, x)$ along with the initial and boundary training data. In addition we are using 10,000 collocation points generated using a Latin Hypercube Sampling strategy. *Bottom:* Comparison of the predicted and exact solutions corresponding to the three temporal snapshots depicted by the white vertical lines in the top panel. The relative $\mathcal{L}_2$ error for this case is $6.7 \cdot 10^{-4}$. Model training took approximately 60 seconds on a single NVIDIA Titan X GPU card.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. arXiv preprint arXiv

Physics-informed neural networks

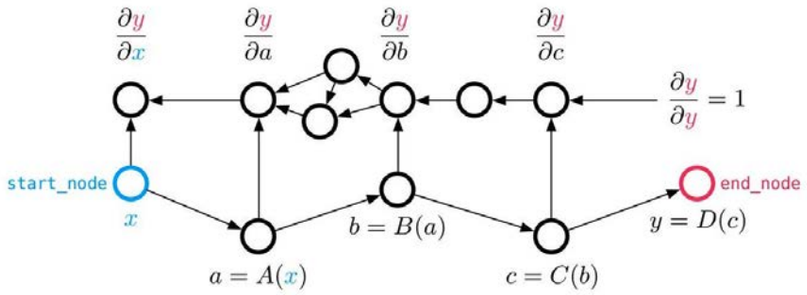
$u(t, \boldsymbol{x})$

$f = u_t + \mathcal{N}[u; \lambda]$



Automatic differentiation

TensorFlow ™



PYTORCH

Data-driven solution of PDEs

$$u_t + \mathcal{N}[u;\lambda] = 0,\ x \in \Omega,\ t \in [0,T]$$

We define $f(t,x)$ to be given by the left-hand-side of equation (1); i.e.,

$$f := u_t + \mathcal{N}[u;\lambda], \tag{2}$$

and proceed by approximating $u(t,x)$ by a deep neural network. This assumption along with equation (2) result in a *physics informed neural network* $f(t,x)$. This network can be derived by applying the chain rule for differentiating compositions of functions using automatic differentiation [11]. It is worth highlighting that the parameters of the differential operator λ turn into parameters of the *physics informed neural network* $f(t,x)$.

$$MSE = MSE_u + MSE_f,$$

$$MSE_u = \frac{1}{N}\sum_{i=1}^{N} |u(t_u^i, x_u^i) - u^i|^2,$$

$$MSE_f = \frac{1}{N}\sum_{i=1}^{N} |f(t_u^i, x_u^i)|^2.$$

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. arXiv preprint arXiv

Physics-informed neural networks

Example: Burgers' equation in 1D

$$
\begin{aligned}
&u_t + uu_x - (0.01/\pi)u_{xx} = 0, \quad x \in [-1, 1], \quad t \in [0, 1], \qquad (3)\\
&u(0, x) = -\sin(\pi x),\\
&u(t, -1) = u(t, 1) = 0.
\end{aligned}
$$

Let us define $f(t, x)$ to be given by

$$f := u_t + uu_x - (0.01/\pi)u_{xx},$$

```
def u(t, x):
    u = neural_net(tf.concat([t,x],1), weights, biases)
    return u
```

Correspondingly, the *physics informed neural network* $f(t, x)$ takes the form

```
def f(t, x):
    u = u(t, x)
    u_t = tf.gradients(u, t)[0]
    u_x = tf.gradients(u, x)[0]
    u_xx = tf.gradients(u_x, x)[0]
    f = u_t + u*u_x - (0.01/tf.pi)*u_xx
    return f
```

