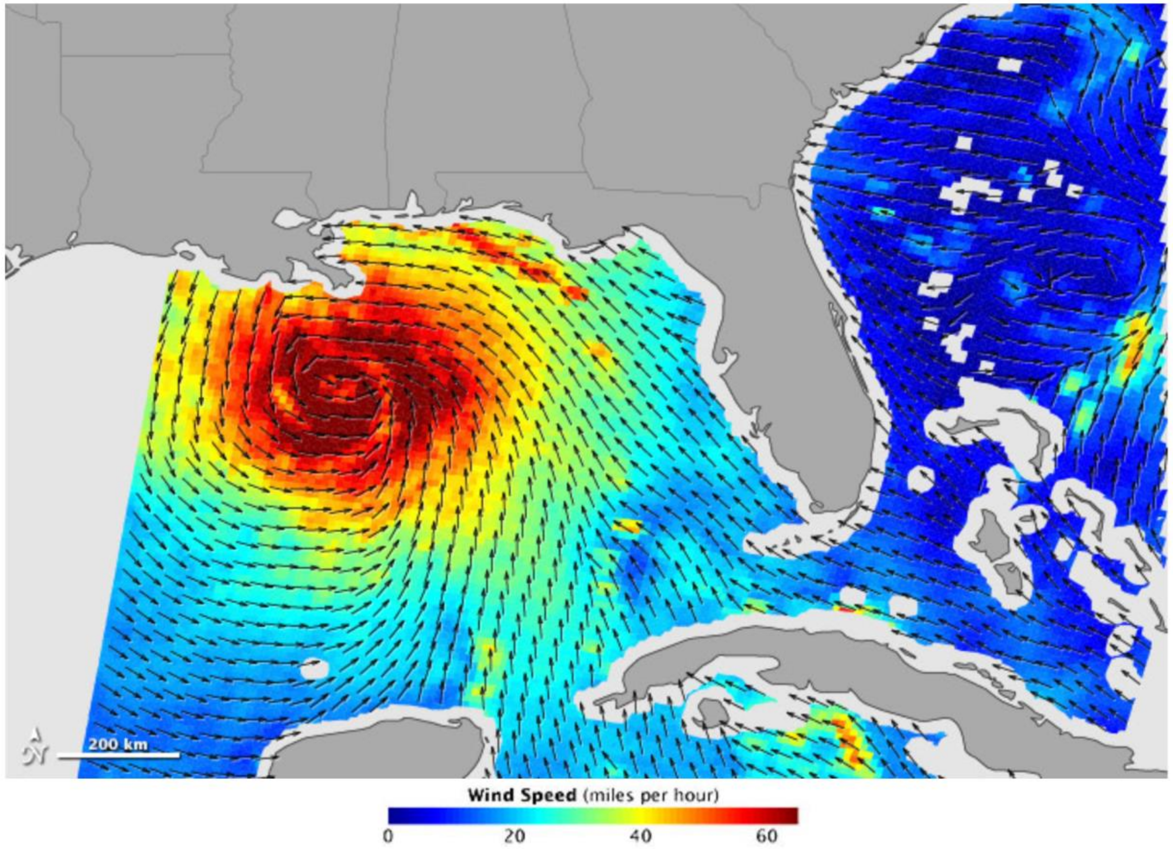
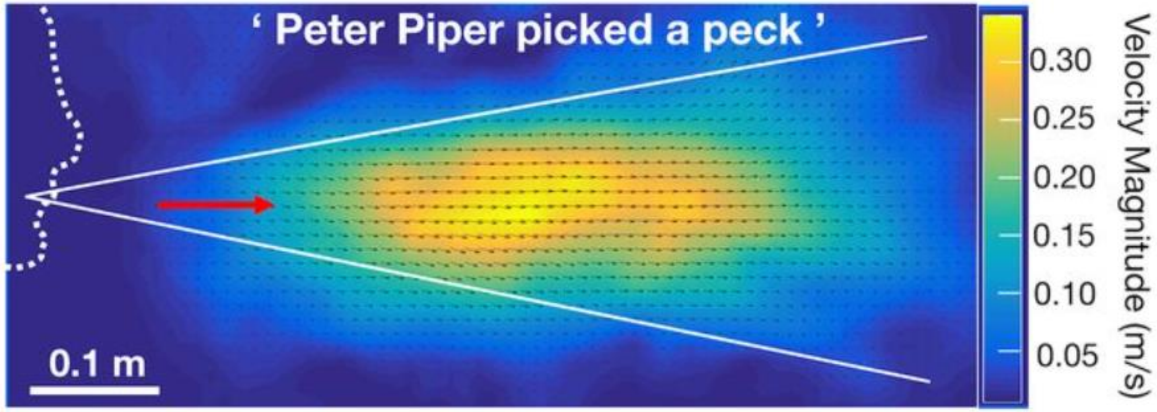
Hidden Fluid Mechanics

Can we infer non-constant “fields”? $q(x, y, t)$

velocity fields

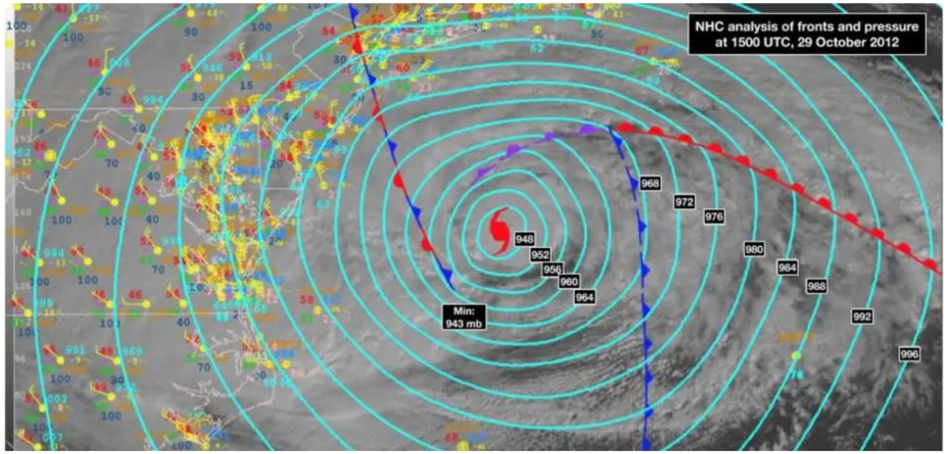


velocity fields



Abkarian et al. (2020)

pressure fields



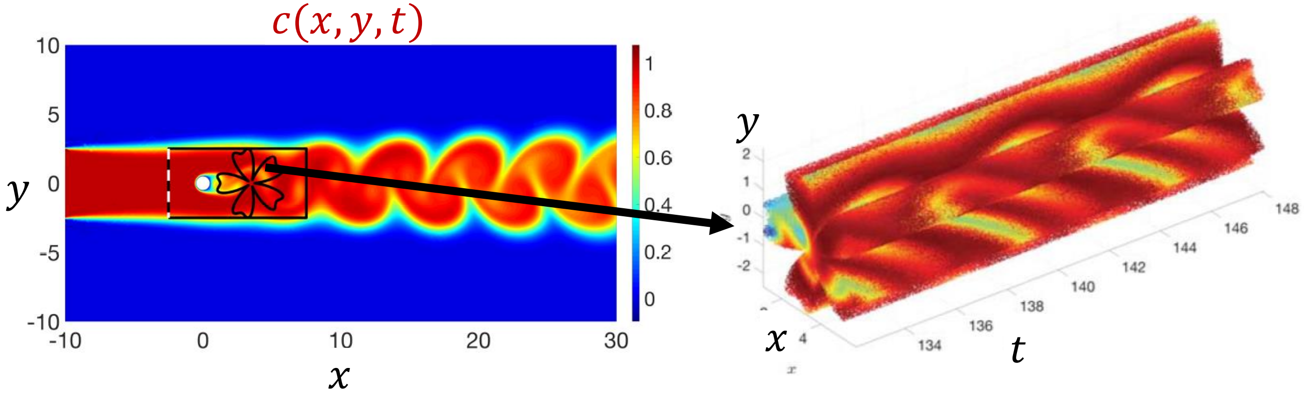
Can we infer non-constant “fields”? $q(x, y, t)$

