Hidden Physics Models

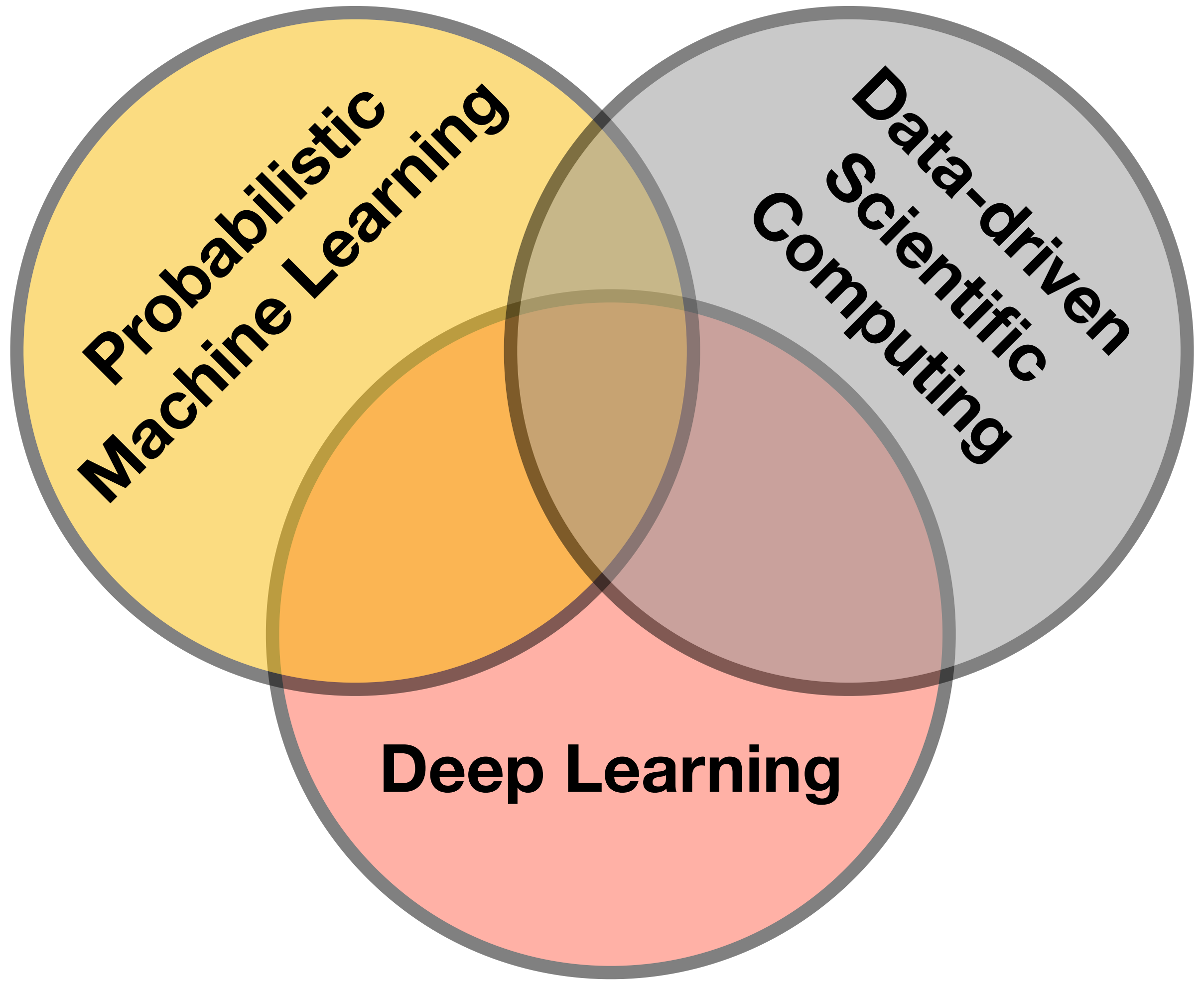
Maziar Raissi

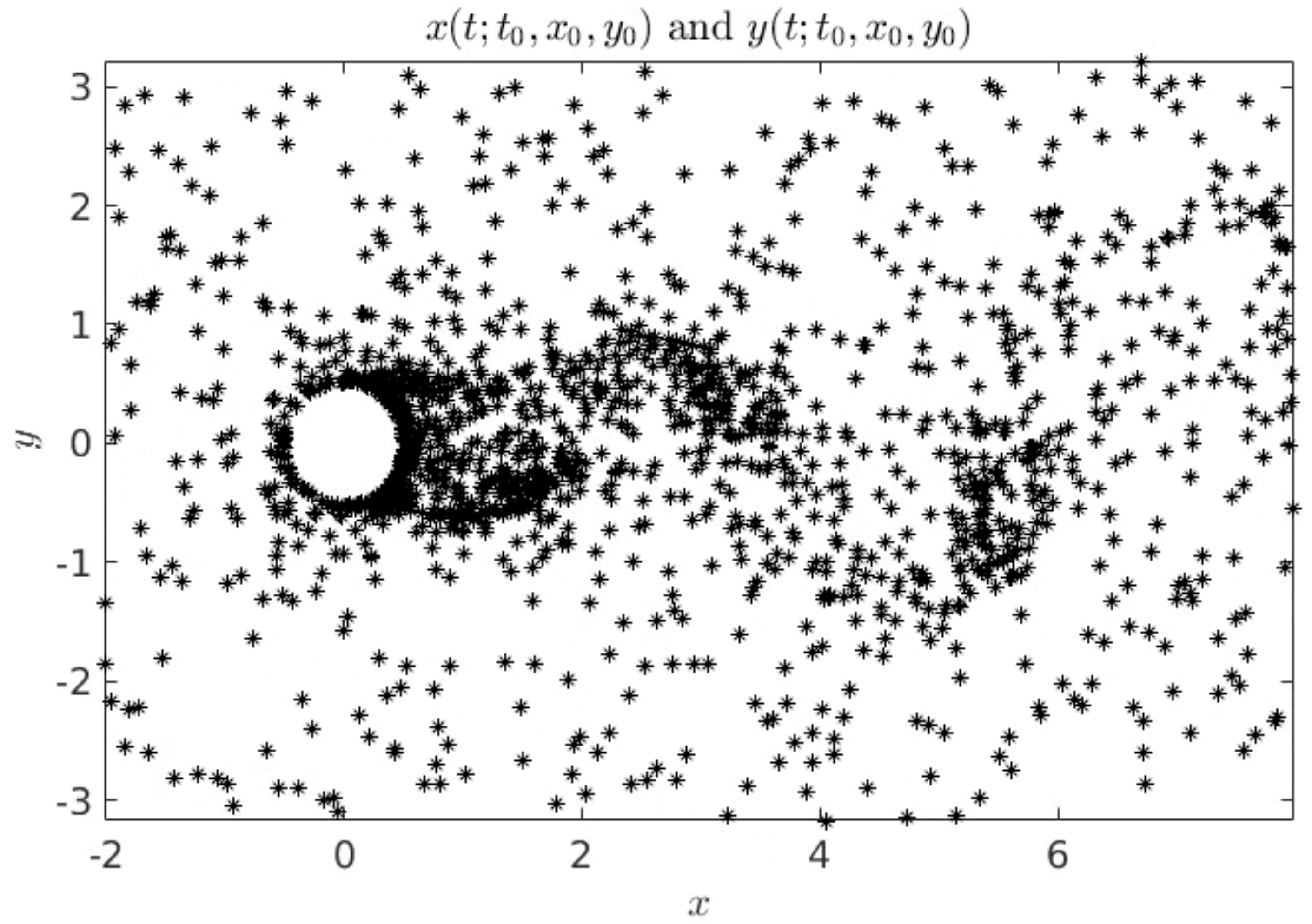
Assistant Professor

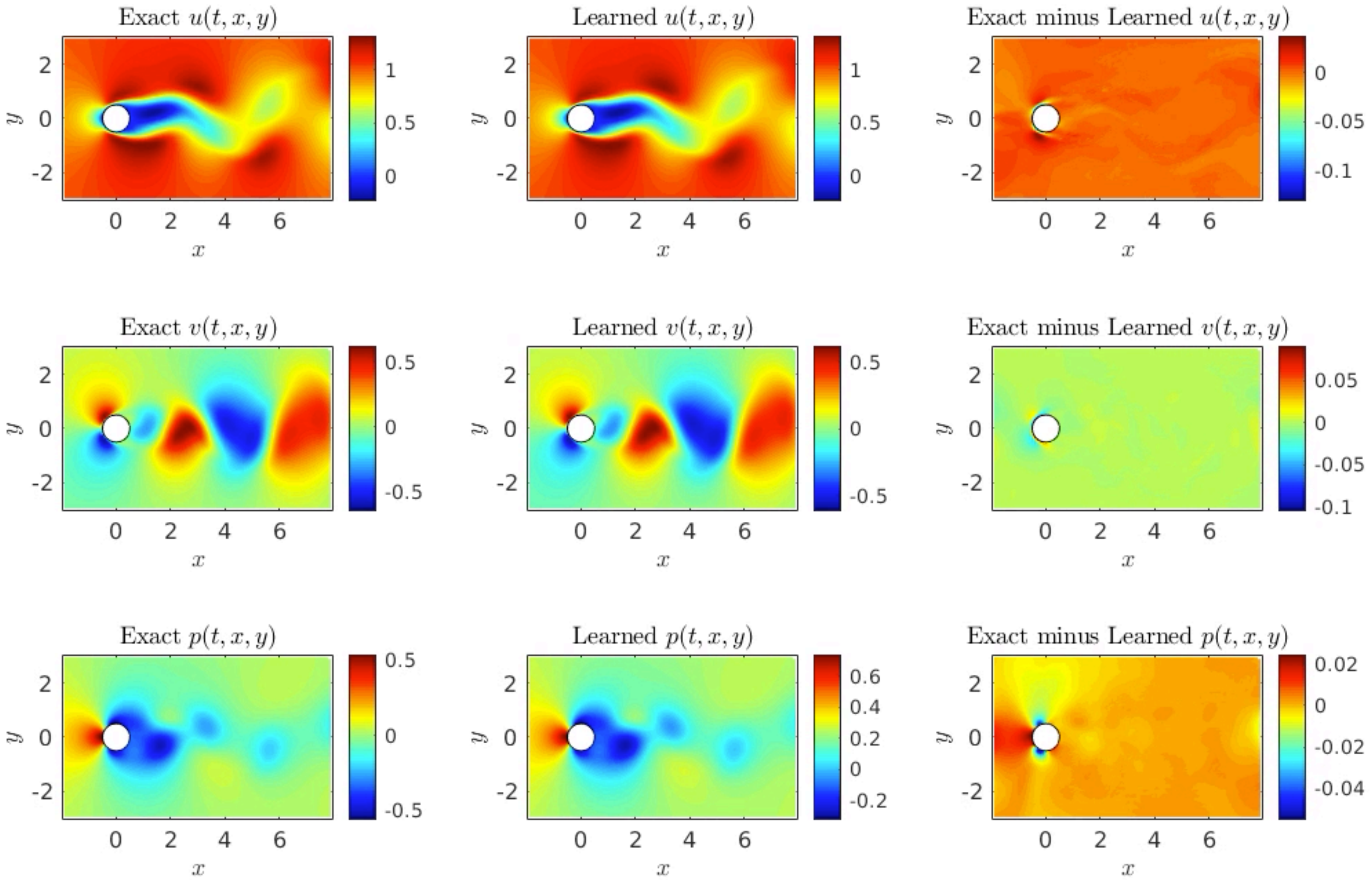
Department of Applied Mathematics

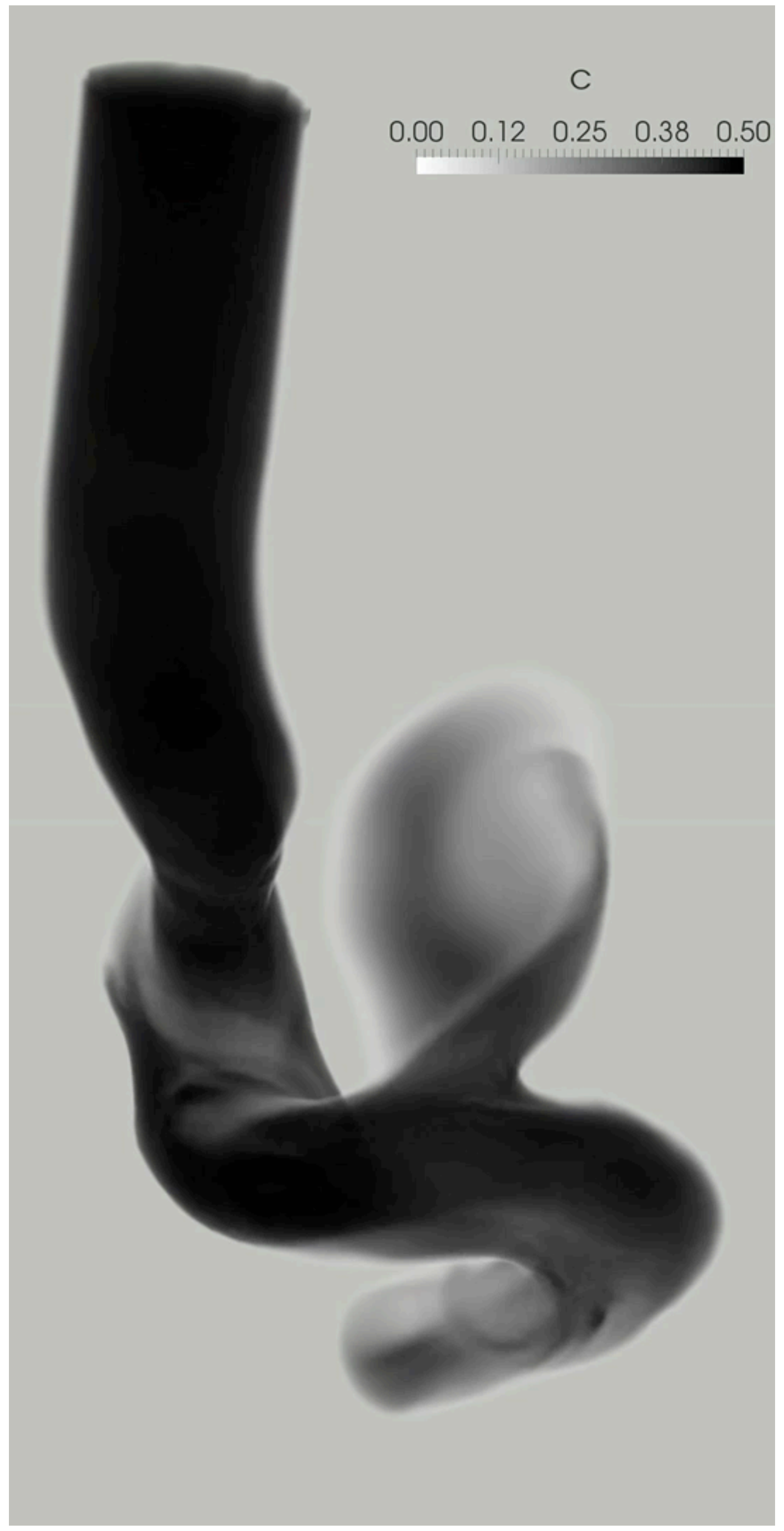
University of Colorado Boulder

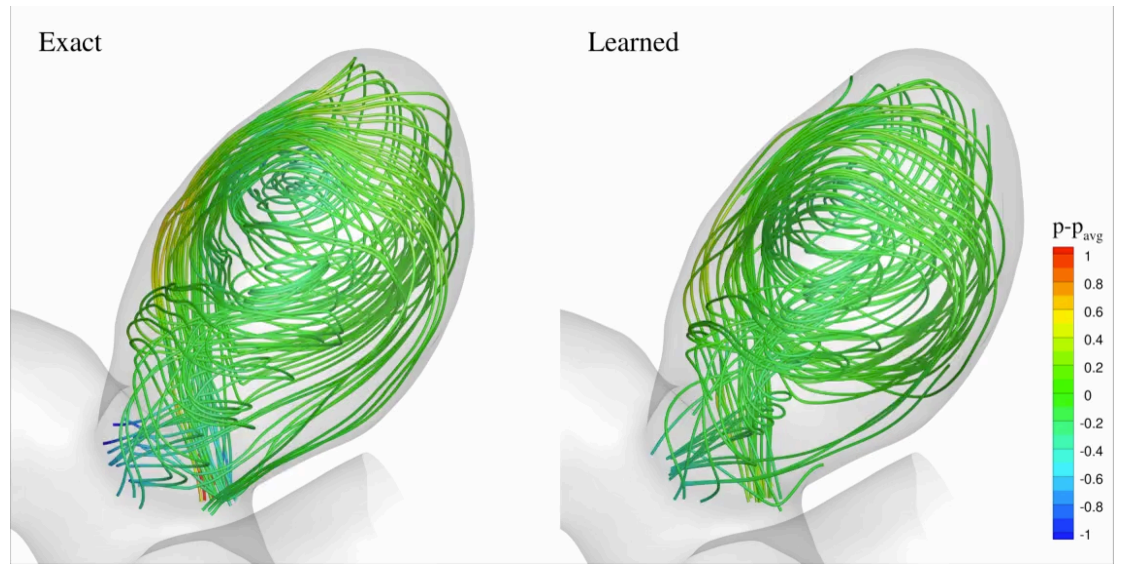
maziar.raissi@colorado.edu



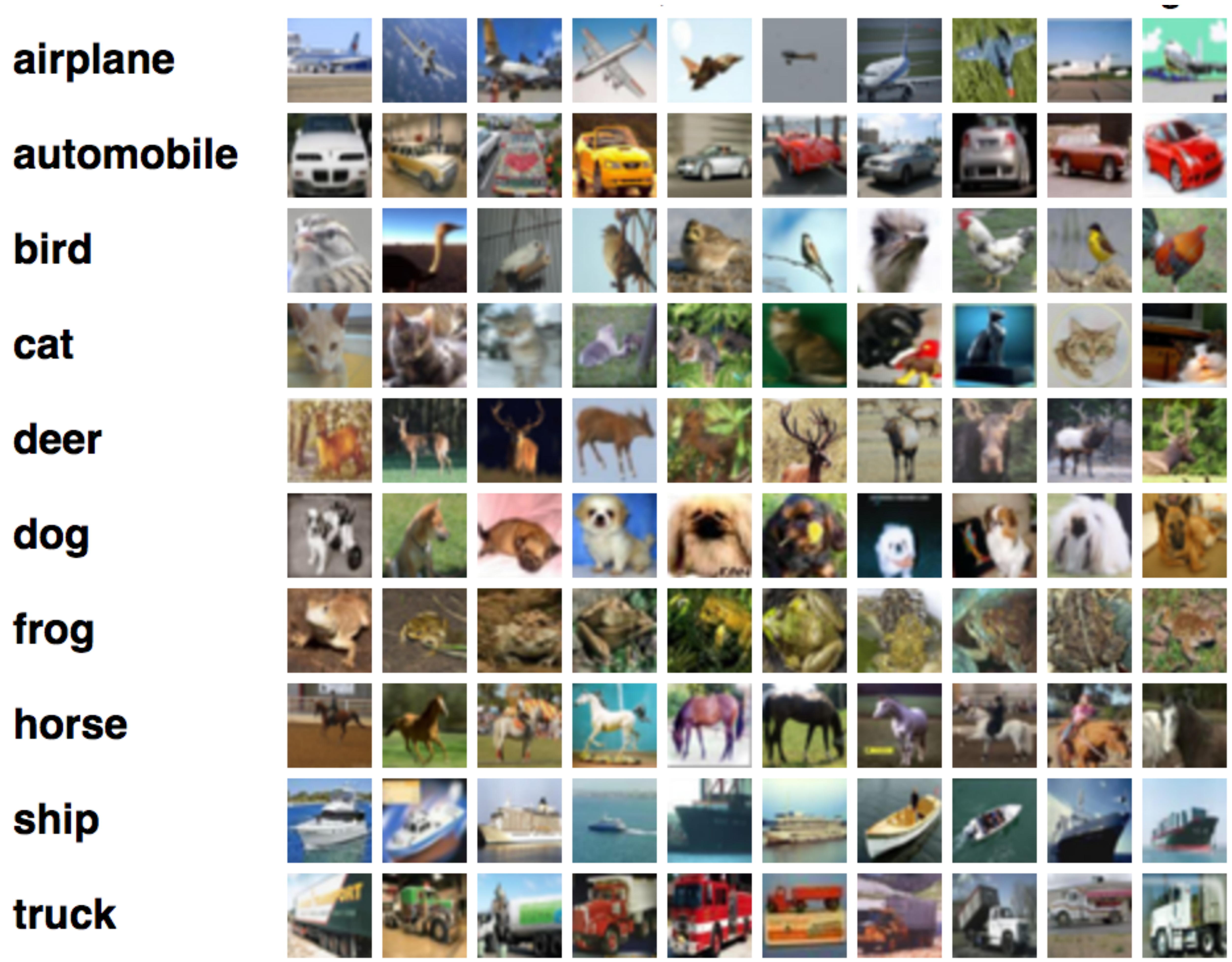


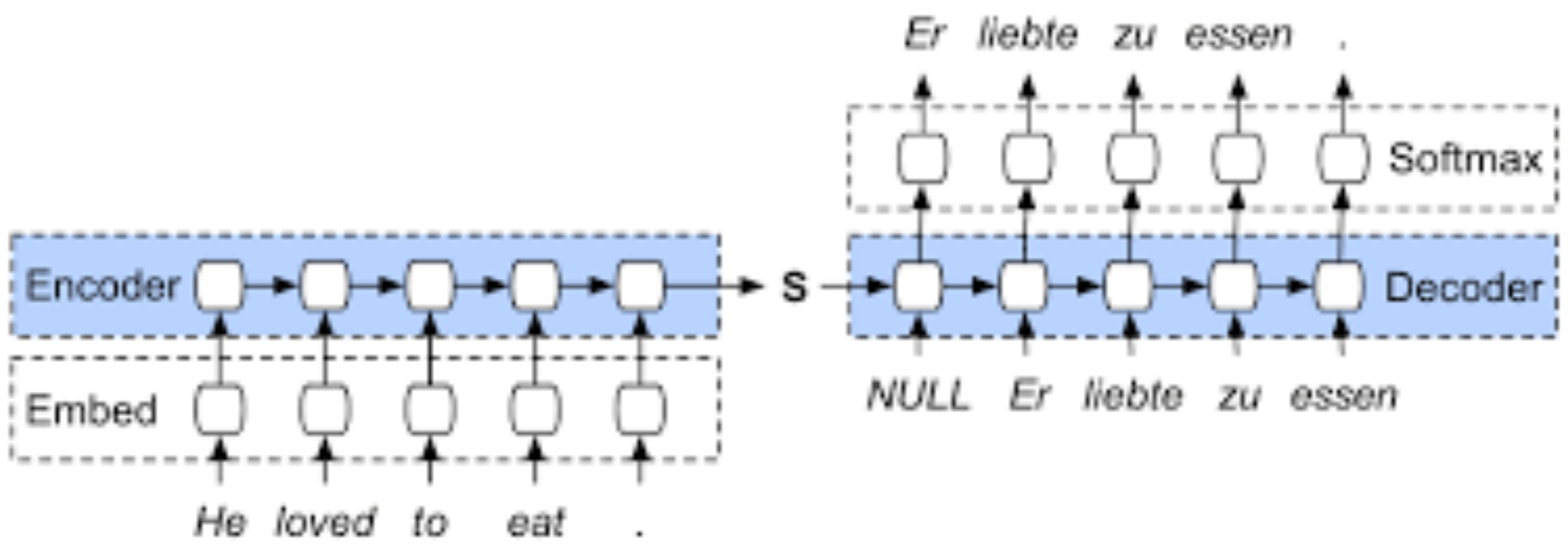


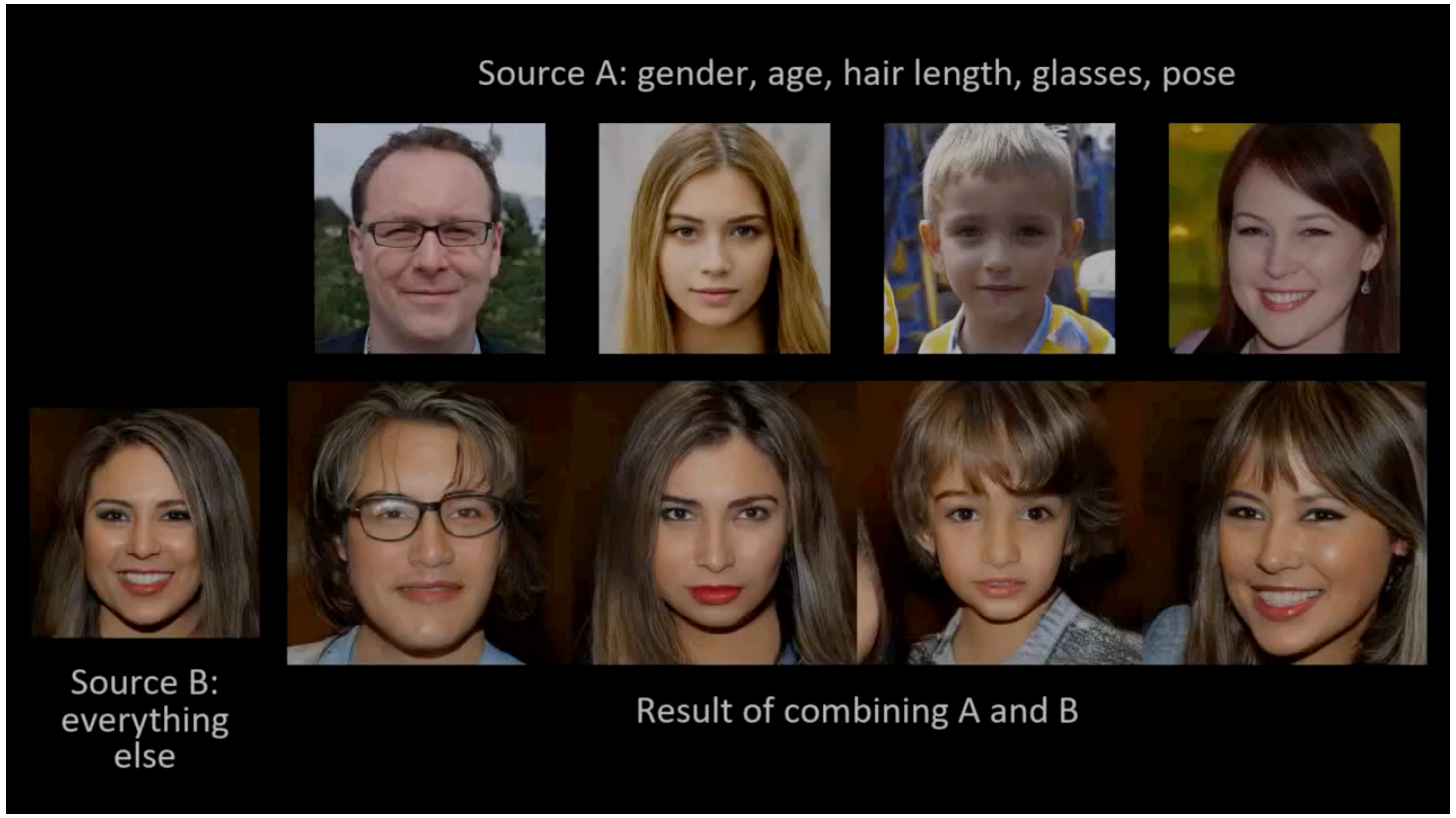




Machine/Deep Learning







https://github.com/maziarraissi/Applied-Deep-Learning

Neural Networks Regression

Data

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}\left(0, \sigma^2\right), \quad i = 1, \ldots, N$$

$$\mathbf{y} = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right)$$

Prior

$$f(x) = W^L h^L + b^L$$