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. arXiv preprint arXiv

Physics-informed neural networks

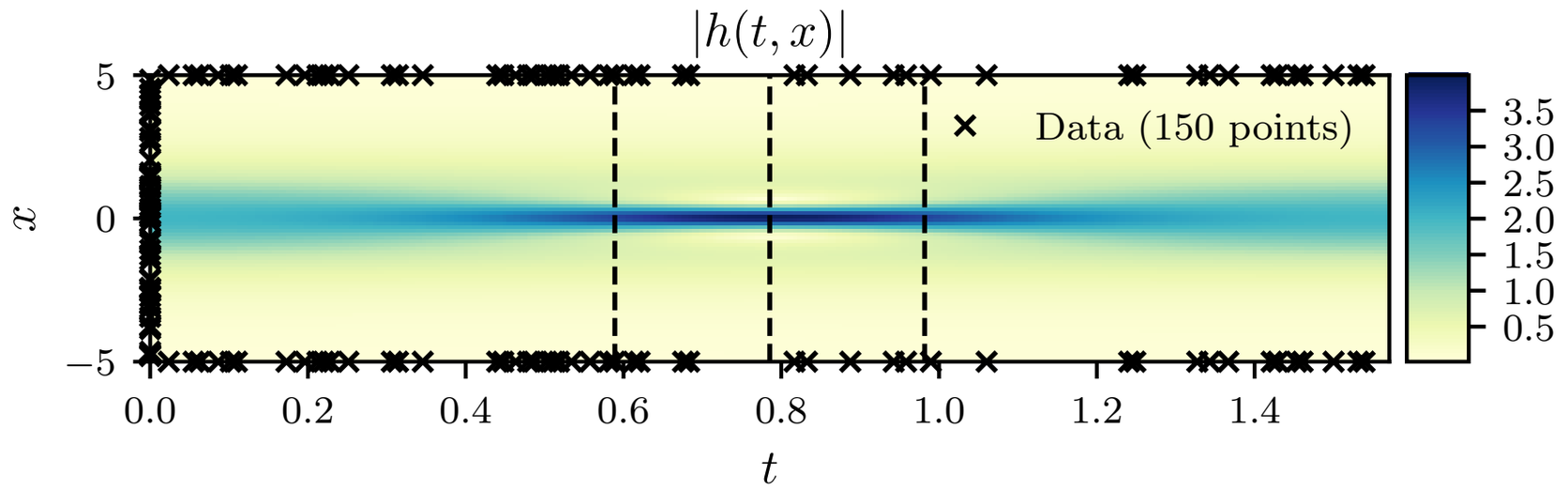


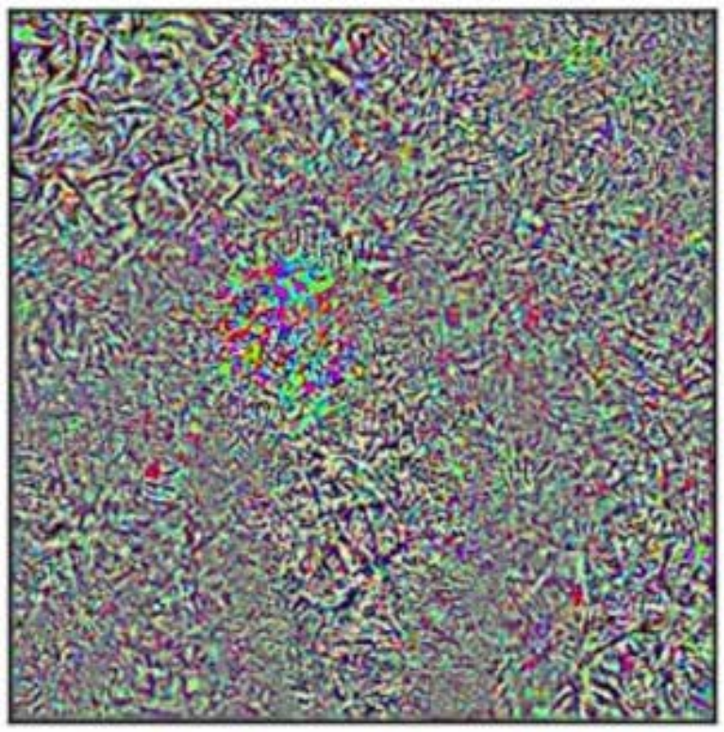
Figure 1: *Shrödinger equation: Top:* Predicted solution $|h(t,x)|$ along with the initial and boundary training data. In addition we are using 20,000 collocation points generated using a Latin Hypercube Sampling strategy. *Bottom:* Comparison of the predicted and exact solutions corresponding to the three temporal snapshots depicted by the dashed vertical lines in the top panel. The relative $\mathbb{L}_2$ error for this case is $1.97 \cdot 10^{-3}$.

Accounting for uncertainty

X

97.3% macaw

+



$\text{sign}(\nabla_X J(\theta, X, Y))$

=

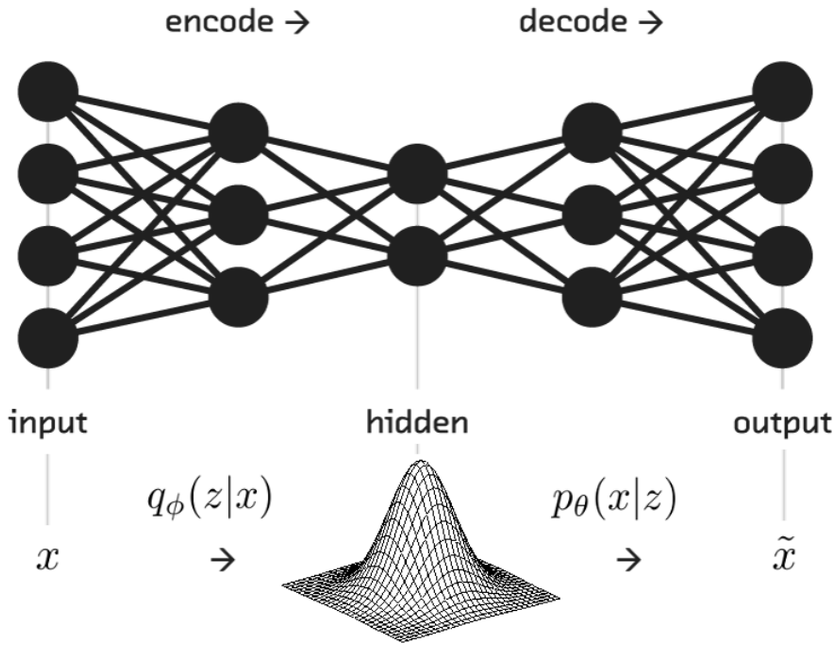
$X + \epsilon.\text{sign}(\nabla_X J(\theta, X, Y))$