If we don’t have u, v, and all we have is dye visualization of the flow, can it be used to infer velocity and pressure fields?

Flow around a cylinder

Given training data of $c(x, y, t)$
find $u(x, y, t), v(x, y, t), p(x, y, t)$

Dye concentration (ground truth) **Training data** $c(x, y, t)$



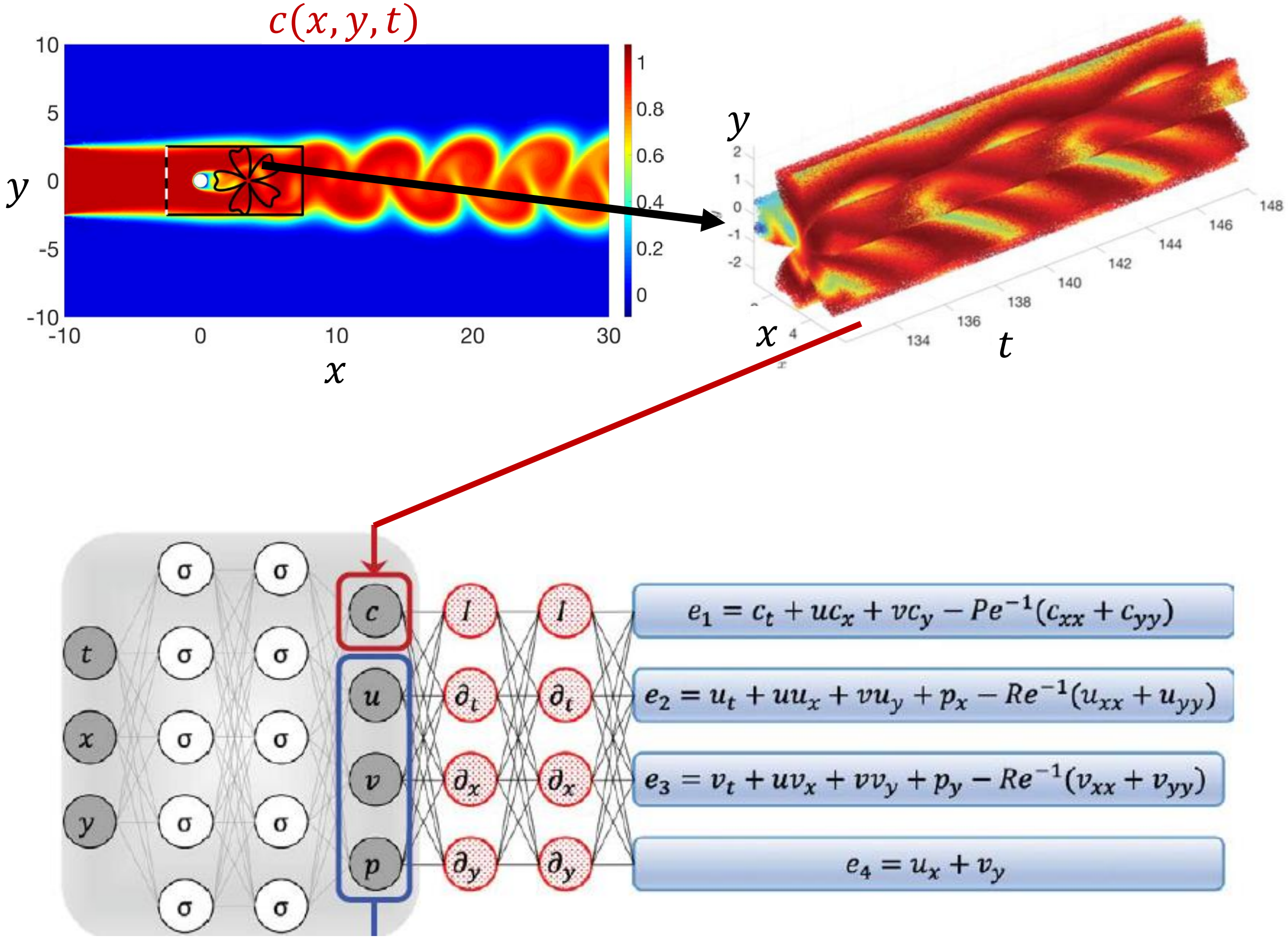
NN input: x, y, t
NN output: c, u, v, p
NN architecture: 10 layers 50 neurons per layer

$$c_t + uc_x + vc_y = Pe^{-1}(c_{xx} + c_{yy})$$
$$u_t + uu_x + vu_y = -p_x + Re^{-1}(u_{xx} + u_{yy})$$
$$v_t + uv_x + vv_y = -p_x + Re^{-1}(v_{xx} + v_{yy})$$
$$u_x + v_y = 0$$

c: the dye concentration
u, v: velocities
p: pressure
Re: Reynolds number (inertia/viscous forces)
Pe: Péclet number (rate of advection/rate of diffusion)

Flow around a cylinder

Dye concentration (ground truth) **Training data** $c(x,y,t)$



Given training data of $c(x,y,t)$ find $u(x,y,t), v(x,y,t), p(x,y,t)$

NN input: x, y, t
NN output: c, u, v, p
NN architecture: 10 layers 50 neurons per layer

Training data (from ground truth):

$$\{t^n, x^n, y^n, c^n\}_{n=1}^{N}$$

Collocation points:

$$\{t^m, x^m, y^m\}_{m=1}^{M}$$

Loss function:

Data points

$$MSE = \frac{1}{N}\sum_{n=1}^{N}\left|c(t^n, x^n, y^n, z^n) - c^n\right|^2$$ **Data loss**

$$+\sum_{i=1}^{5}\frac{1}{M}\sum_{m=1}^{M}\left|e_i(t^m, x^m, y^m, z^m)\right|^2$$ **Equation loss**