$$h^L = \tanh(W^{L-1} h^{L-1} + b^{L-1})$$

$$\vdots$$

$$h^1 = \tanh(W^0 x + b^0)$$

Training

$$\mathbf{y} \sim \mathcal{N}\left(f(\mathbf{x}), \sigma^2 \mathbf{I}\right)$$

$$\min_{W,b} \; (\mathbf{y} - f(\mathbf{x}))'(\mathbf{y} - f(\mathbf{x}))$$

$$\min_{W,b} \sum_{i=1}^{N} |y_i - f(x_i)|^2$$

Prediction

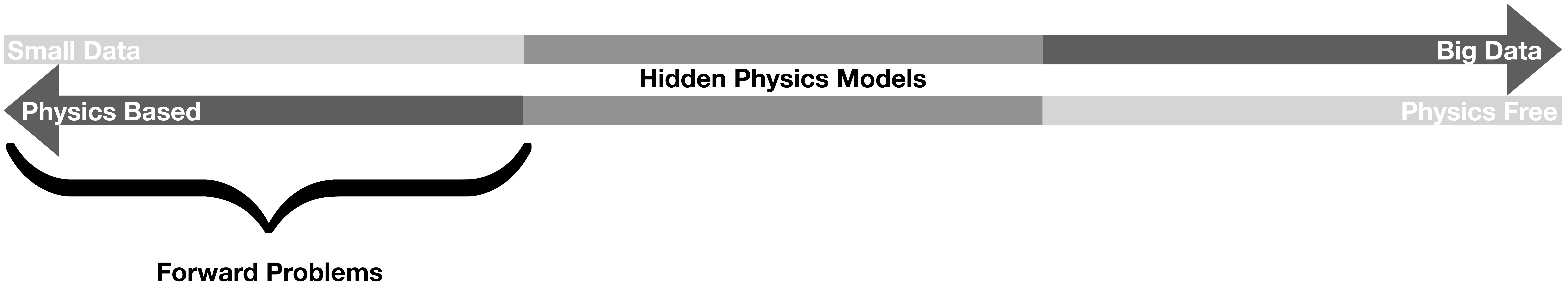
$$f(x^*)$$

The Derivative of a Neural Network is a Neural Network.

Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Boulder

Forward Problems



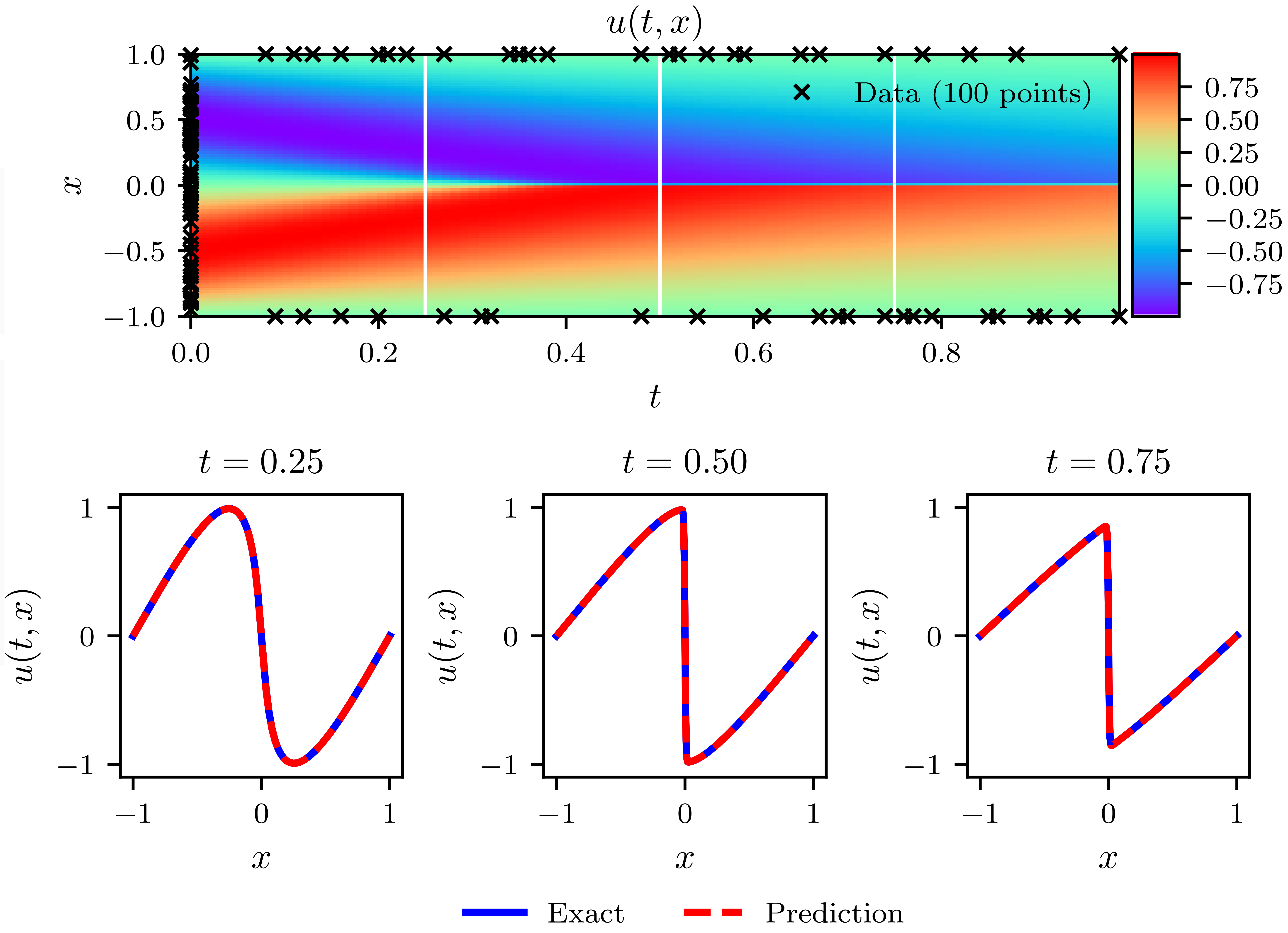
Physics Informed Neural Networks (PINNs)

$$u_t + uu_x - (0.01/\pi)u_{xx} = 0$$

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

```
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
```

$$\sum_{i=1}^{N_u} |u(t_u^i, x_u^i) - u^i|^2 + \sum_{i=1}^{N_f} |f(t_f^i, x_f^i)|^2$$