88.9% bookcase

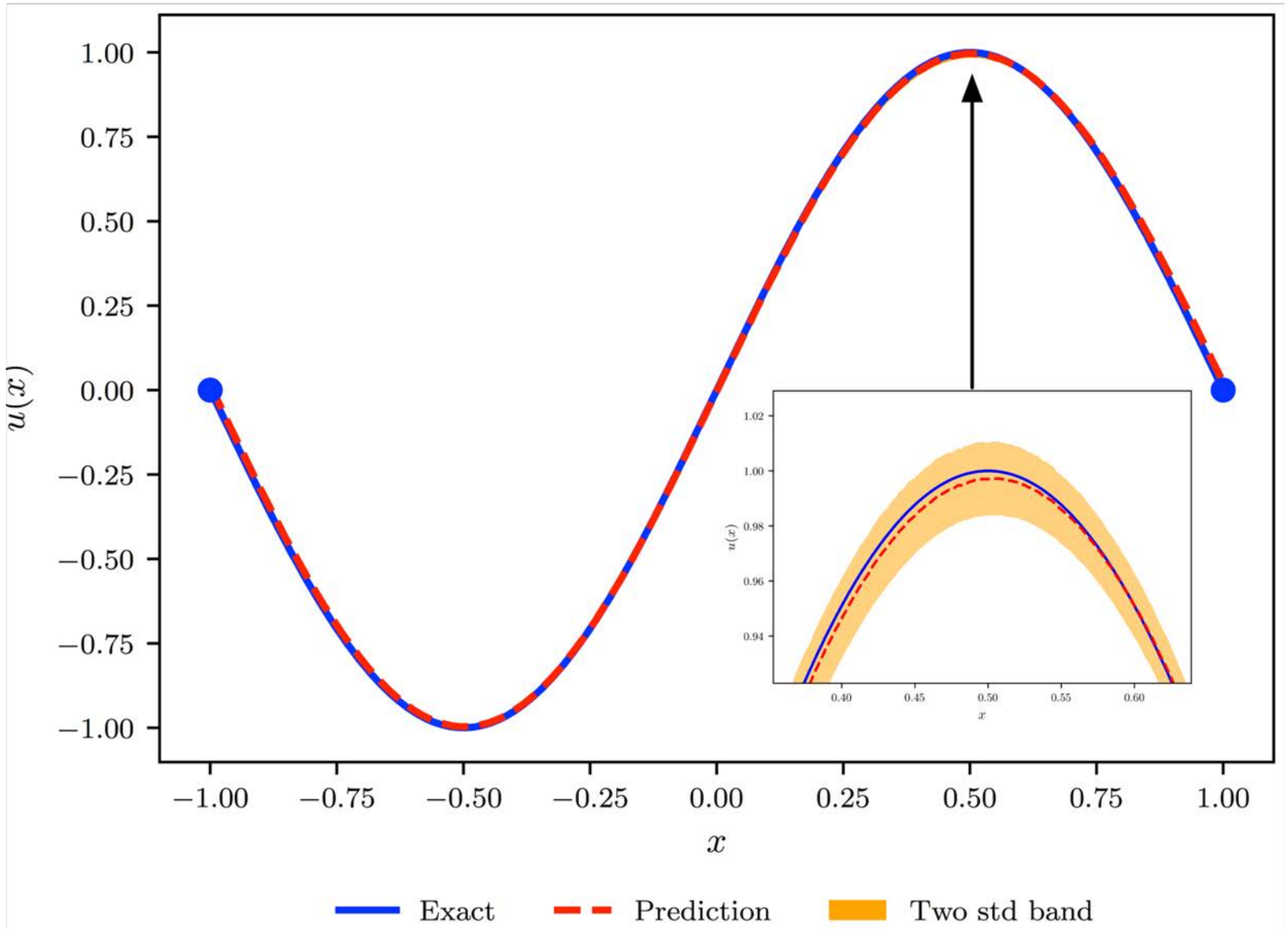
Accounting for uncertainty

Physics informed machine learning with deep generative models

$$\underbrace{p(\boldsymbol{z}|\boldsymbol{u},\boldsymbol{x})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{u}|\boldsymbol{x},\boldsymbol{z})}^{\text{likelihood}}\overbrace{p(\boldsymbol{z}|\boldsymbol{x})}^{\text{prior}}}{\underbrace{p(\boldsymbol{u}|\boldsymbol{x})}_{\text{marginal likelihood}}}, \qquad \text{such that} \qquad u_t + \mathcal{N}_x[u] = 0$$



Example in 1D: $u_{xx} - u^2 u_x = f(x)$



Data-driven discovery of PDEs

$$u_t + \mathcal{N}[u;\lambda] = 0,\ x \in \Omega,\ t \in [0,T]$$

$$\begin{aligned}\mathcal{N}(t,x,u,u_x,u_{xx},\ldots) \;=\; & \alpha_{0,0} + \alpha_{1,0}u + \alpha_{2,0}u^2 + \alpha_{3,0}u^3 + \\ & \alpha_{0,1}u_x + \alpha_{1,1}uu_x + \alpha_{2,1}u^2u_x + \alpha_{3,1}u^3u_x + \\ & \alpha_{0,2}u_{xx} + \alpha_{1,2}uu_{xx} + \alpha_{2,2}u^2u_{xx} + \alpha_{3,2}u^3u_{xx} + \\ & \alpha_{0,3}u_{xxx} + \alpha_{1,3}uu_{xxx} + \alpha_{2,3}u^2u_{xxx} + \alpha_{3,3}u^3u_{xxx}\end{aligned}$$