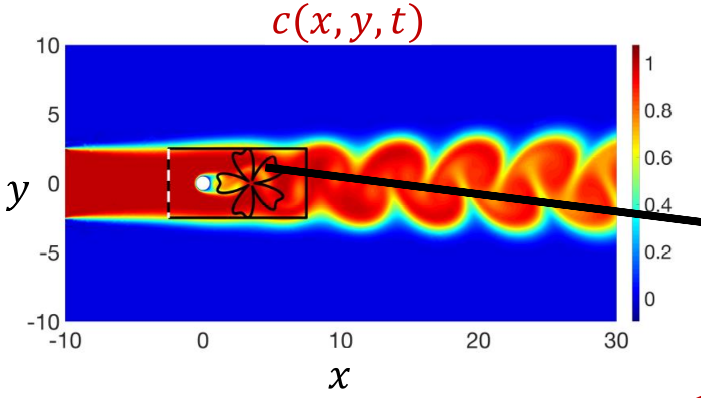
Collocation points

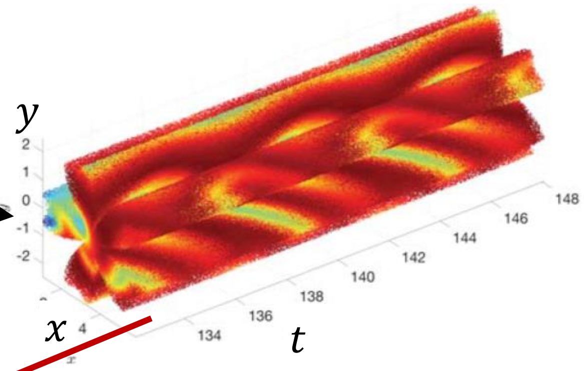
Raissi et al, Science (2020)

Good prediction!

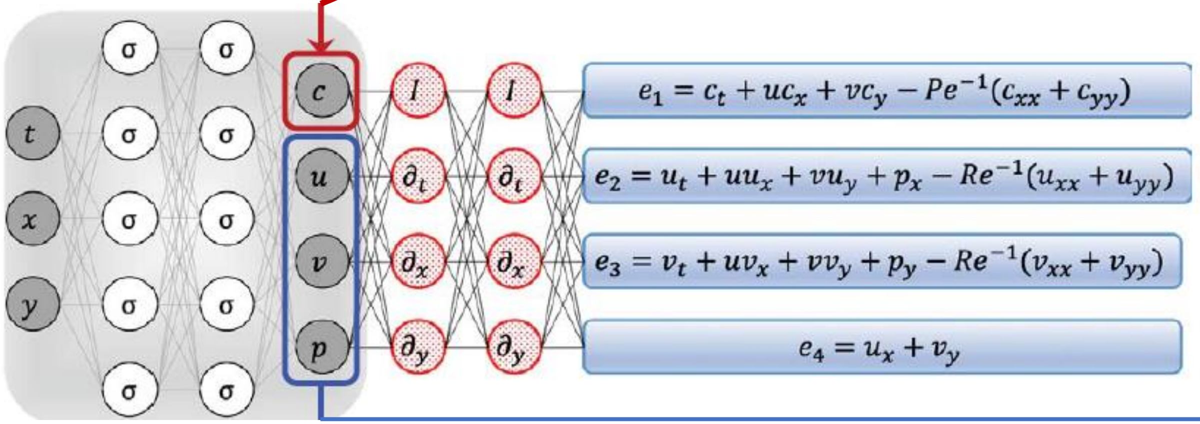
Given training data of $c(x,y,t)$
find $u(x,y,t), v(x,y,t), p(x,y,t)$

Dye concentration (ground truth) **Training data** $c(x,y,t)$

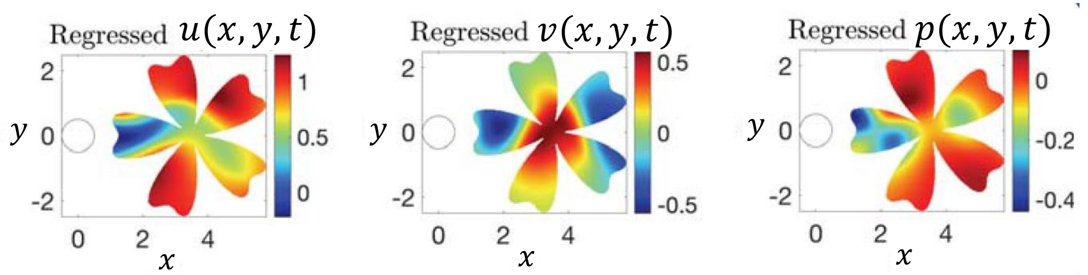




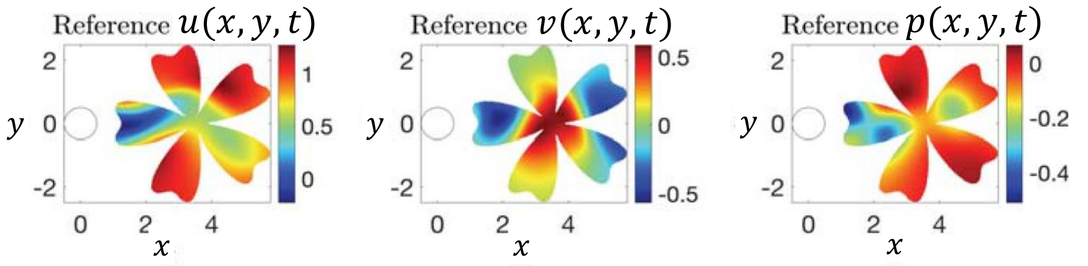
NN input: x, y, t
NN output: c, u, v, p
NN architecture: 10 layers 50 neurons per layer



NN outputs (prediction)



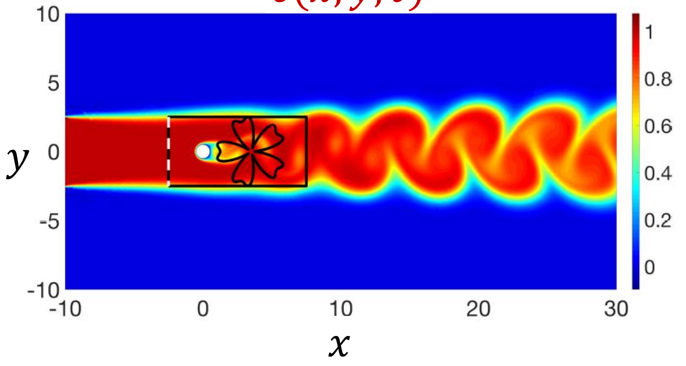
Ground truth (not used for training)



Sparse data?

Dye concentration (ground truth)

$c(x,y,t)$



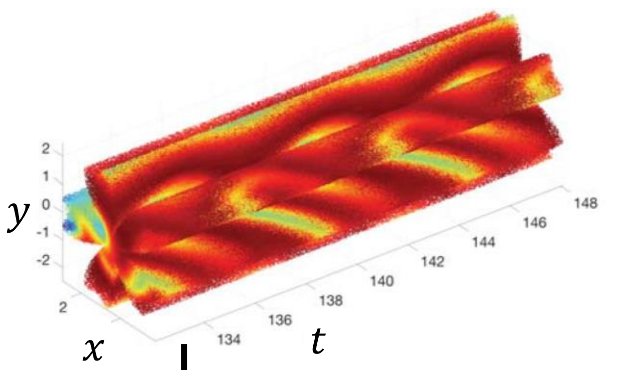
f: NN predicted function
g: ground truth function

Relative L_2 error:

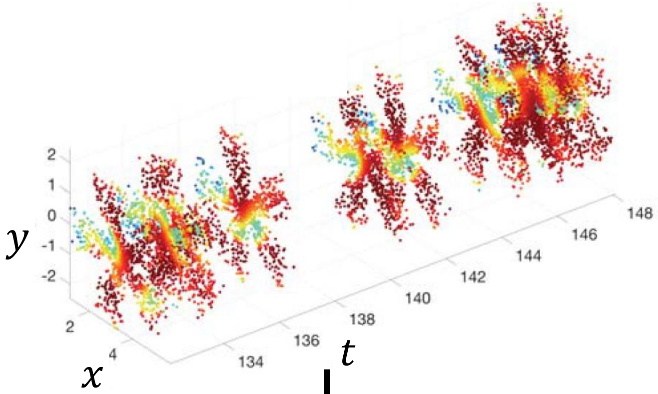
$$\mathcal{E}(f,g) := \left(\frac{1}{N}\sum_{i=1}^{N}[f(x_i)-g(x_i)]^2\right) \Big/ \left(\frac{1}{N}\sum_{i=1}^{N}\left[g(x_i)-\frac{1}{N}\sum_{i=1}^{N}g(x_i)\right]^2\right)$$

It is invariant under **shift** and **scaling** of both the regressed f and the reference functions g; i.e., $\varepsilon(f+a, g+a) = \varepsilon(f,g)$ and $\varepsilon(bf, bg) = \varepsilon(f,g)$.

Dense training data $c(x,y,t)$

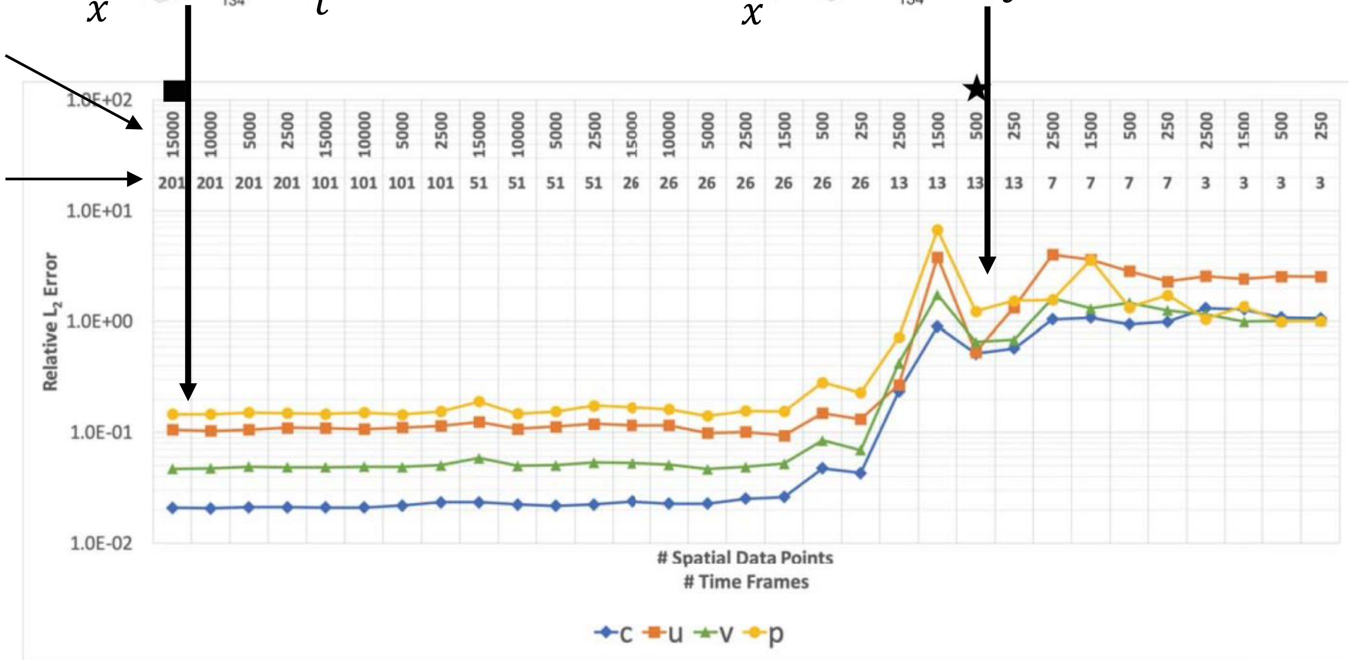


Sparse training data $c(x,y,t)$



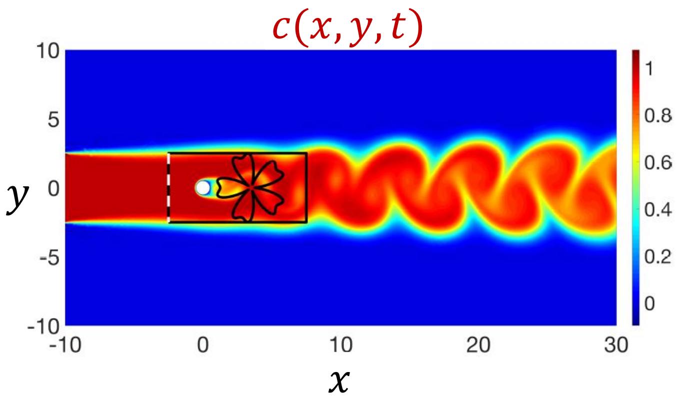
# of data point per time frame

# of time frames



Noisy data?

Dye concentration (ground truth)