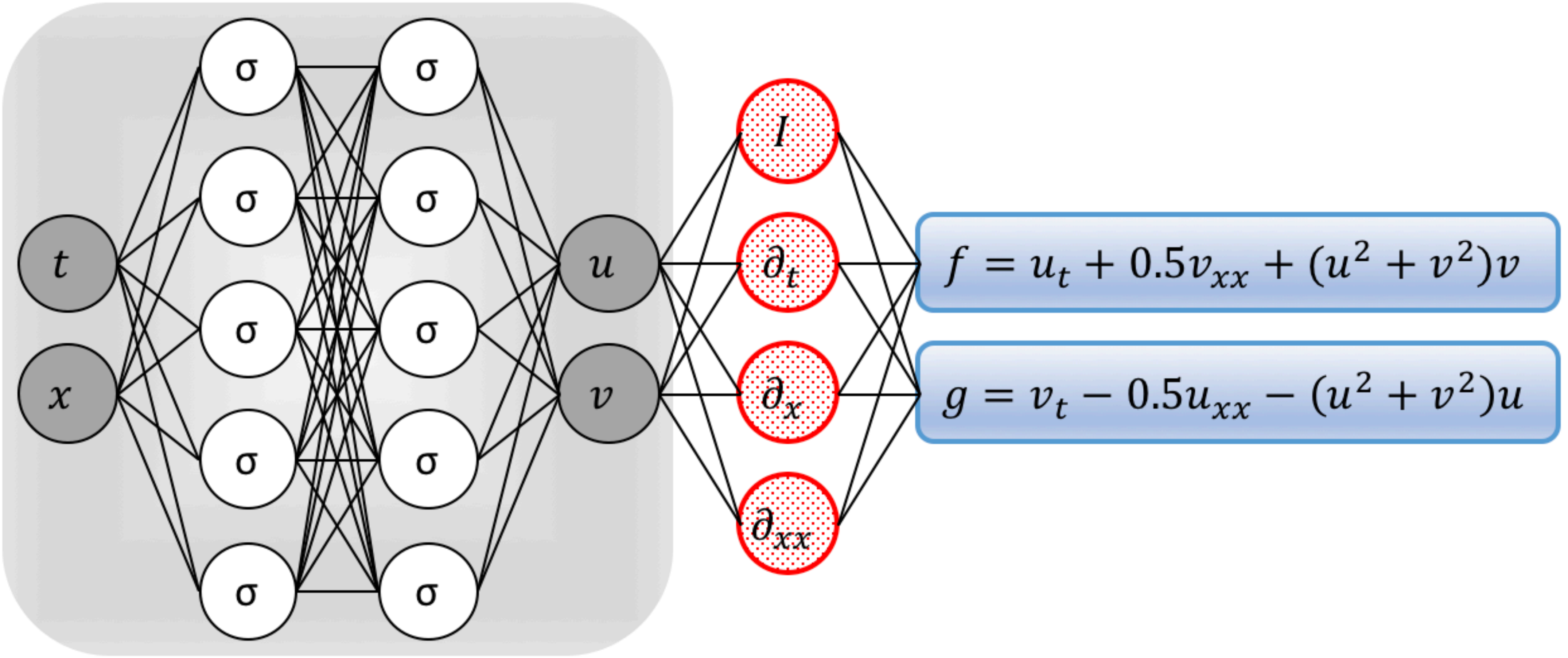


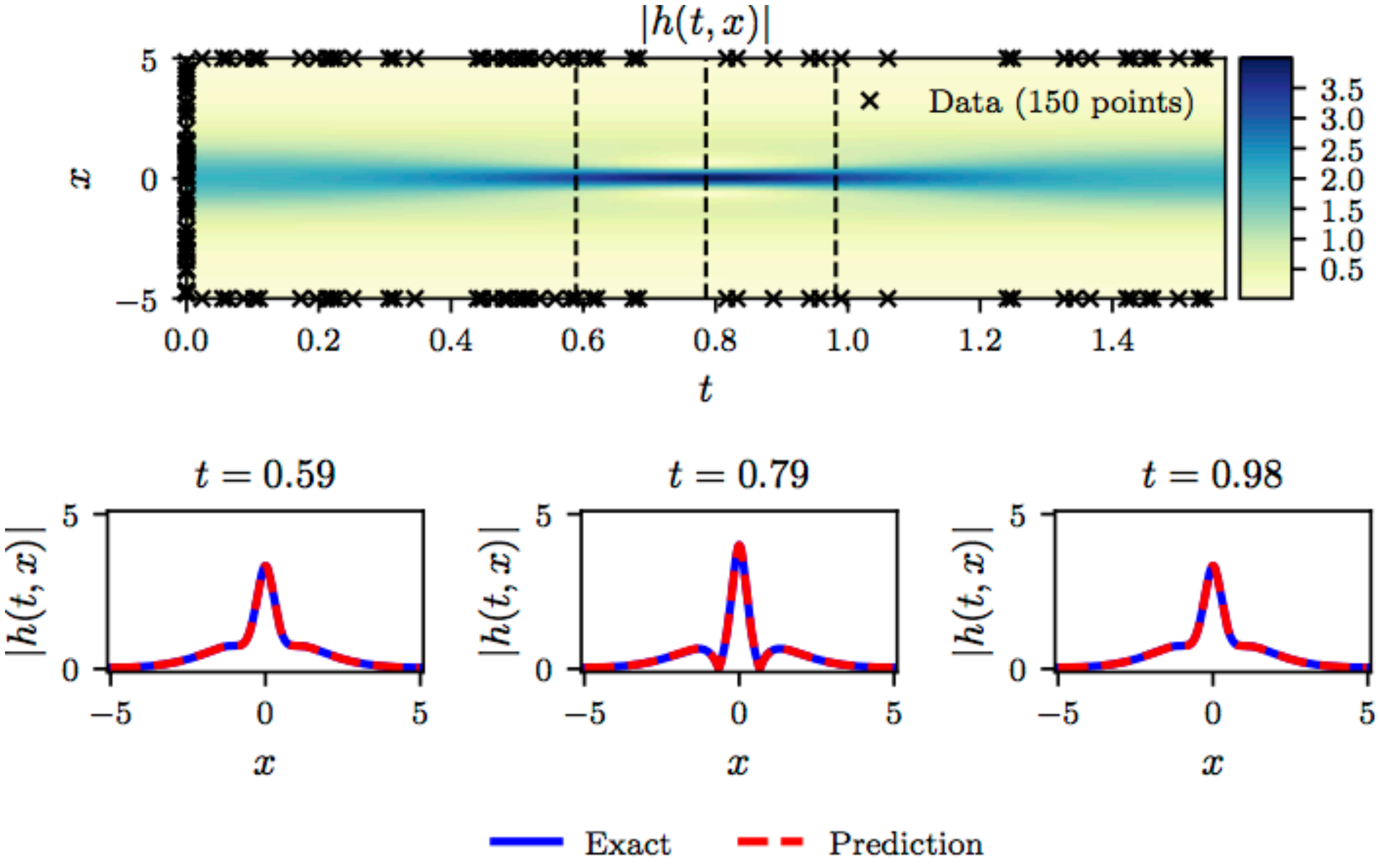
Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations." *Journal of Computational Physics* (2018).

Physics Informed Neural Networks (PINNs)

$$ih_t + 0.5h_{xx} + |h|^2 h = 0$$

$$u = \mathrm{Real}(h), \quad v = \mathrm{Imag}(h)$$





$$MSE = MSE_0 + MSE_b + MSE_f + MSE_g$$

Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations." *Journal of Computational Physics* (2018).

Boulder

Physics Informed Neural Networks (PINNs)

	FEM/FVM/FDM	PINNs	ROMs
Solution Space	Basis Functions	Neural Networks	Smart Basis Functions
Differential Operators	Discretization/Weak-form	Automatic Differentiation	Discretization/Weak-form
Solver	Linear/non-linear/Iterative	Gradient Descent (Training)	Linear/non-linear/Iterative
Evaluate	Interpolation	Inference	Interpolation

Forward-Backward Stochastic Neural Networks

Deep Learning of High-dimensional Partial Differential Equations

$$
\begin{aligned}
dX_t &= \mu(t, X_t, Y_t, Z_t)dt + \sigma(t, X_t, Y_t)dW_t \\
X_0 &= \xi \\
dY_t &= \varphi(t, X_t, Y_t, Z_t)dt + Z_t'\sigma(t, X_t, Y_t)dW_t \\
Y_T &= g(X_T)
\end{aligned}
$$

$$
\begin{aligned}
Y_t &= u(t, X_t) \\
Z_t &= Du(t, X_t)
\end{aligned}
$$

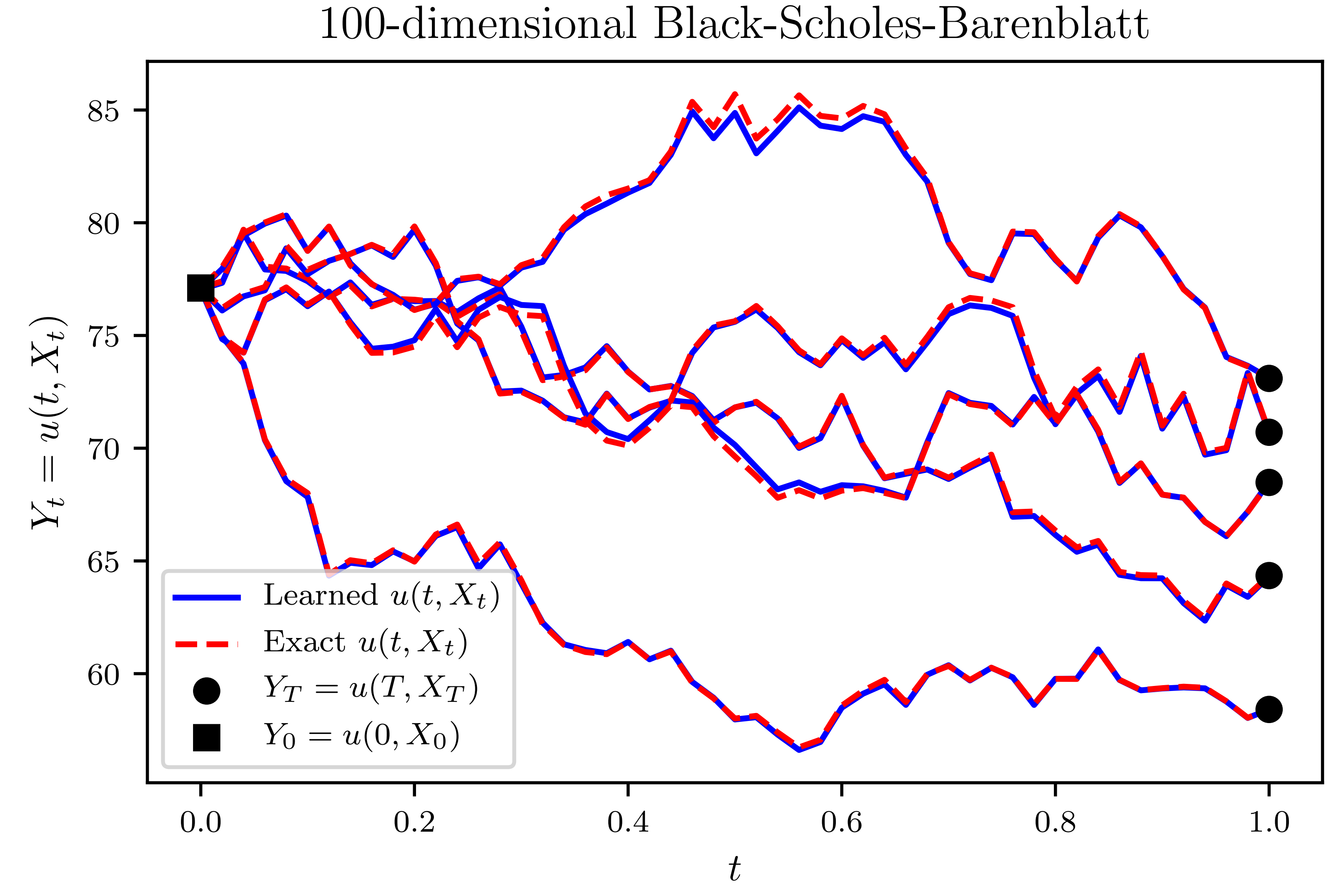
Forward-Backward Stochastic Differential Equation

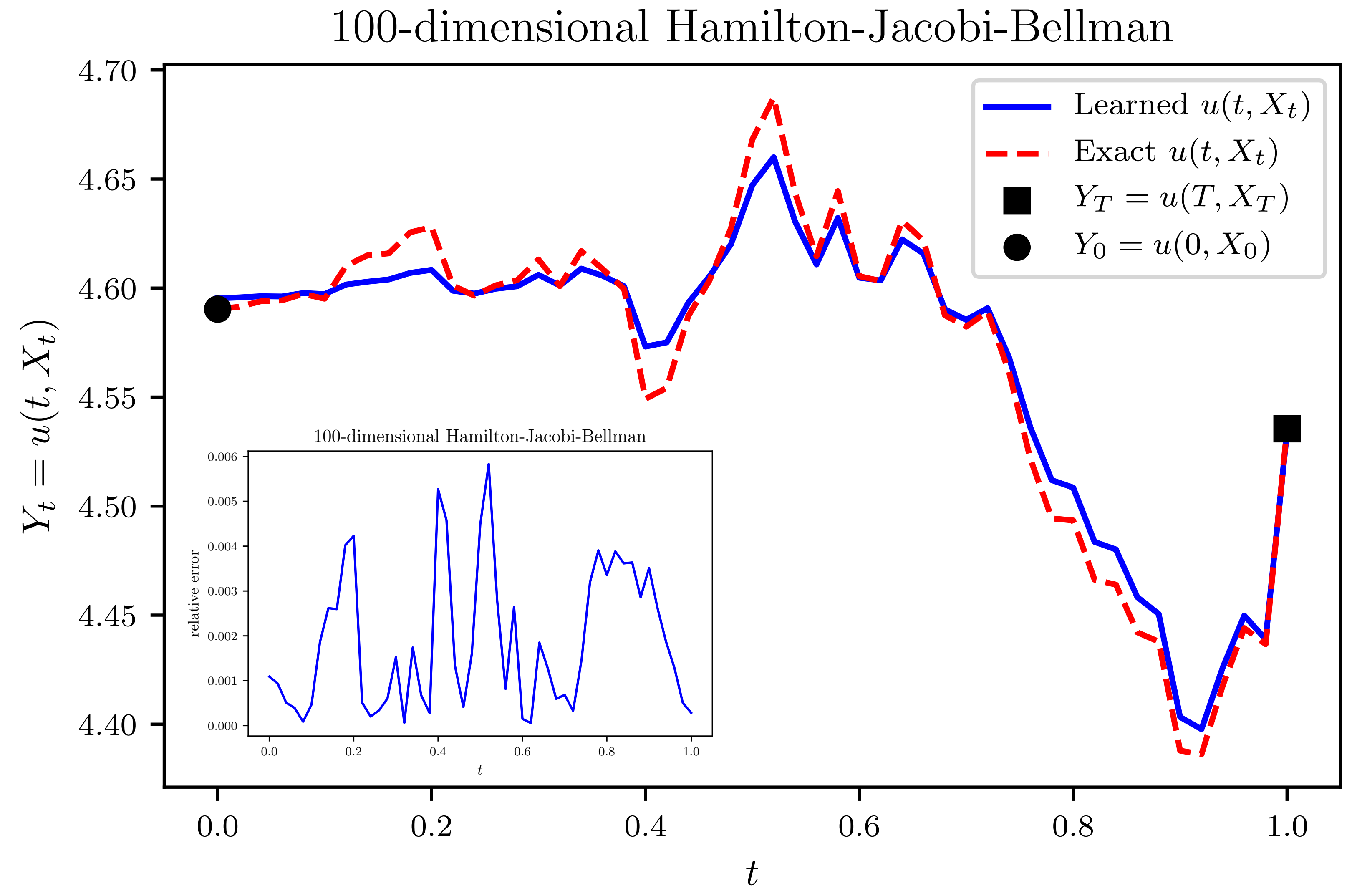
$u(t,x) \longrightarrow$ **Deep Neural Network**

$Du(t,x) \longrightarrow$ **Automatic Differentiation**

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \varphi(t, x, u, Du) - \mu(t, x, u, Du)'Du - \frac{1}{2}\mathrm{Tr}[\sigma(t, x, u)\sigma(t, x, u)'D^2u] \\
u(T, x) &= g(x)
\end{aligned}
$$

Quasi-Linear Partial Differential Equation

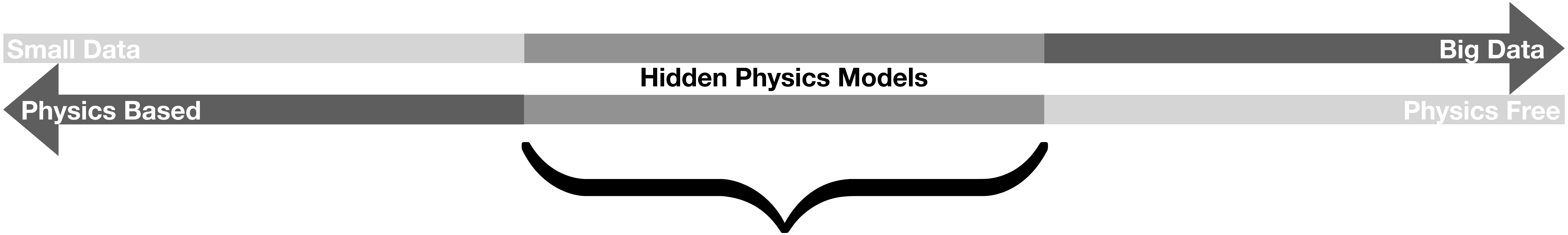




Raissi, Maziar. "Forward-Backward Stochastic Neural Networks: Deep Learning of High-dimensional Partial Differential Equations." *arXiv preprint arXiv:1804.07010* (2018).