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. arXiv preprint

Data-driven discovery of PDEs

$$u_t + \mathcal{N}[u;\lambda] = 0,\ x \in \Omega,\ t \in [0,T]$$

We define $f(t,x)$ to be given by the left-hand-side of equation (1); i.e.,

$$f := u_t + \mathcal{N}[u;\lambda], \tag{2}$$

and proceed by approximating $u(t,x)$ by a deep neural network. This assumption along with equation (2) result in a *physics informed neural network* $f(t,x)$. This network can be derived by applying the chain rule for differentiating compositions of functions using automatic differentiation [11]. It is worth highlighting that the parameters of the differential operator λ turn into parameters of the *physics informed neural network* $f(t,x)$.

$$MSE = MSE_u + MSE_f,$$

$$MSE_u = \frac{1}{N}\sum_{i=1}^{N} |u(t_u^i, x_u^i) - u^i|^2,$$

$$MSE_f = \frac{1}{N}\sum_{i=1}^{N} |f(t_u^i, x_u^i)|^2.$$

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. arXiv preprint

Physics-informed neural networks

Example: Burgers' equation in 1D

$$u_t + \lambda_1 u u_x - \lambda_2 u_{xx} = 0. \tag{3}$$

Let us define $f(t, x)$ to be given by

$$f := u_t + \lambda_1 u u_x - \lambda_2 u_{xx}, \tag{4}$$

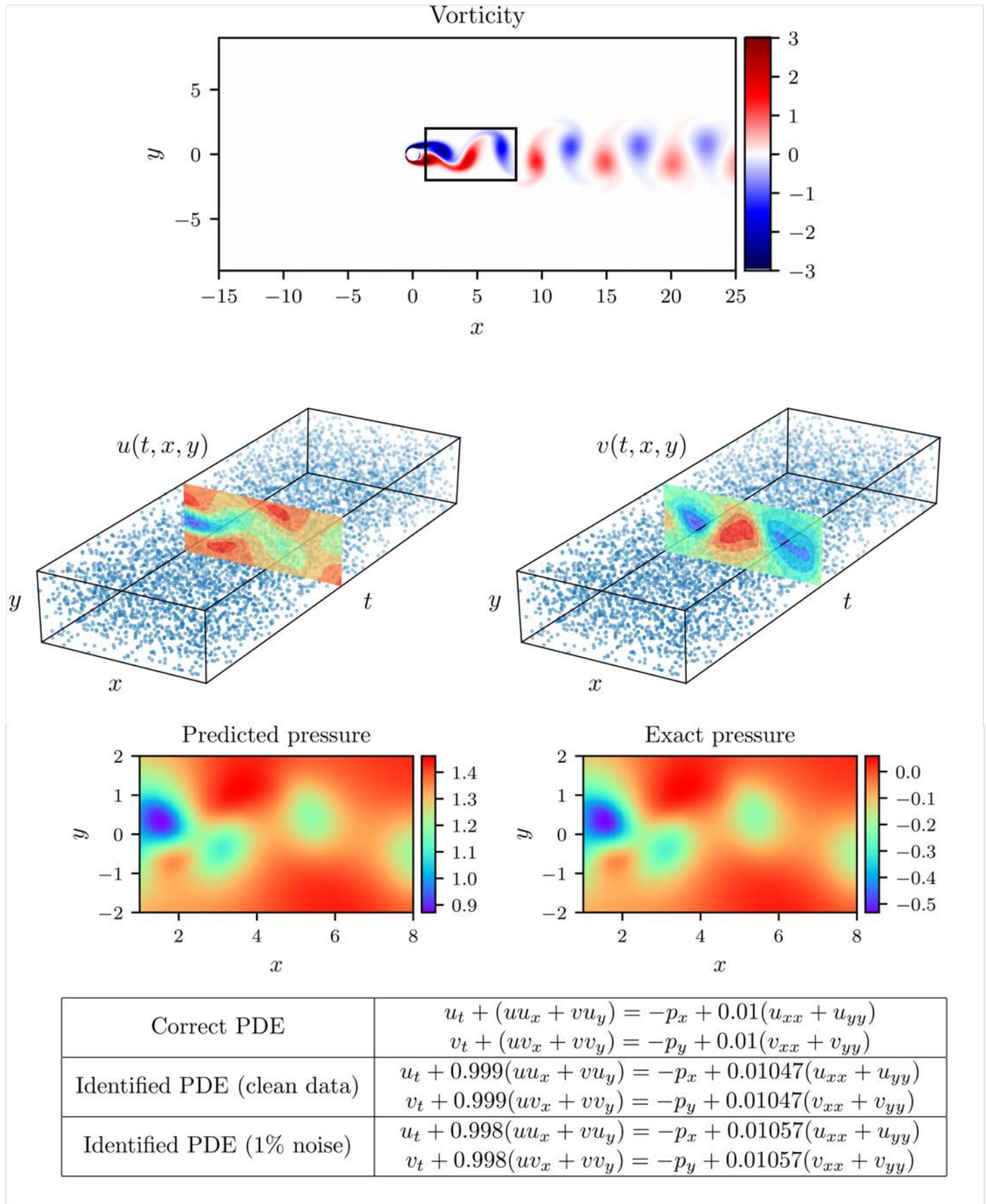
```
def net_u(self, x, t):
    u = self.neural_net(tf.concat([x,t],1), self.weights, self.biases)
    return u

def net_f(self, x,t):
    lambda_1 = self.lambda_1
    lambda_2 = tf.exp(self.lambda_2)
    u = self.net_u(x,t)
    u_t = tf.gradients(u, t)[0]
    u_x = tf.gradients(u, x)[0]
    u_xx = tf.gradients(u_x, x)[0]
    f = u_t + lambda_1*u*u_x - lambda_2*u_xx

    return f
```