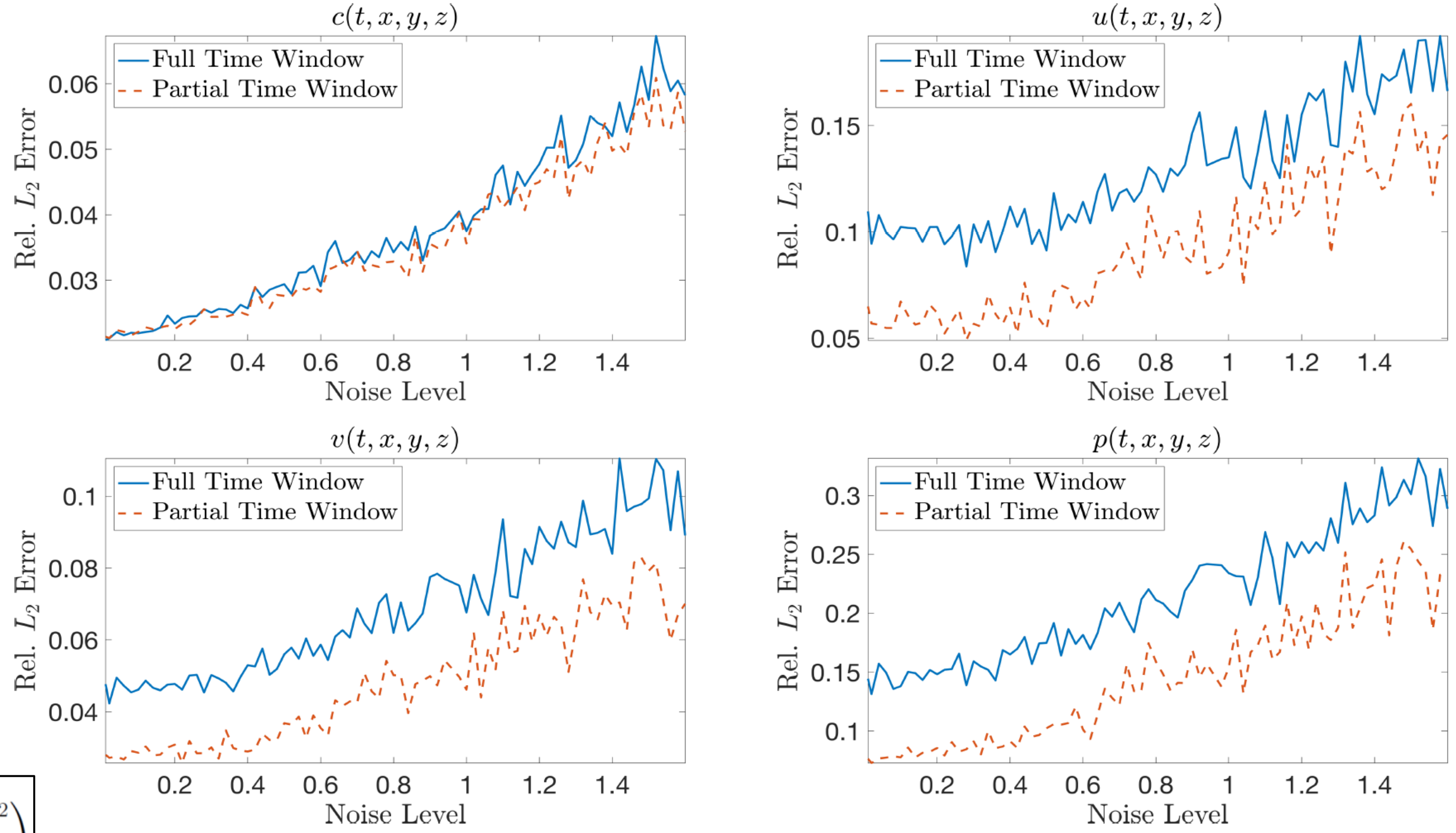


f: NN predicted function
g: ground truth function

Relative L_2 error:

$$\boxed{\mathcal{E}(f,g) := \left(\frac{1}{N}\sum_{i=1}^{N}[f(x_i)-g(x_i)]^2\right) \Big/ \left(\frac{1}{N}\sum_{i=1}^{N}\left[g(x_i)-\frac{1}{N}\sum_{i=1}^{N}g(x_i)\right]^2\right)}$$

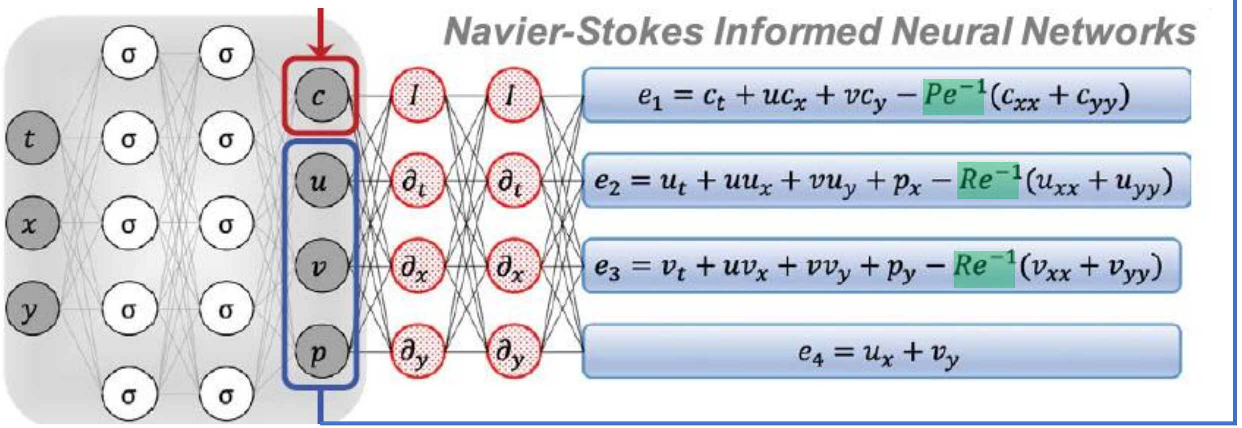
It is invariant under **shift** and **scaling** of both the regressed f and the reference functions g; i.e., $\varepsilon(f+a, g+a) = \varepsilon(f,g)$ and $\varepsilon(bf, bg) = \varepsilon(f,g)$.



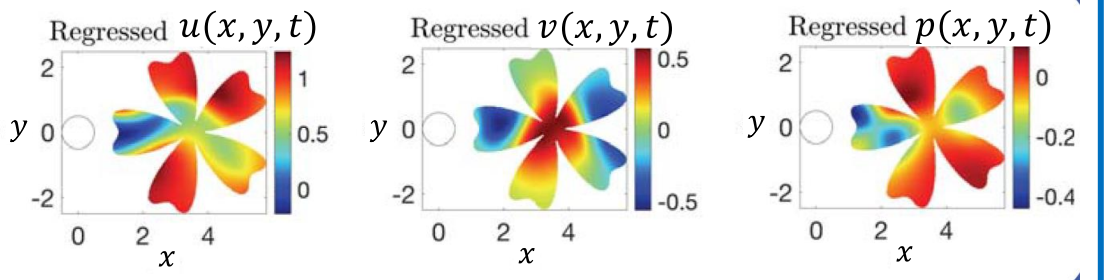
Infer Re, Pe

In addition to the velocity $u(x,y,t), v(x,y,t)$ and pressure fields $p(x,y,t)$, it is possible to discover other unknown parameters of the flow field such as the Re, Pe, based solely on observations of dye visualization $\boldsymbol{c(x,y,t)}$

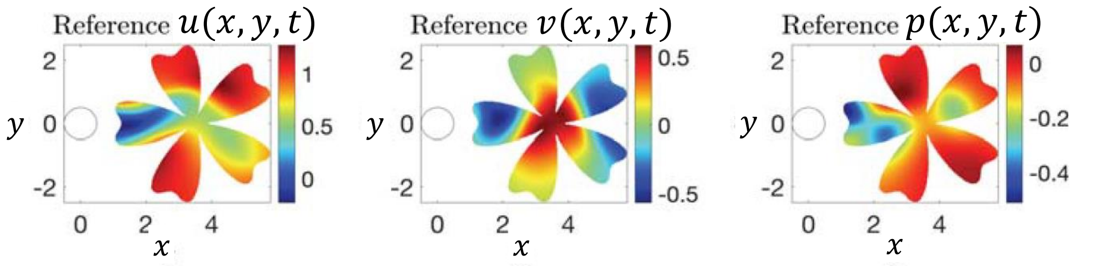
		10^6 iterations of training	
	Reference	Inferred	Rel. Error
Pe	100	93.41	6.59%
Re	100	93.16	6.84%



NN outputs (prediction)



Ground truth (not used for training)