Boulder

Inverse & Data Assimilation Problems

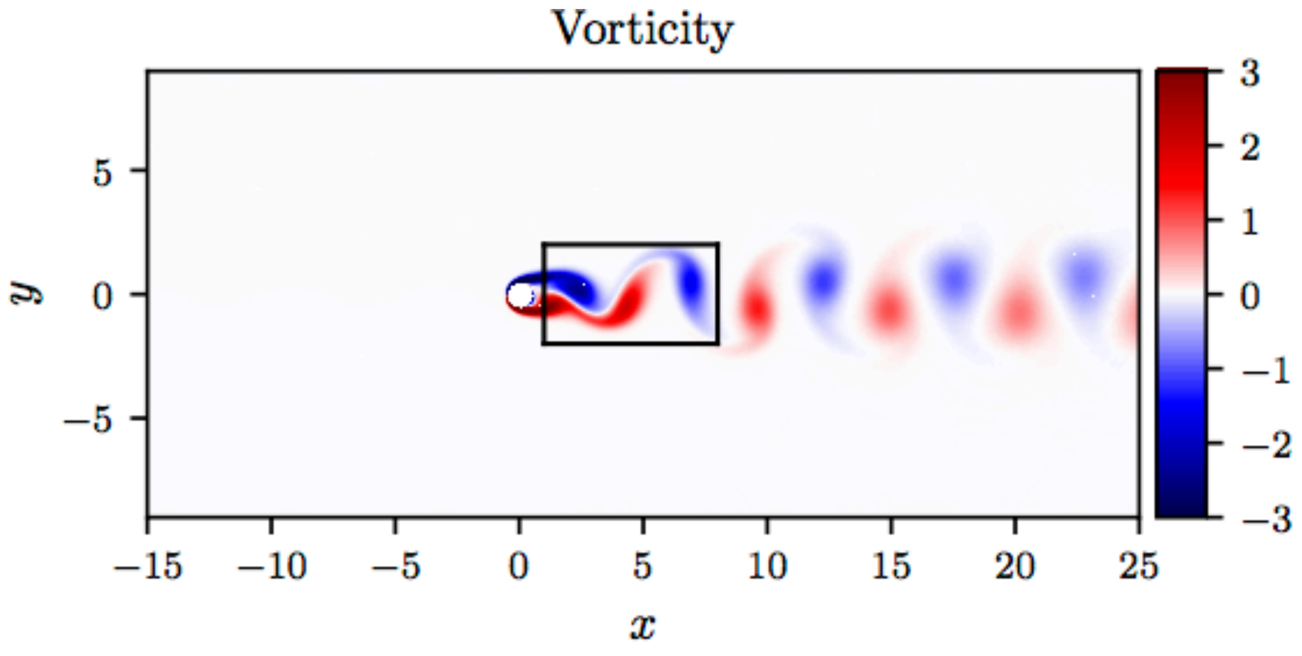


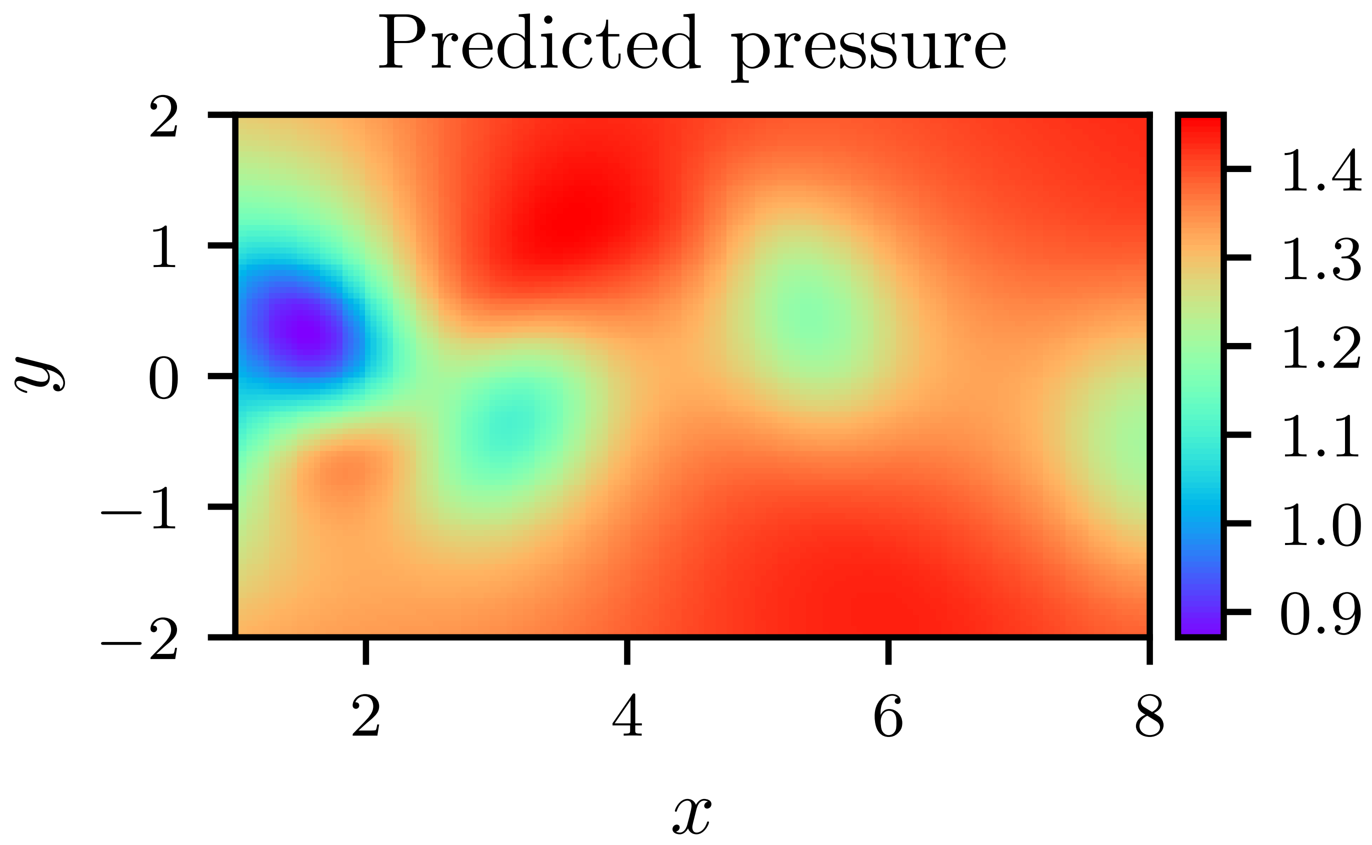
Inverse & Data Assimilation Problems

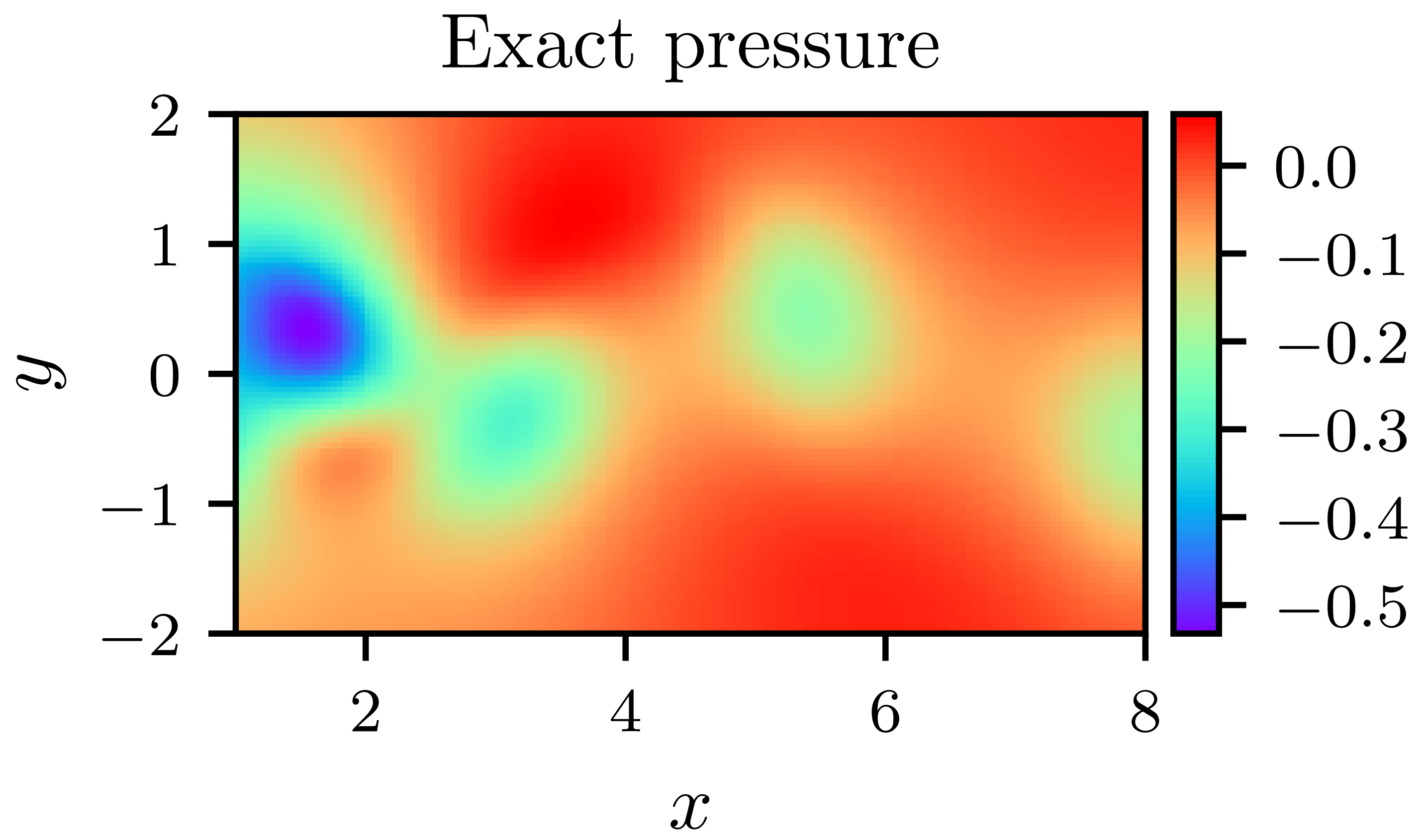
Physics Informed Neural Networks (PINNs)

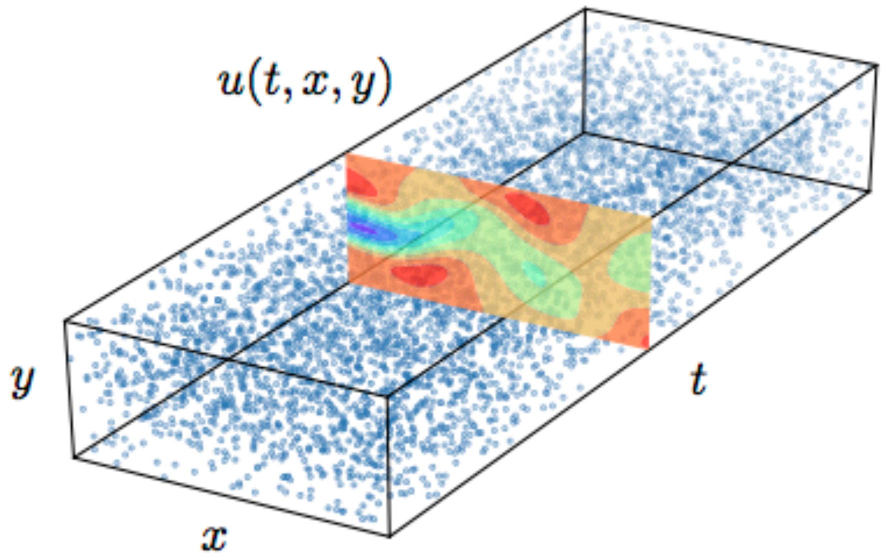
$$u_t + \lambda_1(uu_x + vu_y) = -p_x + \lambda_2(u_{xx} + u_{yy})$$
$$v_t + \lambda_1(uv_x + vv_y) = -p_y + \lambda_2(v_{xx} + v_{yy})$$
$$u_x + v_y = 0$$

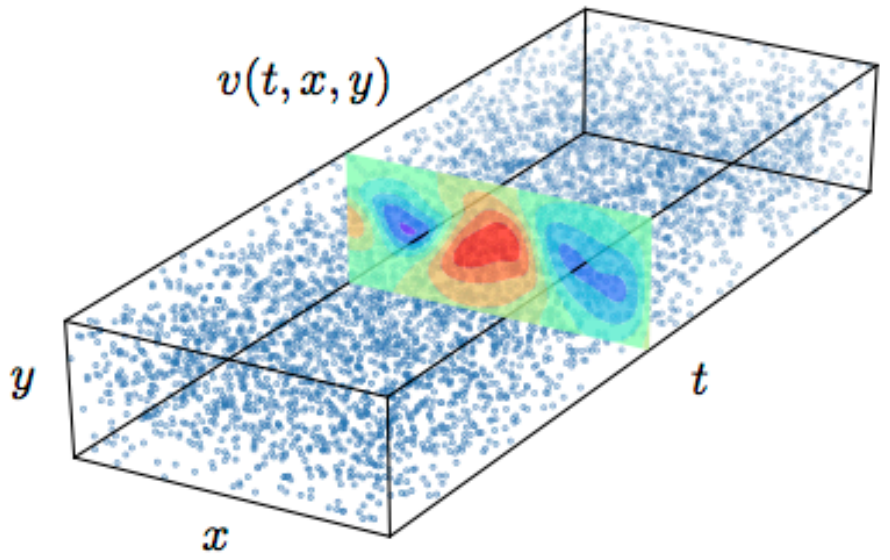
$$f := u_t + \lambda_1(uu_x + vu_y) + p_x - \lambda_2(u_{xx} + u_{yy})$$
$$g := v_t + \lambda_1(uv_x + vv_y) + p_y - \lambda_2(v_{xx} + v_{yy})$$
$$h := u_x + v_y$$