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. arXiv preprint

Physics-informed neural networks



Correct PDE	$u_t + (uu_x + vu_y) = -p_x + 0.01(u_{xx} + u_{yy})$ $v_t + (uv_x + vv_y) = -p_y + 0.01(v_{xx} + v_{yy})$
Identified PDE (clean data)	$u_t + 0.999(uu_x + vu_y) = -p_x + 0.01047(u_{xx} + u_{yy})$ $v_t + 0.999(uv_x + vv_y) = -p_y + 0.01047(v_{xx} + v_{yy})$
Identified PDE (1% noise)	$u_t + 0.998(uu_x + vu_y) = -p_x + 0.01057(u_{xx} + u_{yy})$ $v_t + 0.998(uv_x + vv_y) = -p_y + 0.01057(v_{xx} + v_{yy})$

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. arXiv preprint

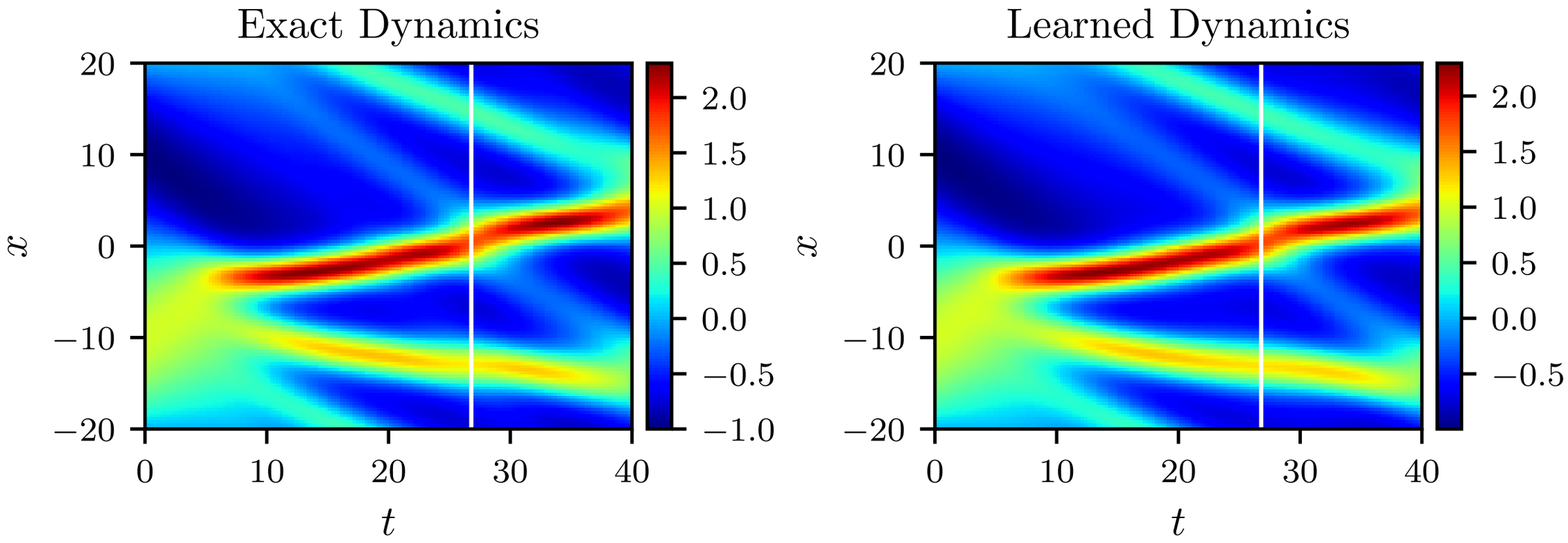
Systems identification (PDEs)

$$u_t = \mathcal{N}(t, x, u, u_x, u_{xx}, \ldots)$$

Example:

The KdV equation:

$$u_t = -uu_x - u_{xxx}$$



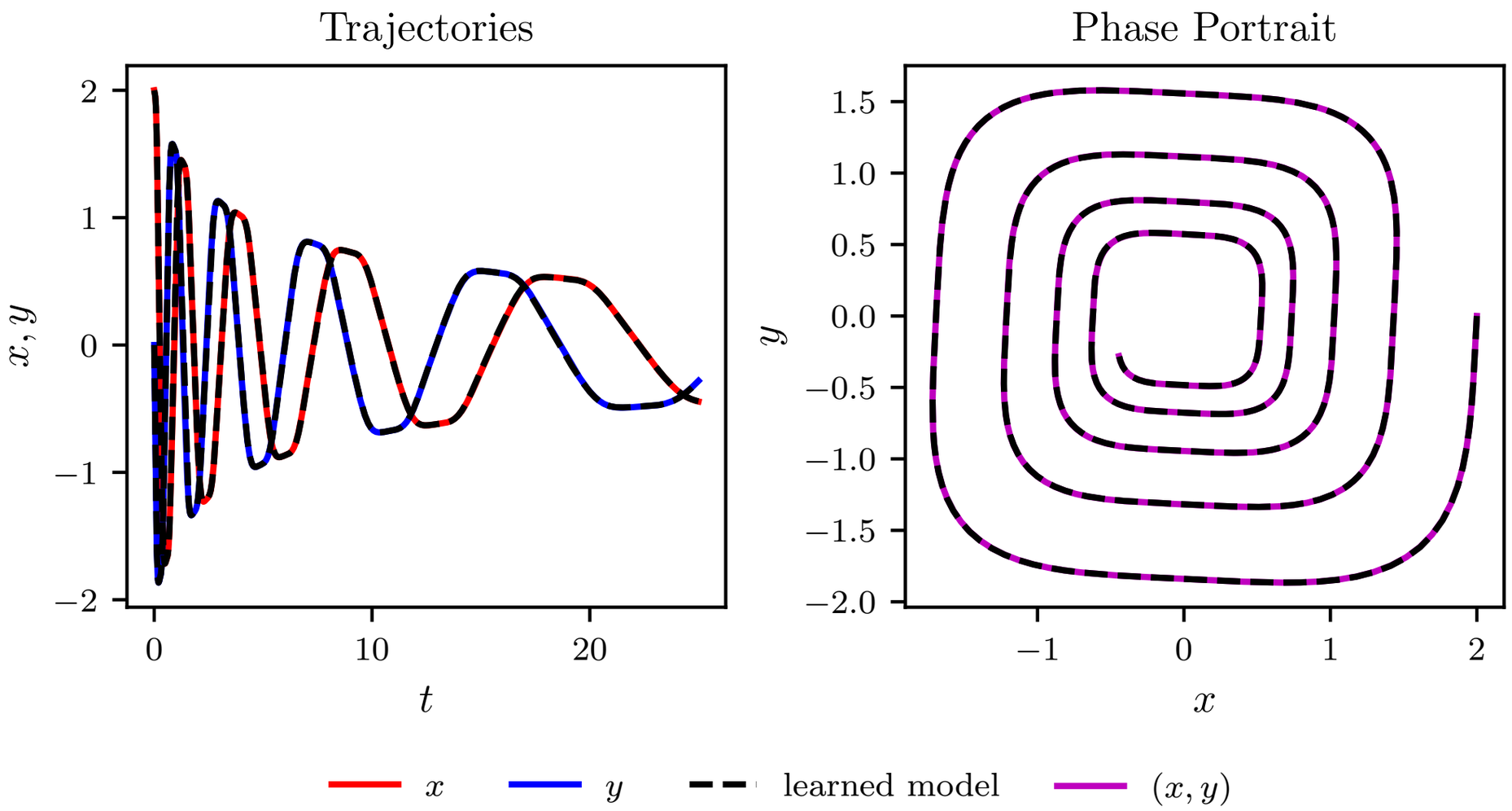
Raissi, M. (2018). Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations. arXiv preprint arXiv:1801.06637.

Systems identification (ODEs)

$$\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t)\right) \longrightarrow \boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \frac{1}{2}\Delta t\left(\boldsymbol{f}(\boldsymbol{x}_n) + \boldsymbol{f}(\boldsymbol{x}_{n-1})\right), \quad n = 1, \ldots, N$$

Harmonic Oscillator:

$$\dot{x} = -0.1\ x^3 + 2.0\ y^3$$
$$\dot{y} = -2.0\ x^3 - 0.1\ y^3$$

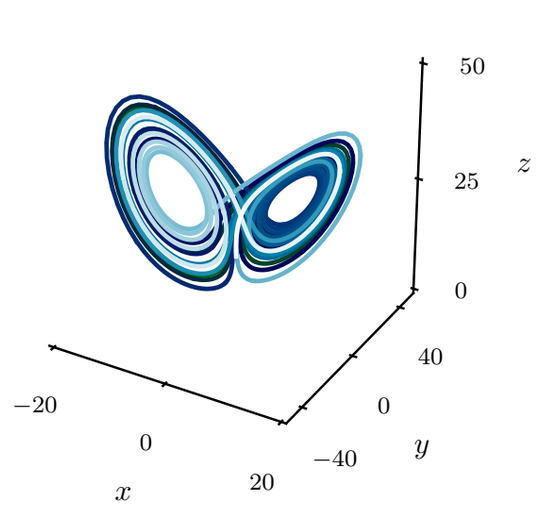


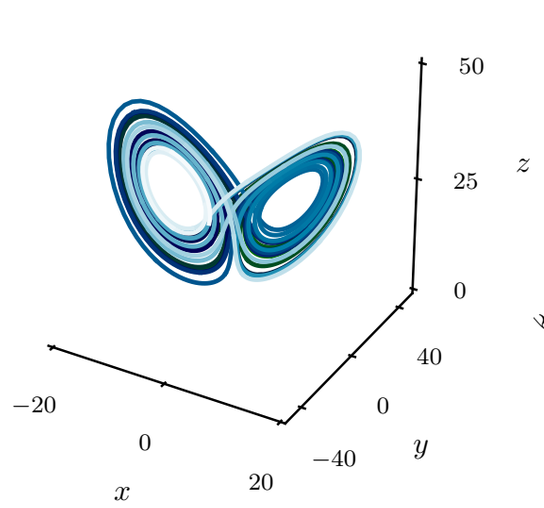
Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2018). Multistep Neural Networks for Data-driven Discovery of Nonlinear Dynamical Systems. arXiv preprint

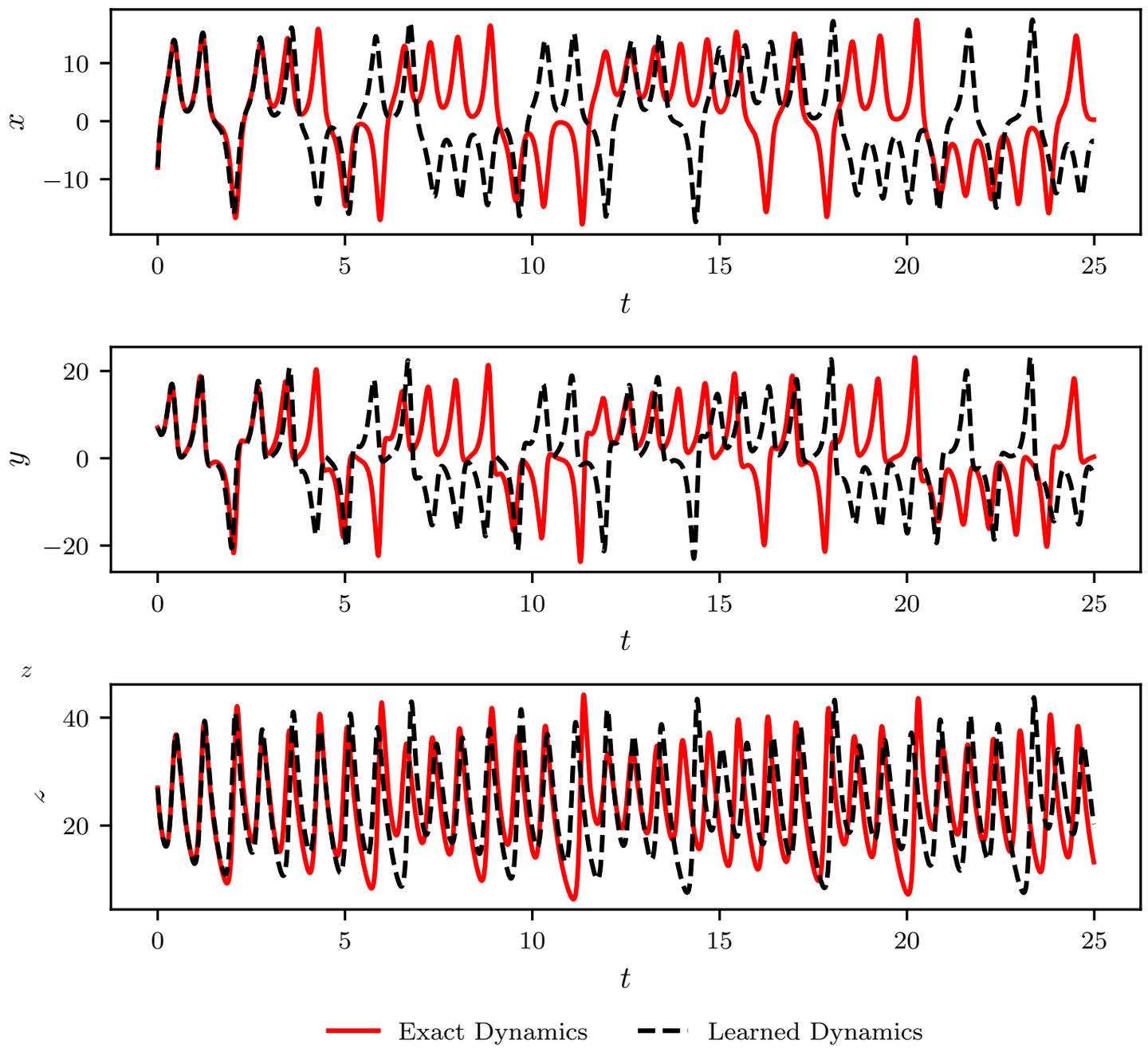
Systems identification (ODEs)