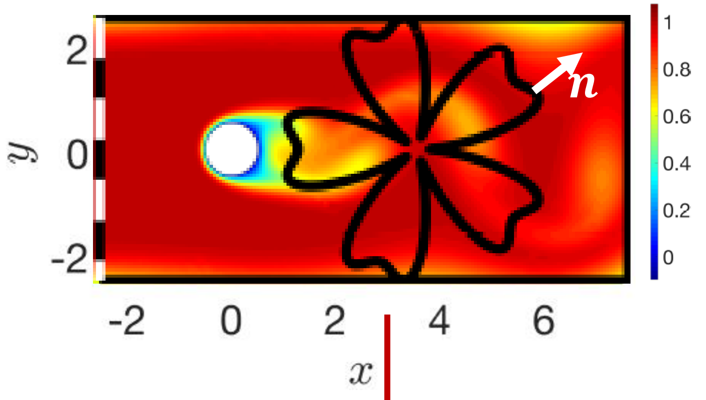


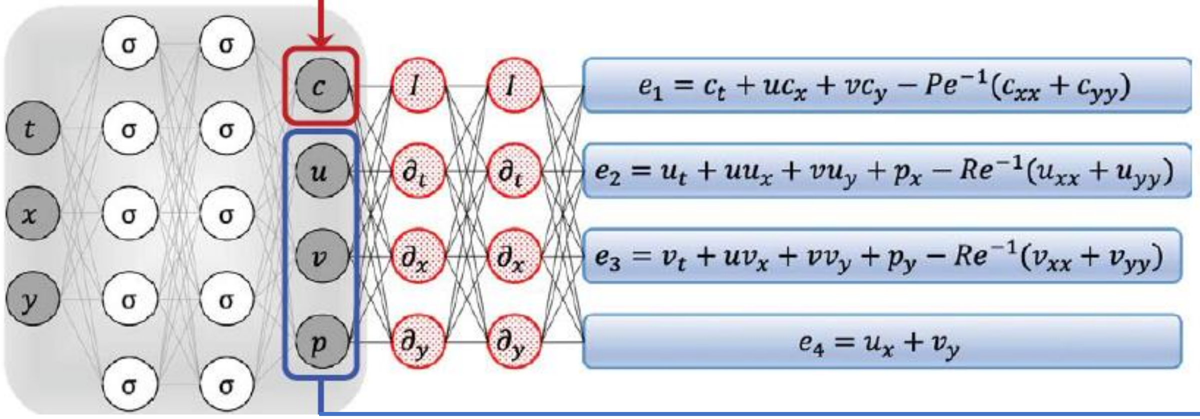
Can the training domain be selected anywhere?

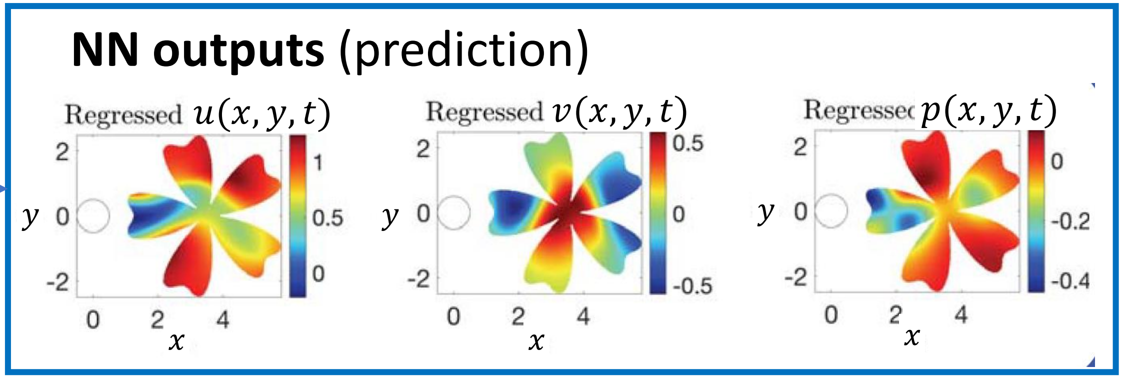
What would happen to u, v inversion, if $c_x,\ c_y = 0$ in most of the domain (the flower shape)?

Hint: The first eqn (e_1) connects $c(x, y, t)$ with velocities.

Training data $c(x, y, t)$





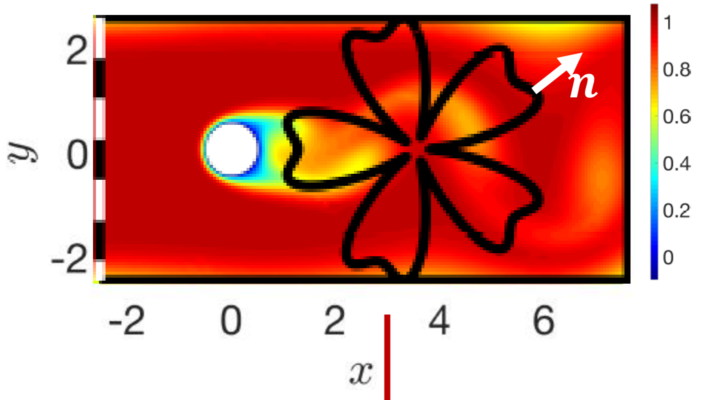


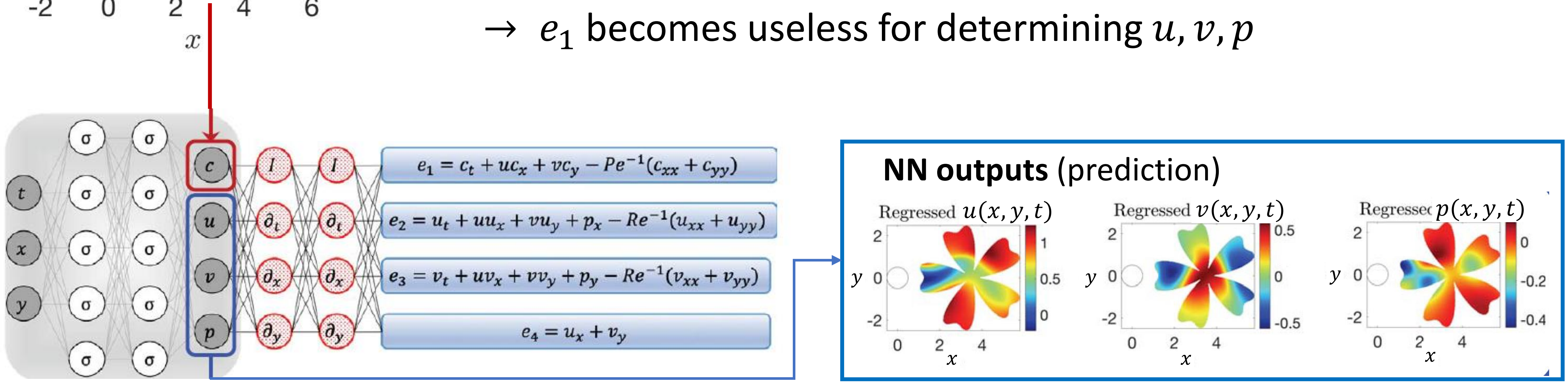
Can the training domain be selected anywhere?

What would happen to u, v inversion, if $c_x,\ c_y = 0$ in most of the domain (the flower shape)?