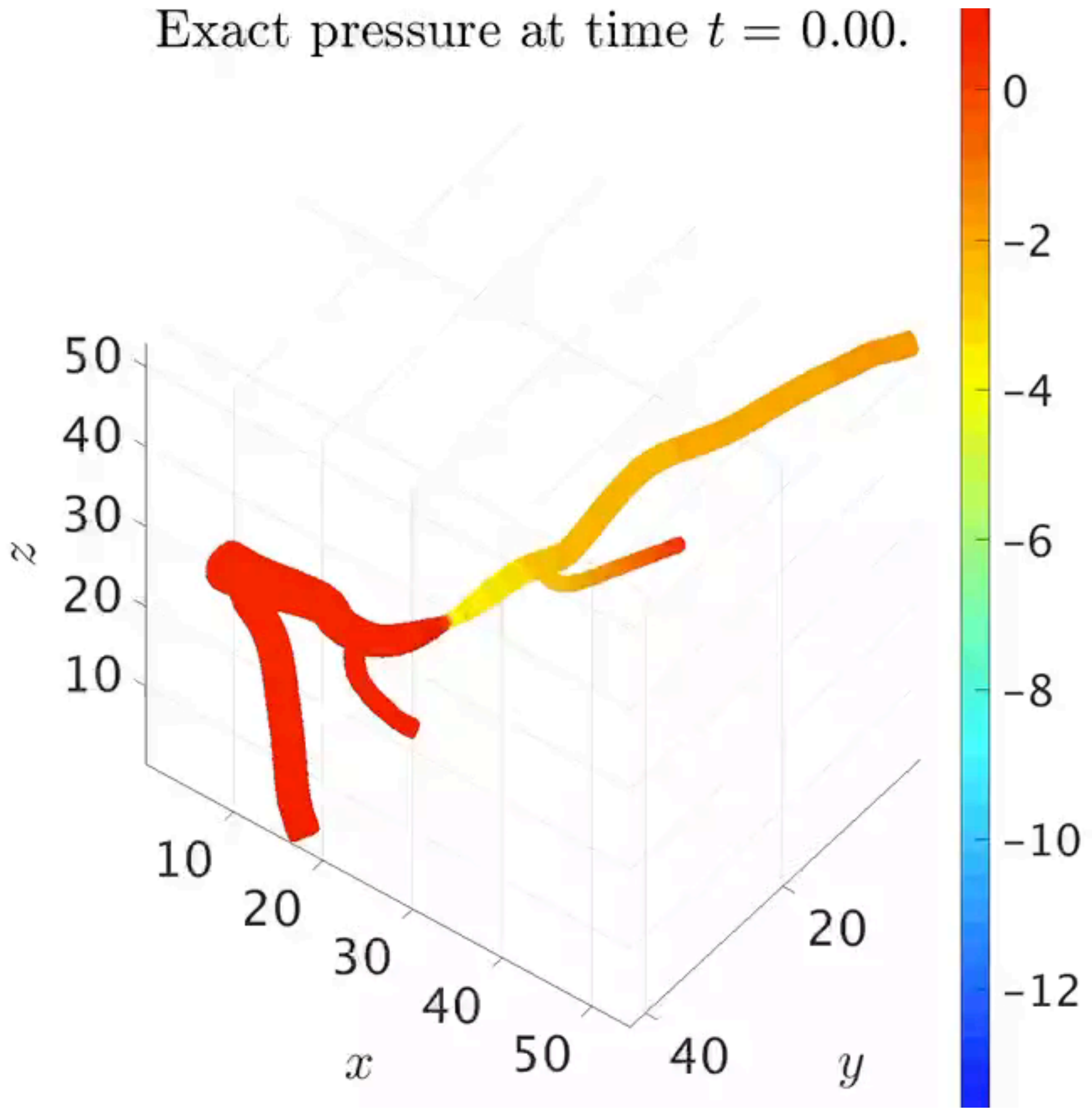


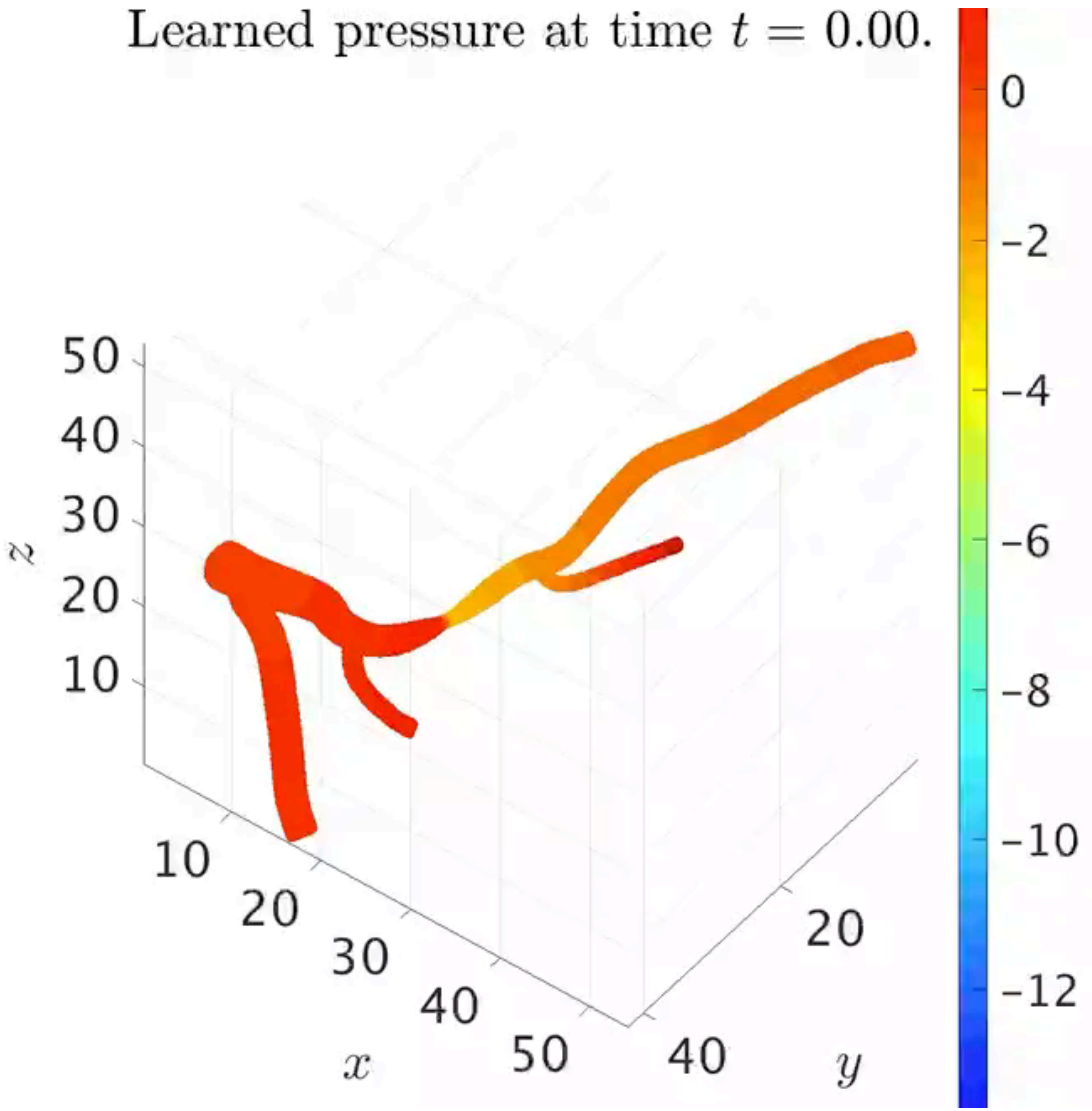


Correct PDE	$u_t + (uu_x + vu_y) = -p_x + 0.01(u_{xx} + u_{yy})$ $v_t + (uv_x + vv_y) = -p_y + 0.01(v_{xx} + v_{yy})$
Identified PDE (clean data)	$u_t + 0.999(uu_x + vu_y) = -p_x + 0.01047(u_{xx} + u_{yy})$ $v_t + 0.999(uv_x + vv_y) = -p_y + 0.01047(v_{xx} + v_{yy})$
Identified PDE (1% noise)	$u_t +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

Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations." *Journal of Computational Physics* (2018).

Physics Informed Neural Networks (PINNs)

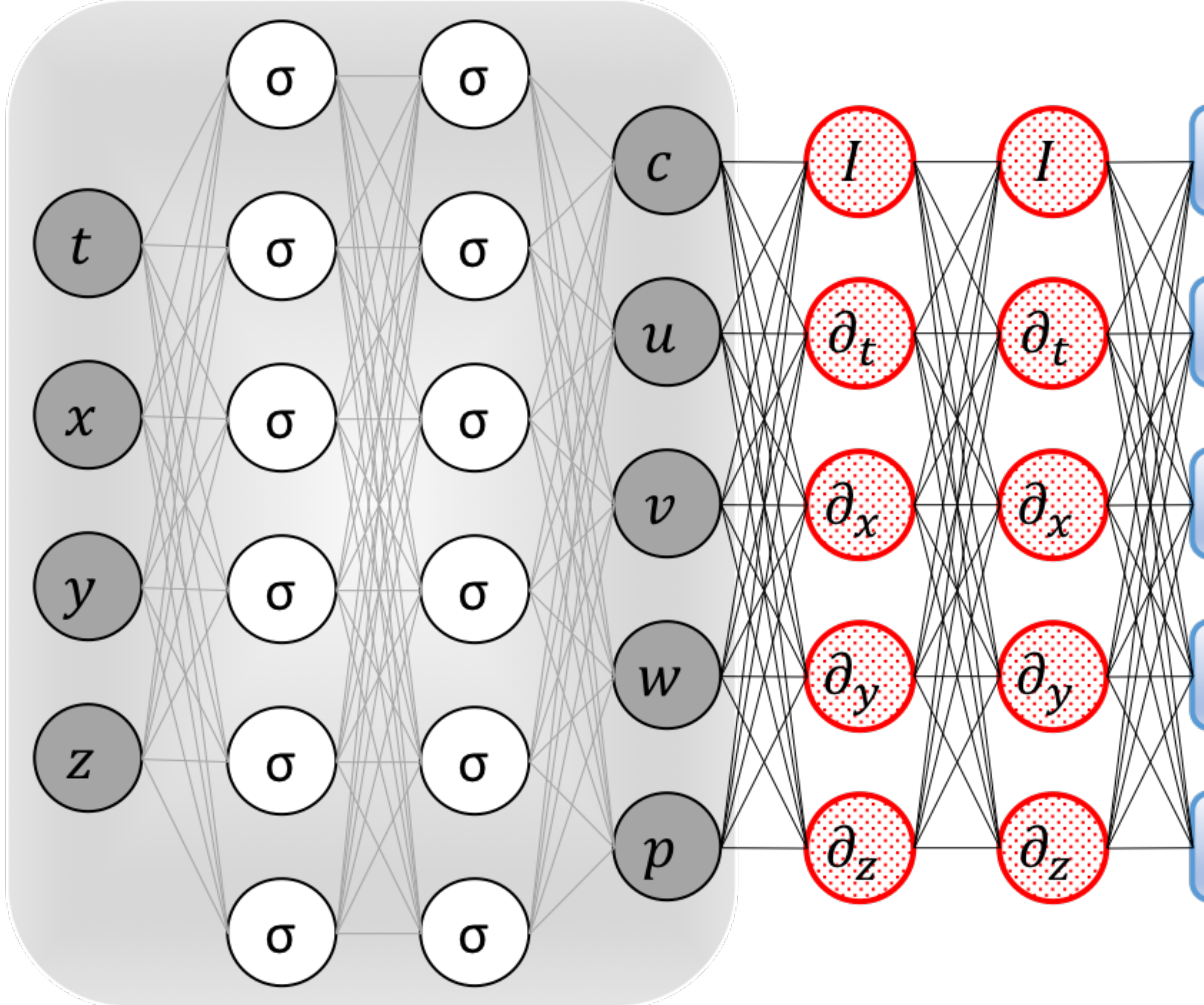




Hidden Fluid Mechanics

Pec = 40 **Pec = 100** **Pec = 250**

Hidden Fluid Mechanics



$$e_1 = c_t + uc_x + vc_y + wc_z - Pec^{-1}(c_{xx} + c_{yy} + c_{zz})$$

$$e_2 = u_t + uu_x + vu_y + wu_z + p_x - Re^{-1}(u_{xx} + u_{yy} + u_{zz})$$

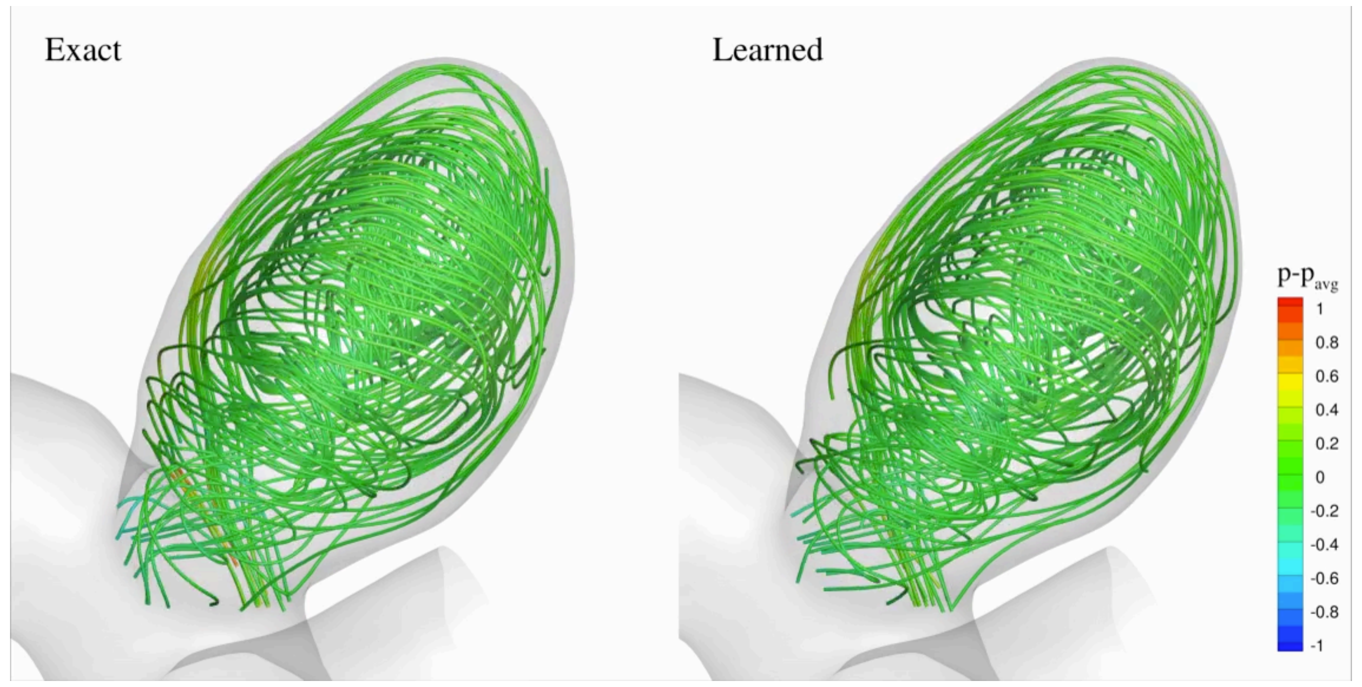
$$e_3 = v_t + uv_x + vv_y + wv_z + p_y - Re^{-1}(v_{xx} + v_{yy} + v_{zz})$$

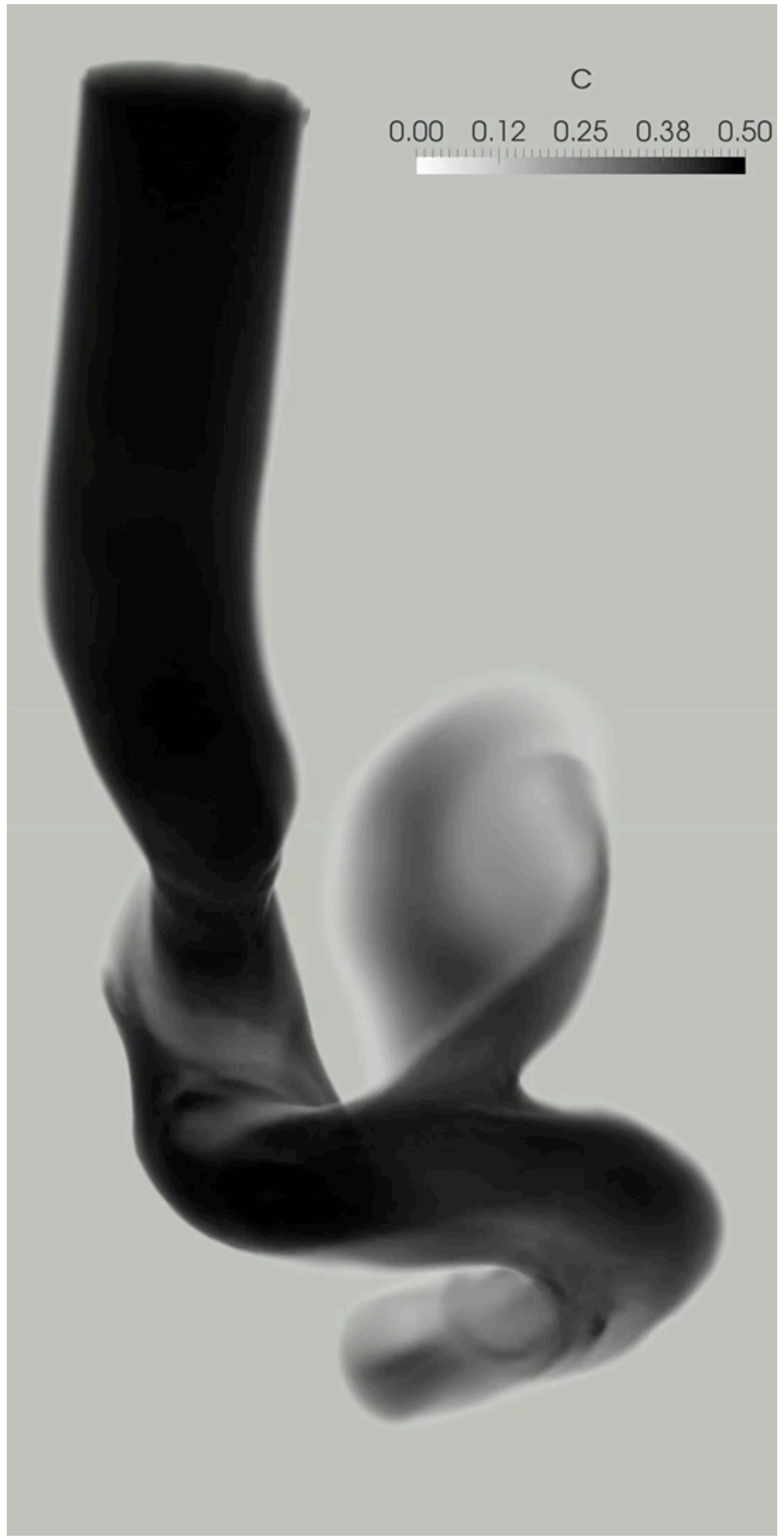
$$e_4 = w_t + uw_x + vw_y + ww_z + p_z - Re^{-1}(w_{xx} + w_{yy} + w_{zz})$$

$$e_5 = u_x + v_y + w_z$$

Hidden Physics

$$SSE = \sum_{n=1}^{N} |c(t^n, x^n, y^n, z^n) - c^n|^2 + \sum_{i=1}^{5} \sum_{n=1}^{N} |e_i(t^n, x^n, y^n, z^n)|^2$$