Lorenz System:

$$\dot{x} = 10(y - x),$$
$$\dot{y} = x(28 - z) - y,$$
$$\dot{z} = xy - (8/3)z.$$



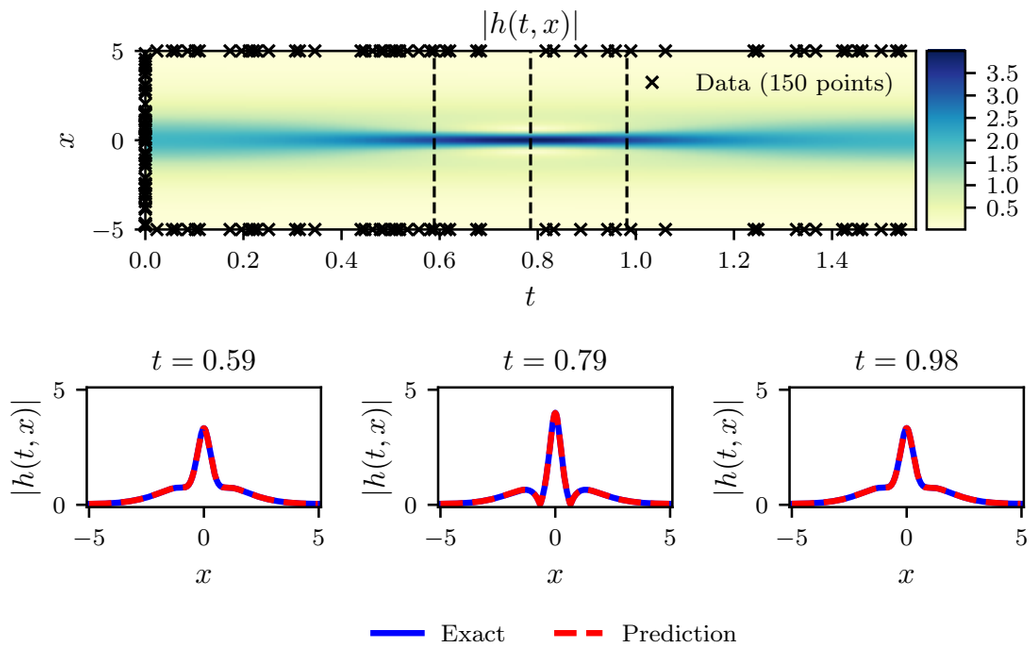




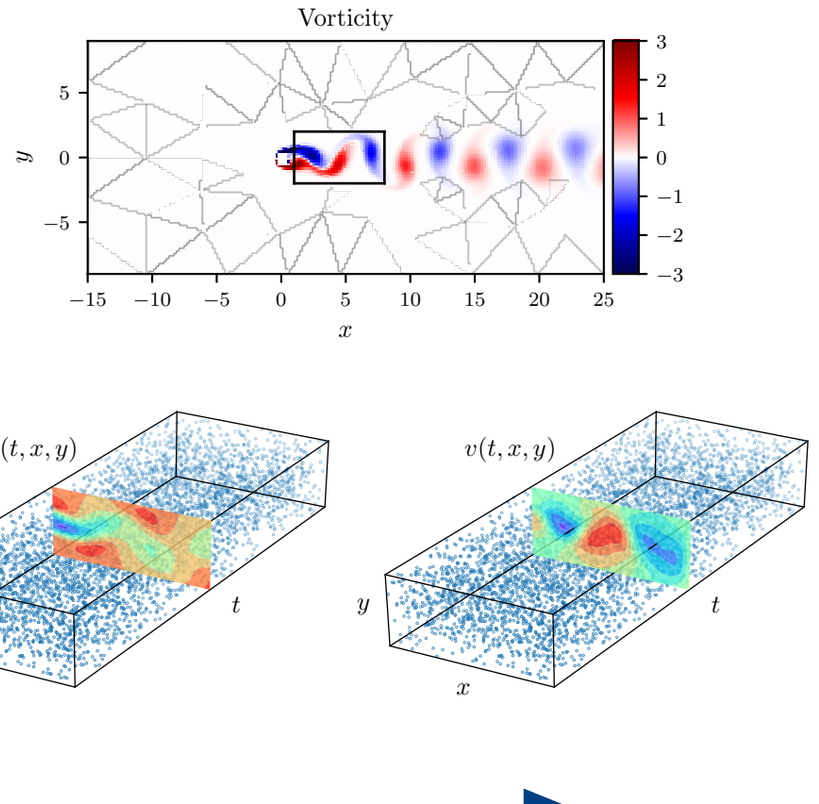
Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2018). Multistep Neural Networks for Data-driven Discovery of Nonlinear Dynamical Systems. arXiv preprint

Summary

Data-driven solution of differential equations



Data-driven discovery of differential equations



Data ⟷ Models

Predictive Intelligence Lab
at the University of Pennsylvania

Penn
University of Pennsylvania

Acknowledgements:

- Maziar Raissi (Brown University)
- George Karniadakis (Brown University)

Email: pgp@seas.upenn.edu

Code: https://github.com/PredictiveIntelligenceLab