The first eqn (e_1) connects $c(x, y, t)$ with velocities.
If $c_x,\ c_y = 0$ in the domain
→ The advection terms in the first equation vanish
→ e_1 becomes useless for determining u, v, p

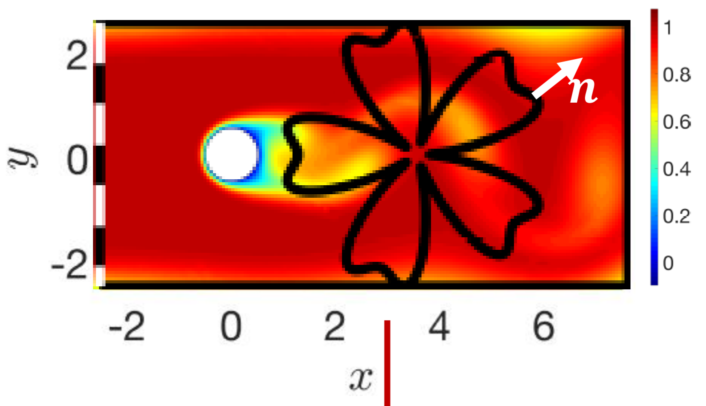
Training data $c(x, y, t)$



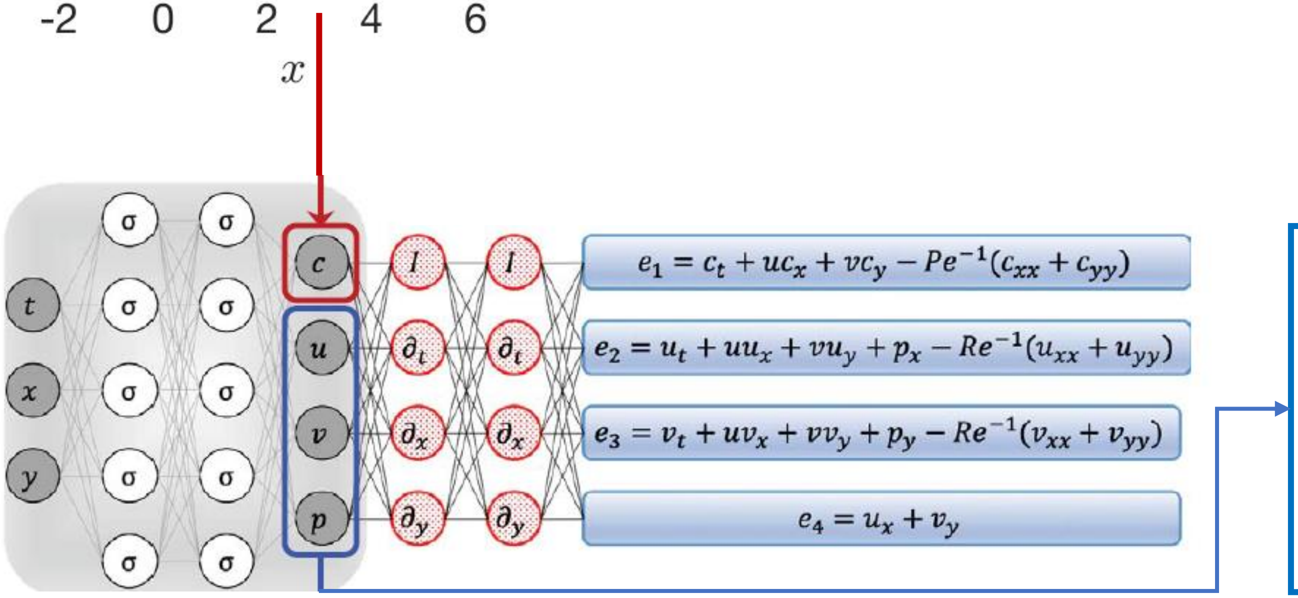


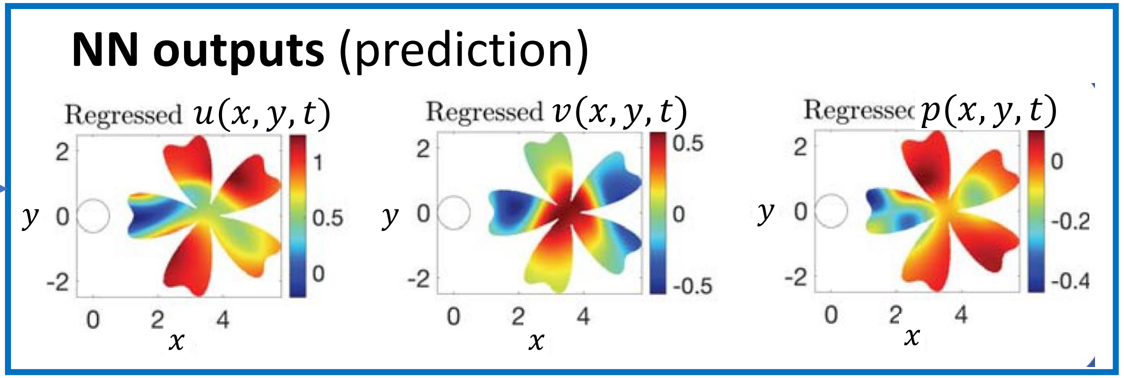
Can the training domain be selected anywhere?

Training data $c(x, y, t)$



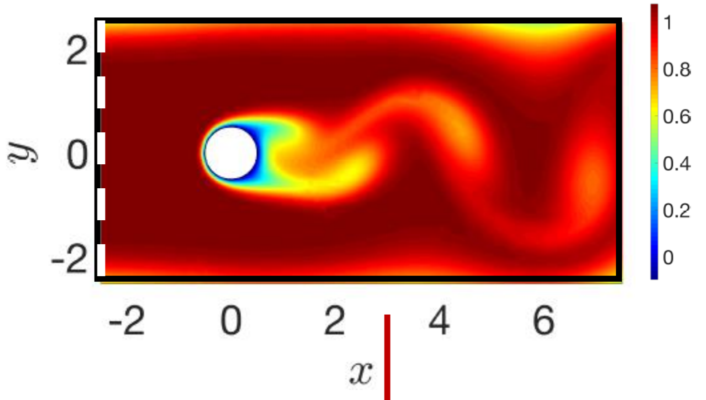
There must be sufficient gradients of concentration ($c_x,\ c_y \neq 0$) in order for the method to be able to infer a single solution $u(x, y, t), v(x, y, t)$