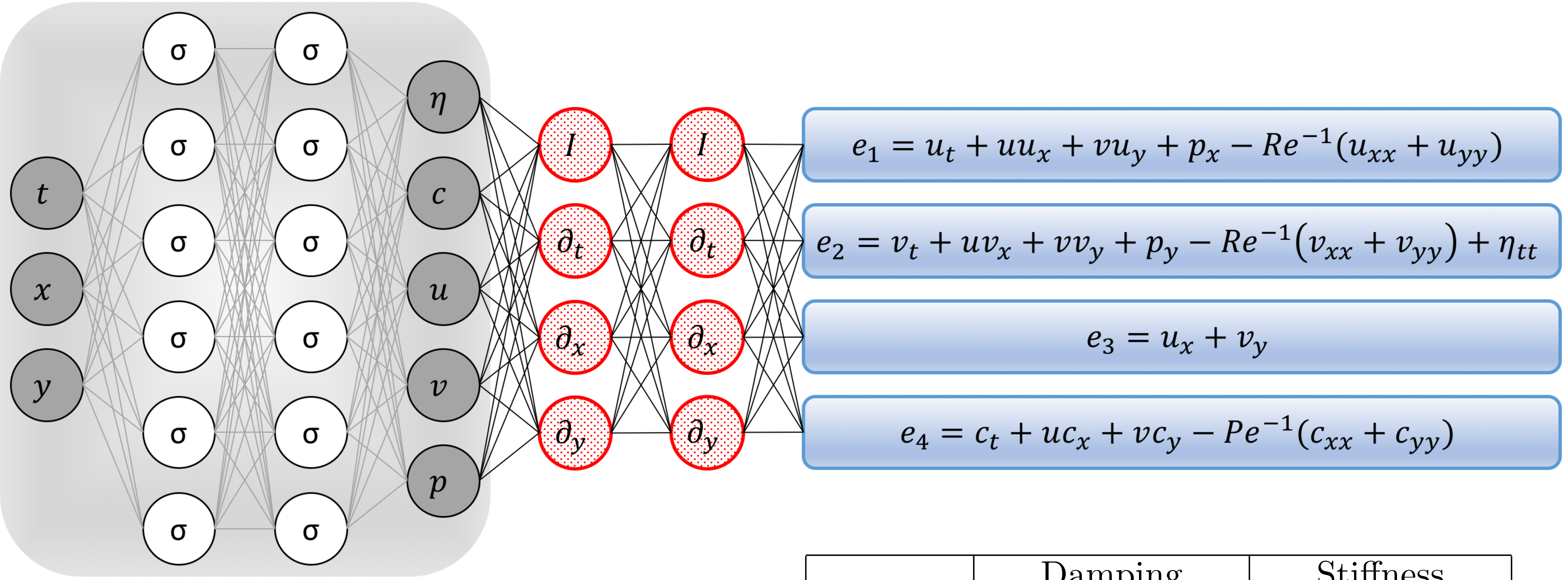




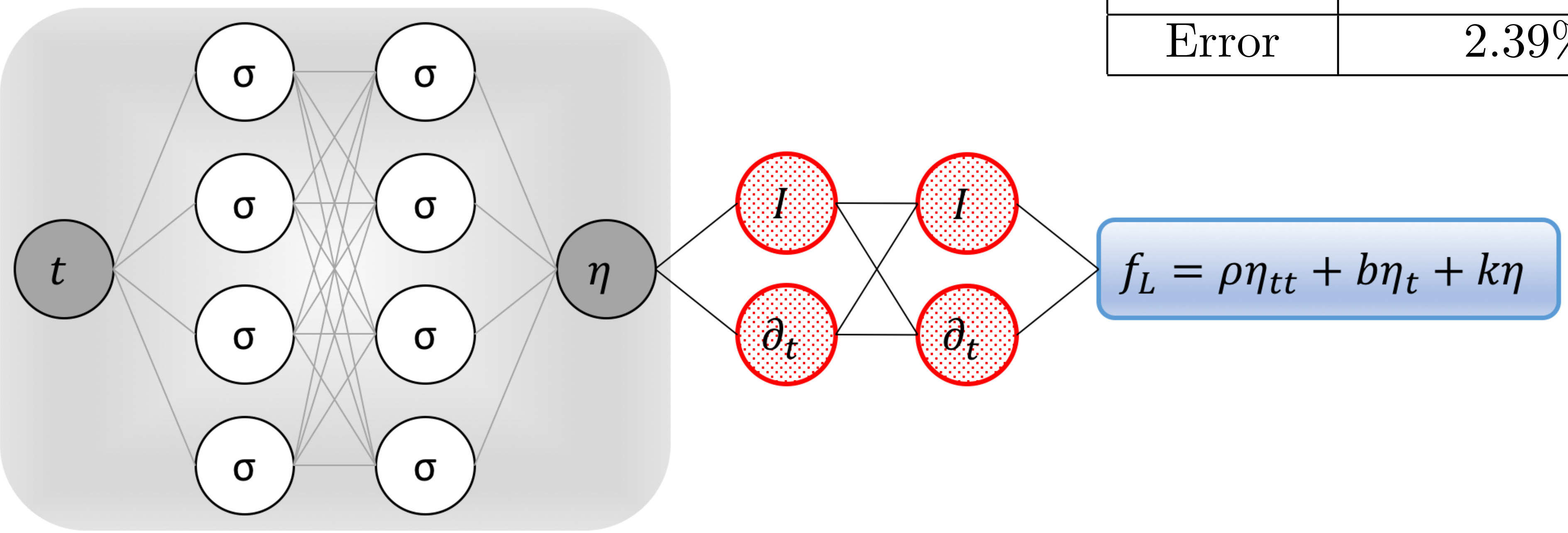
Raissi, Maziar, Alireza Yazdani, and George Em Karniadakis. "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations." *Science* (2020).

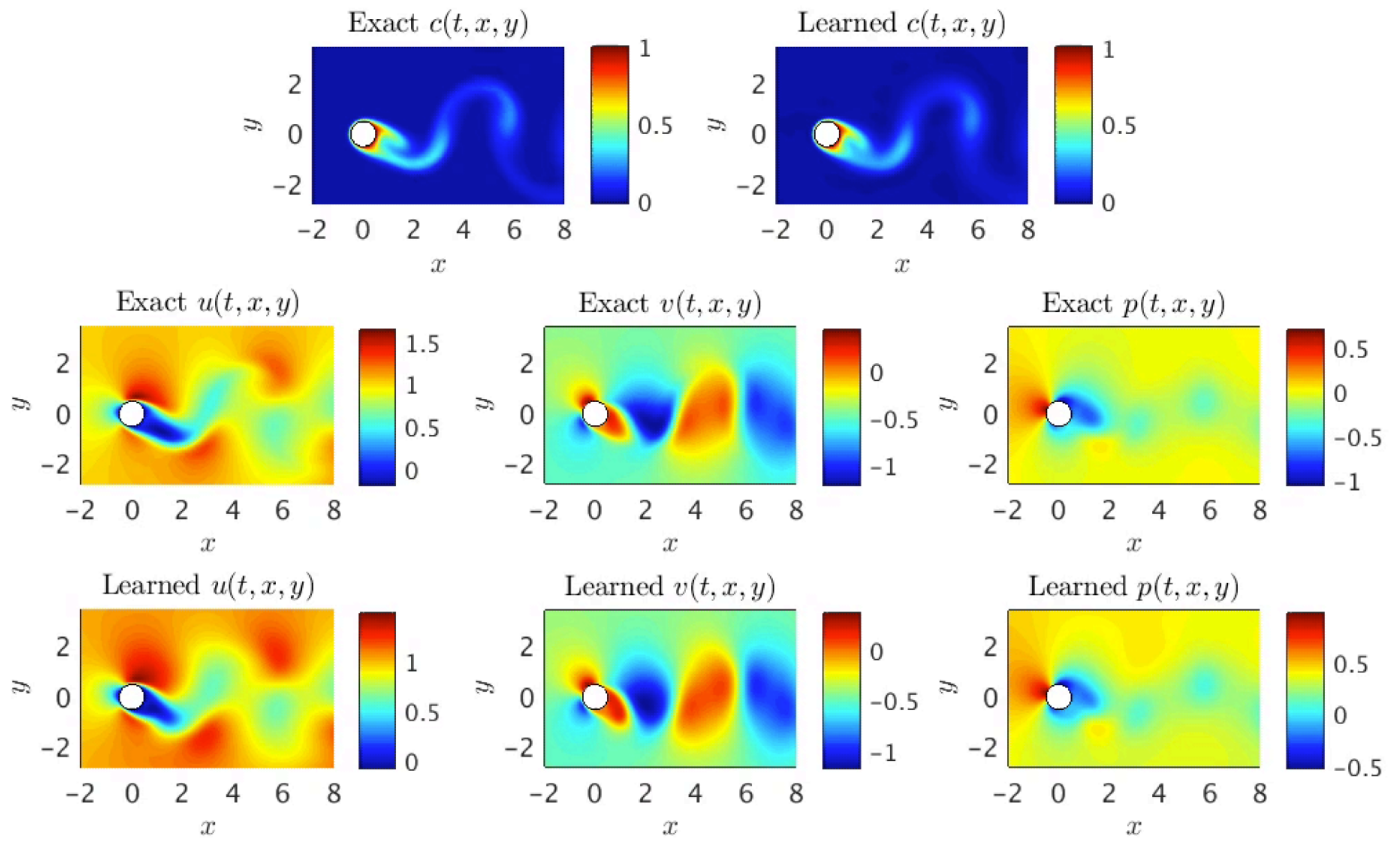
Boulder

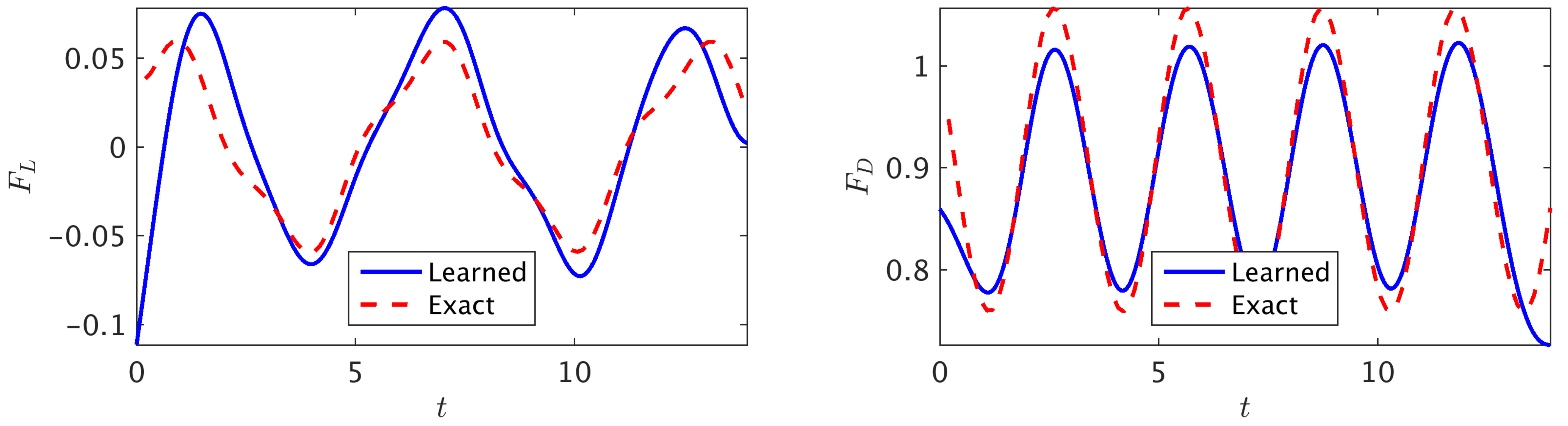
Deep Learning of Vortex Induced Vibrations



	Damping	Stiffness
Exact	$b = 0.084$	$k = 2.2020$
Learned	$b = 0.08600664$	$k = 2.2395933$
Error	2.39%	1.71%



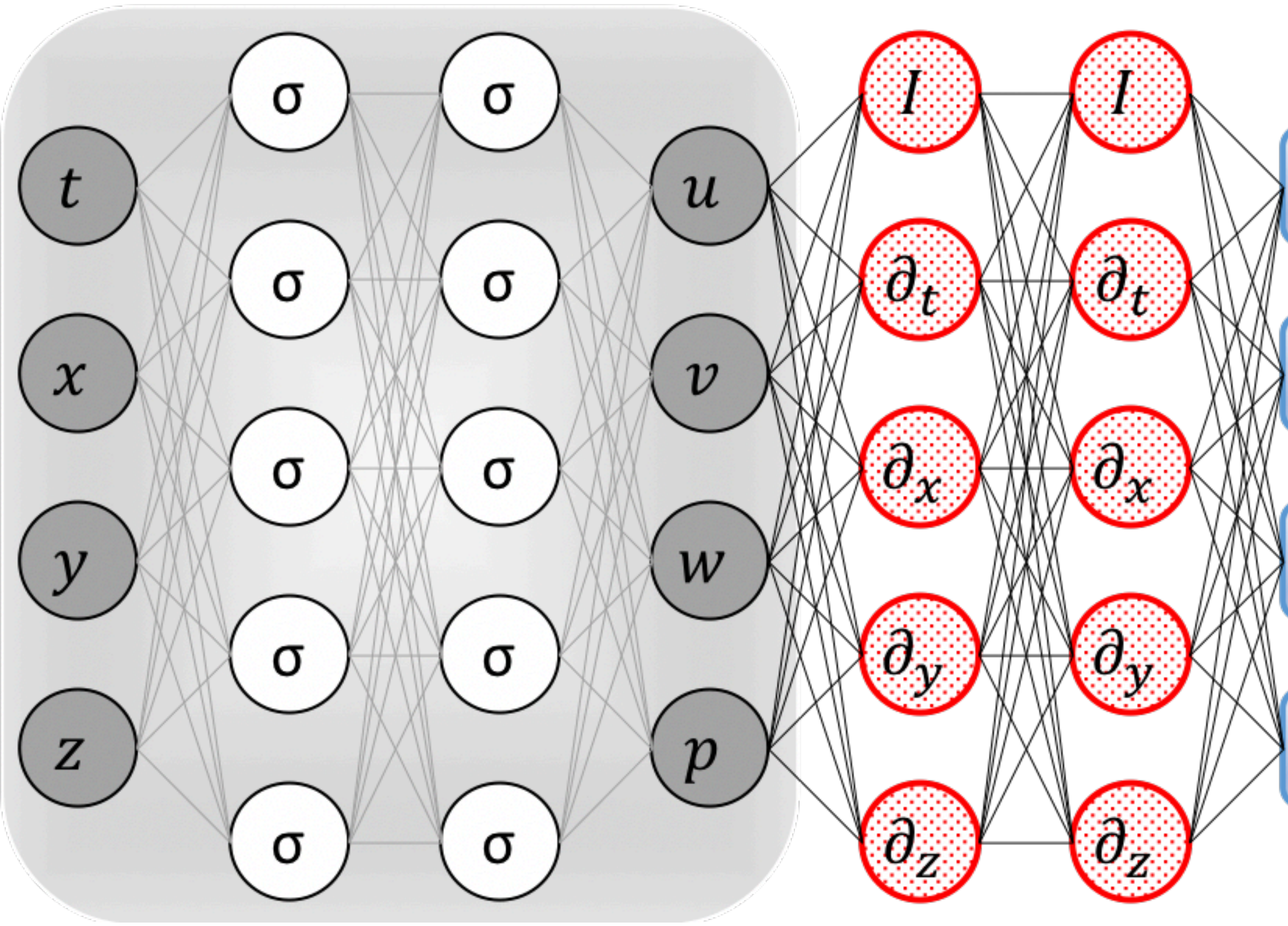




Raissi, Maziar, Zhicheng Wang, Michael S. Triantafyllou, and George Em Karniadakis. "Deep Learning of Vortex Induced Vibrations." *Journal of Fluid Mechanics* (2018).

Boulder

Eulerian-Lagrangian Neural Networks

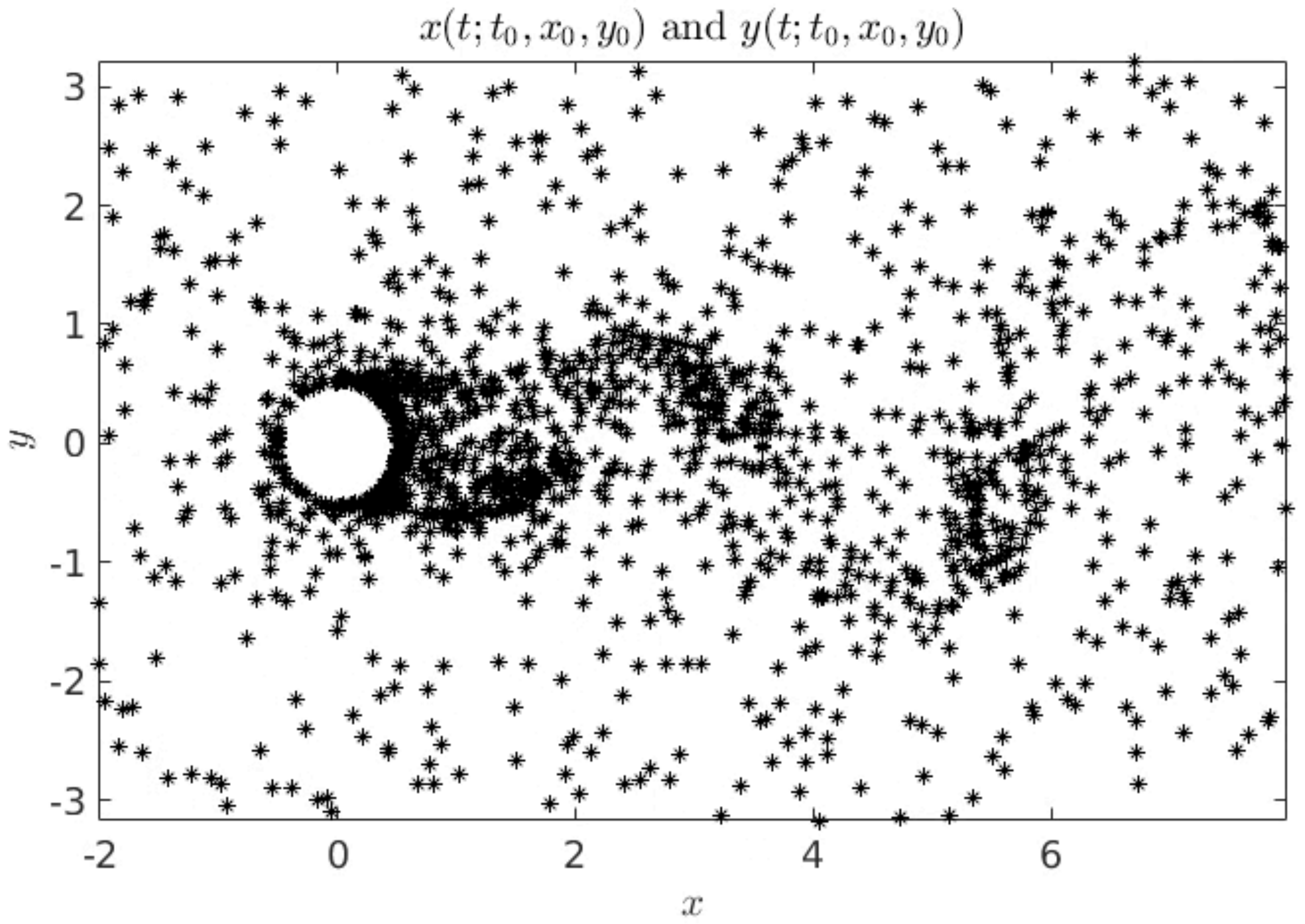


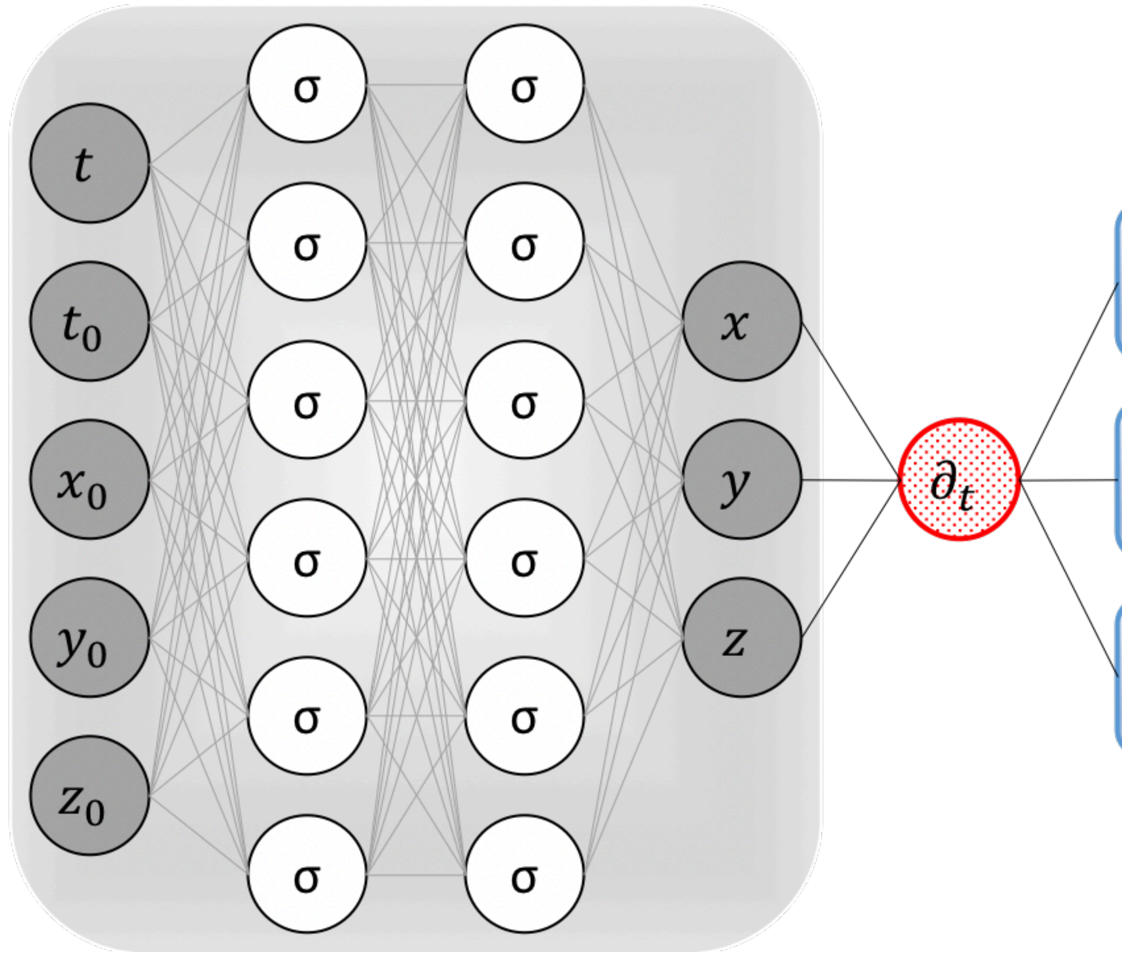
$$e_1 = u_t + uu_x + vu_y + wu_z + p_x - Re^{-1}(u_{xx} + u_{yy} + u_{zz})$$

$$e_2 = v_t + uv_x + vv_y + wv_z + p_y - Re^{-1}(v_{xx} + v_{yy} + v_{zz})$$

$$e_3 = w_t + uw_x + vw_y + ww_z + p_z - Re^{-1}(w_{xx} + w_{yy} + w_{zz})$$

$$e_4 = u_x + v_y + w_z$$

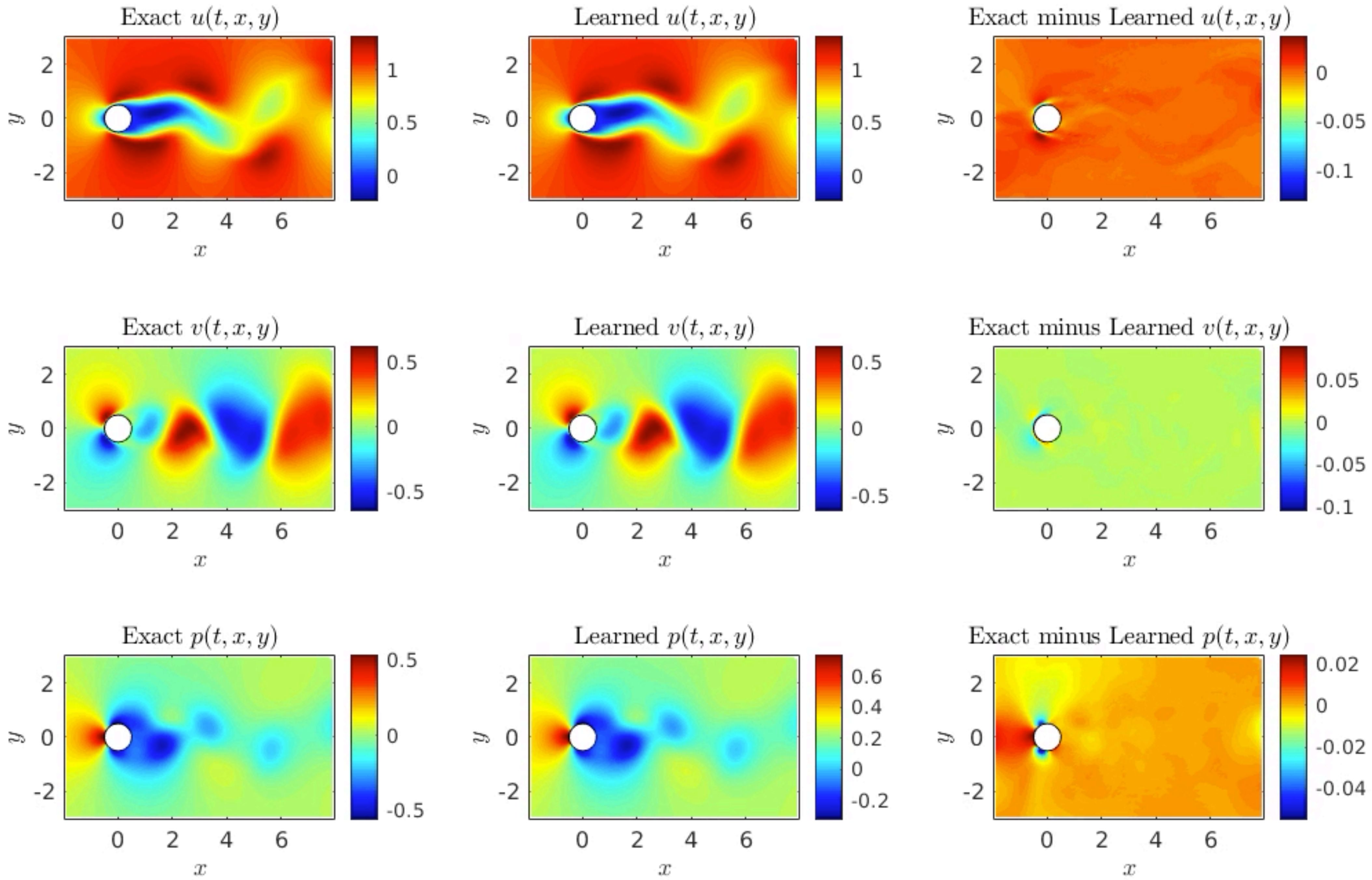




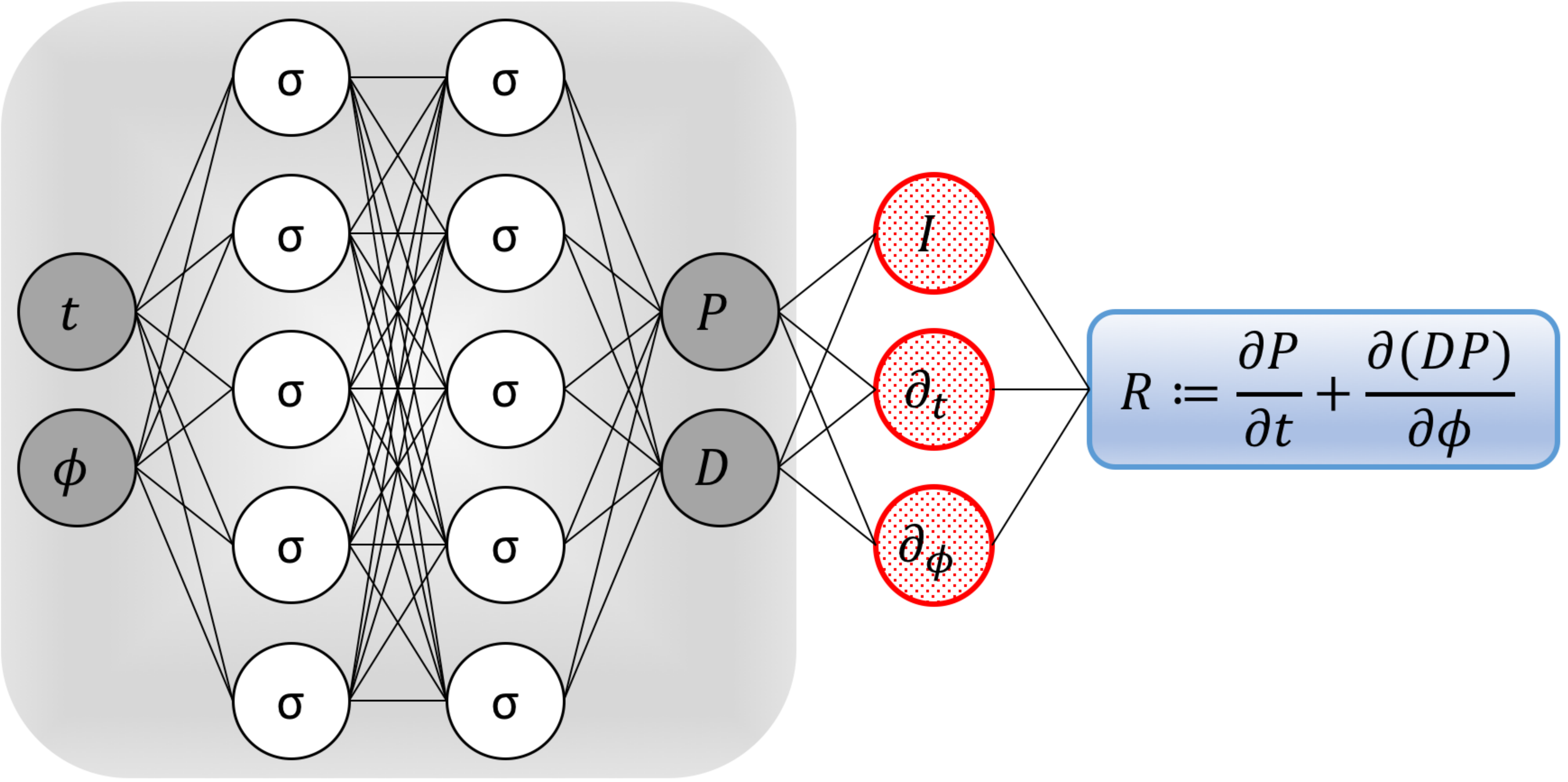
$$f_1 = \frac{dx}{dt} - u(t, x, y, z)$$

$$f_2 = \frac{dy}{dt} - v(t, x, y, z)$$

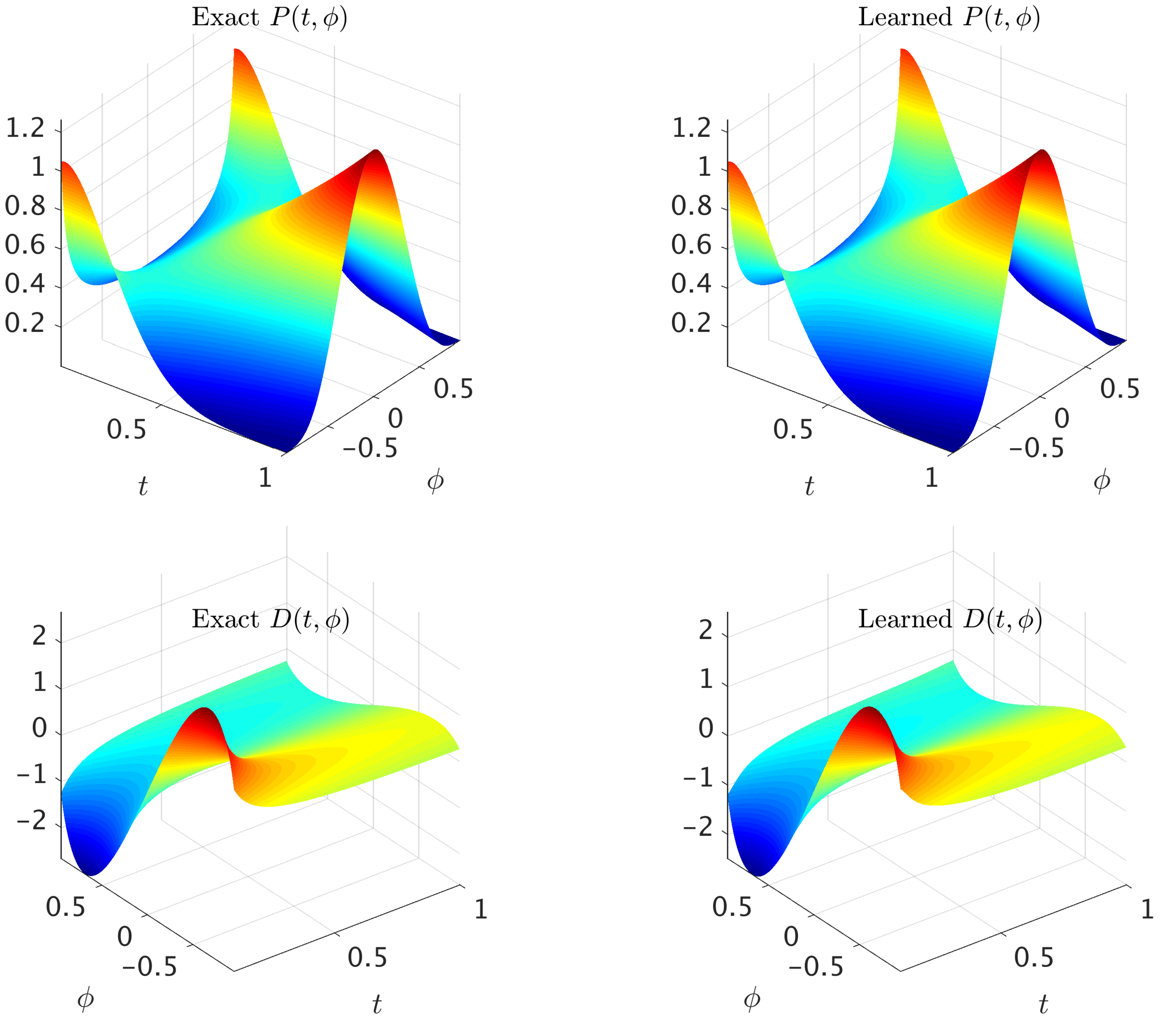
$$f_3 = \frac{dz}{dt} - w(t, x, y, z)$$



Deep Learning of Turbulent Scalar Mixing

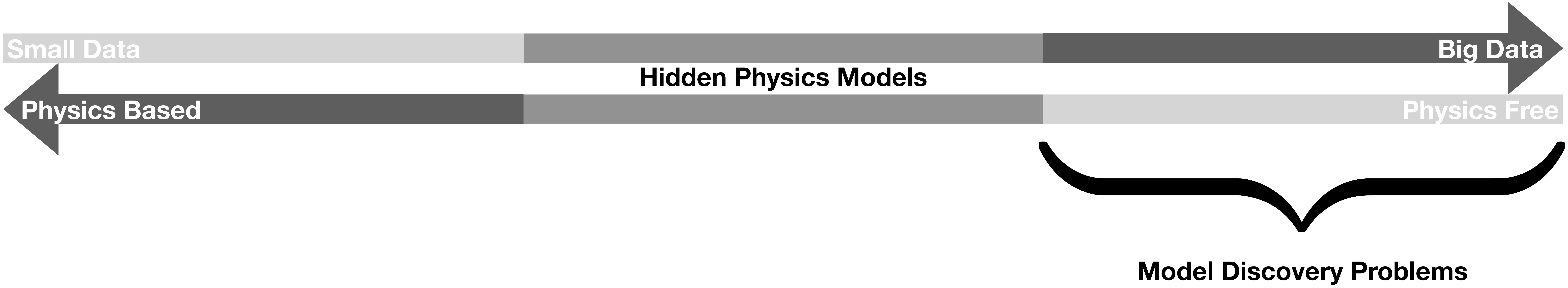


$$SSE = \sum_{n=1}^{N} |P(t^n, \phi^n) - P^n|^2 + \sum_{n=1}^{N} |R(t^n, \phi^n)|^2$$



Raissi, Maziar, Hessam Babaee, and Peyman Givi. "Deep learning of turbulent scalar mixing." *Physical Review Fluids* 4.12 (2019): 124501.

Model Discovery Problems

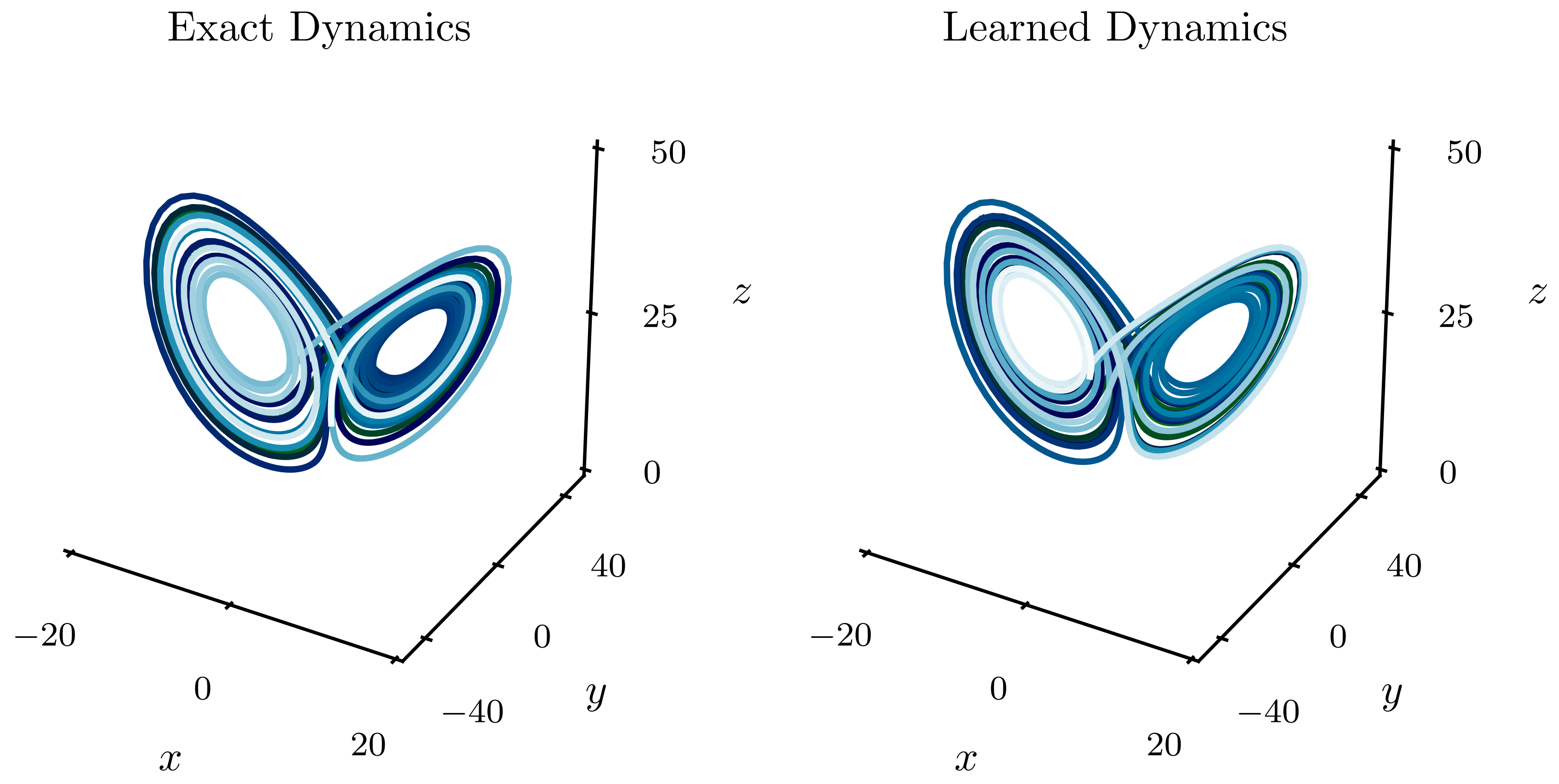


Multi-step Neural Networks

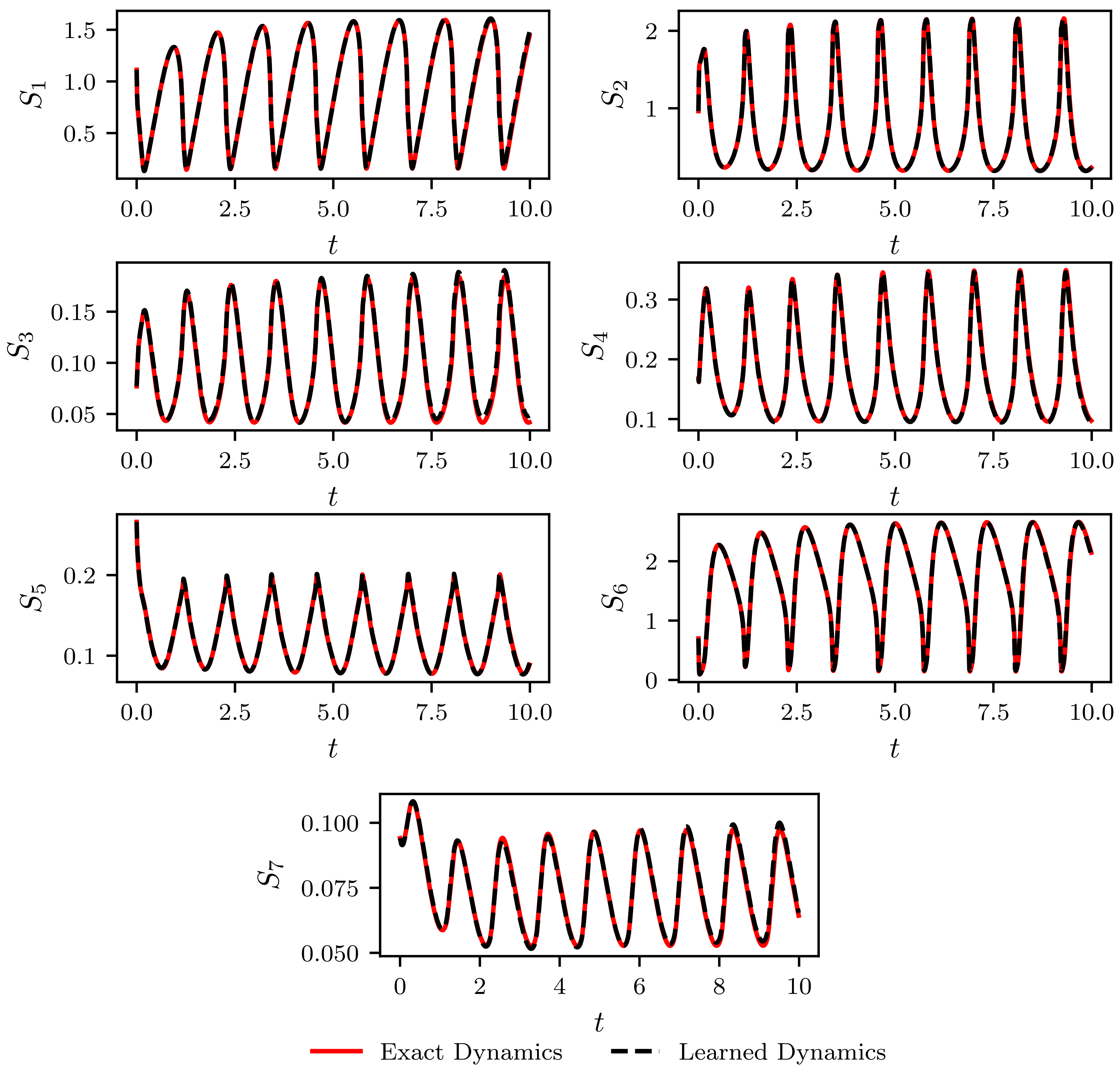
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

$$\mathbf{x}^{n+1} = \mathbf{x}^{n} + \frac{\Delta t}{2}\left(\mathbf{f}(\mathbf{x}^{n}) + \mathbf{f}(\mathbf{x}^{n+1})\right)$$

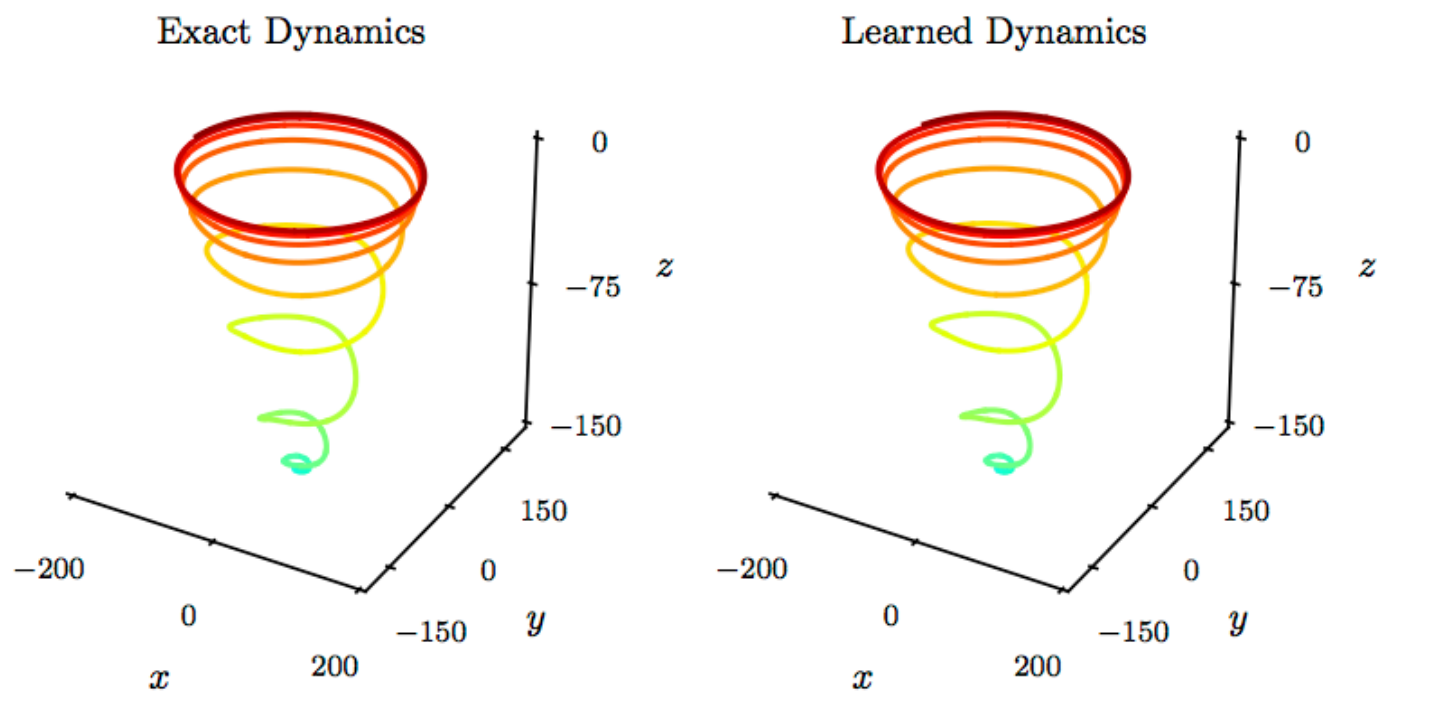
Lorenz System



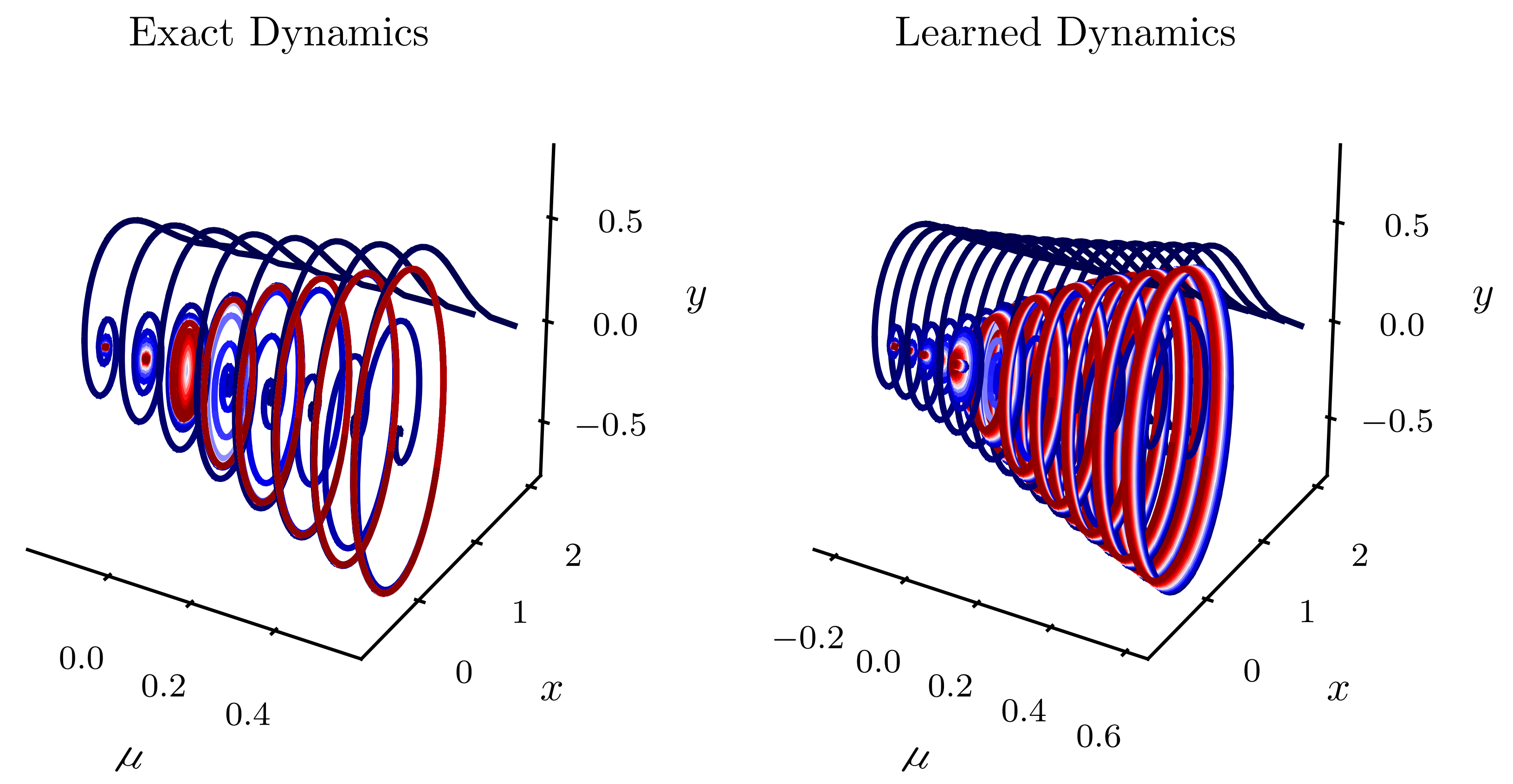
Glycolytic Oscillator



Navier-Stokes Equations



Hopf Bifurcation



Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Multistep neural networks for data-driven discovery of nonlinear dynamical systems." *arXiv preprint arXiv:1801.01236* (2018).

Deep Hidden Physics Models

$$u_t = \mathcal{N}(t, x, u, u_x, u_{xx}, \ldots) \qquad f = u_t - \mathcal{N}(t, x, u, u_x, u_{xx}, \ldots) \qquad \sum_{i=1}^{N} |u(t_i, x_i) - u_i|^2 + |f(t_i, x_i)|^2$$

Maziar Raissi (2018). Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations. *Journal of Machine Learning Research*, 19, 1-24.

Boulder

Hidden Physics Models

Thank you!

DARPA EQUiPS N66001-15-2-4055

MURI/ARO W911NF-15-1-0562

AFOSR FA9550-17-1-001

NIH U01HL116323

NSF DMS-1736088

644 followers

https://github.com/maziarraissi

NumericalGP
Numerical Gaussian Processes for Time-dependent and Non-linear Partial Differential Equations
MATLAB 40 31

HPM
Hidden physics models: Machine learning of nonlinear partial differential equations
MATLAB 77 54

ParametricGP
Parametric Gaussian Process Regression for Big Data
Python 35 16

DeepHPMs
Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations
Python 175 126

PINNs
Physics Informed Deep Learning: Data-driven Solutions and Discovery of Nonlinear Partial Differential Equations
Python 813 448

FBSNNs
Forward-Backward Stochastic Neural Networks: Deep Learning of High-dimensional Partial Differential Equations
Python 76 50