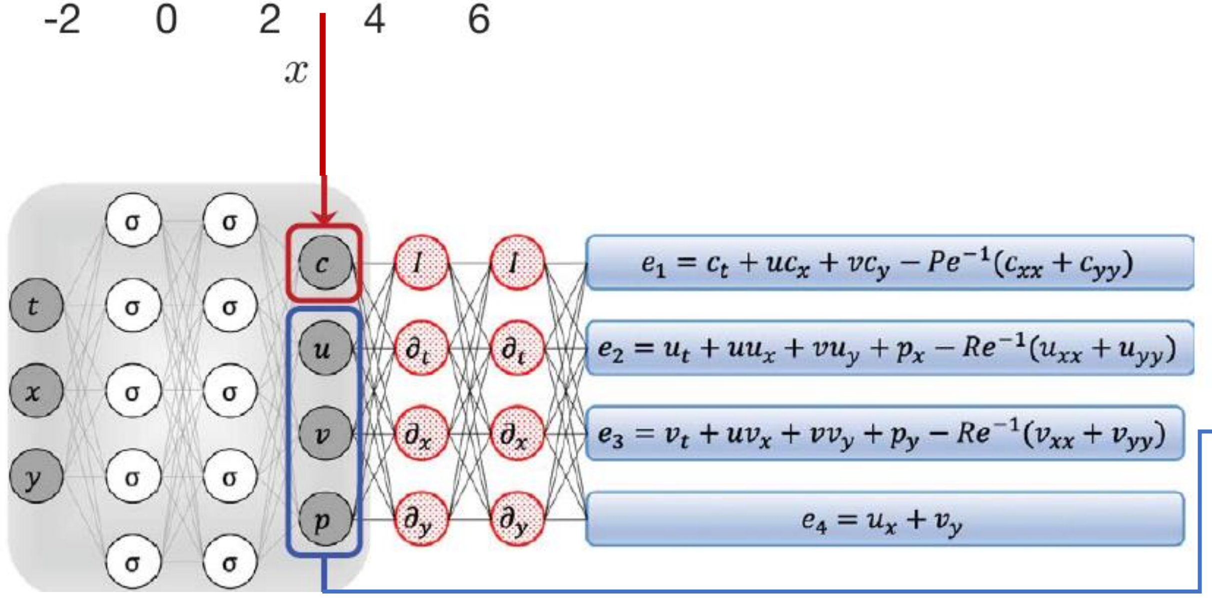


Insufficient c gradients

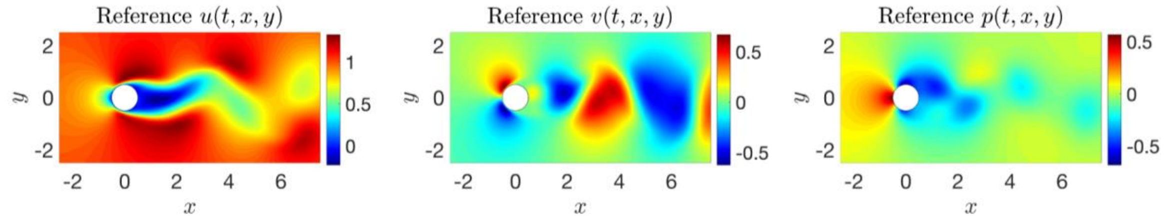
Training data $c(x, y, t)$



What can we do to help invert for u, v if unfortunately, in most of the domain $c_x,\ c_y = 0$?

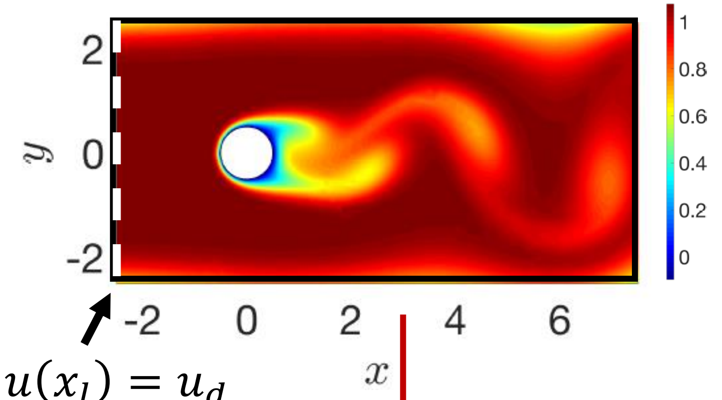


Ground truth (not used for training)



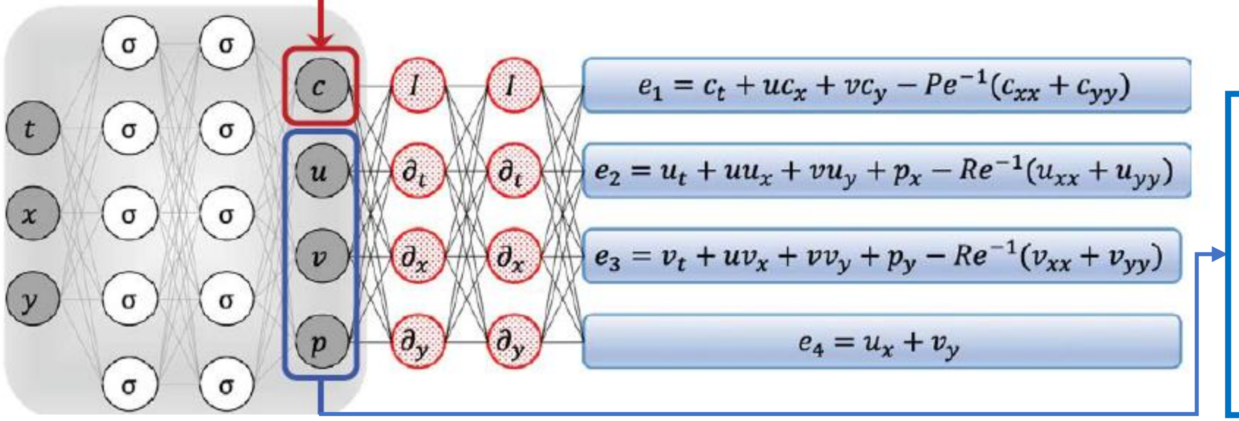
Insufficient c gradients

Training data $c(x, y, t)$

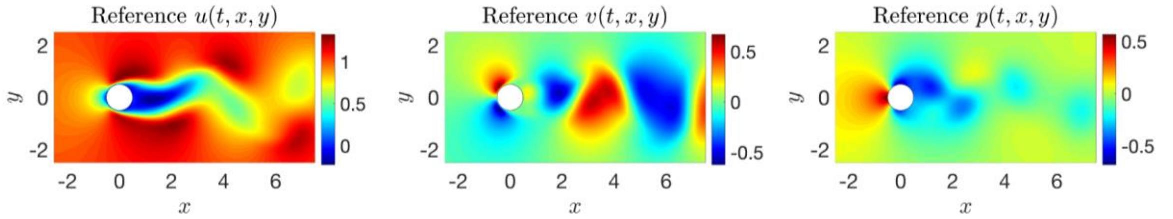


$u(x_l) = u_d$
$v(x_l) = v_d$

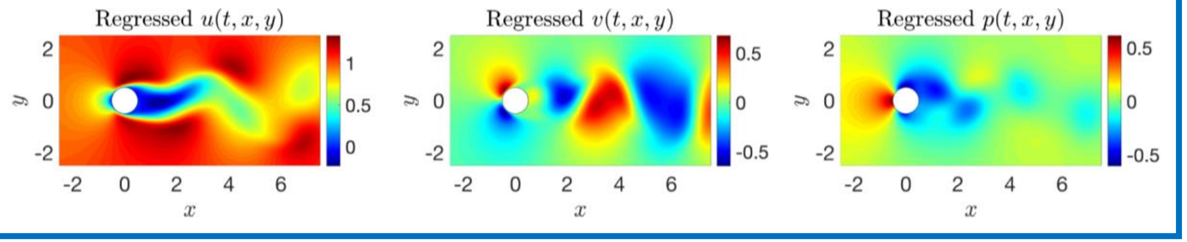
We could impose boundary conditions (extra info) at the left boundary $u(x_l) = u_d,\ v(x_l) = v_d$ to guide the prediction



Ground truth (not used for training)

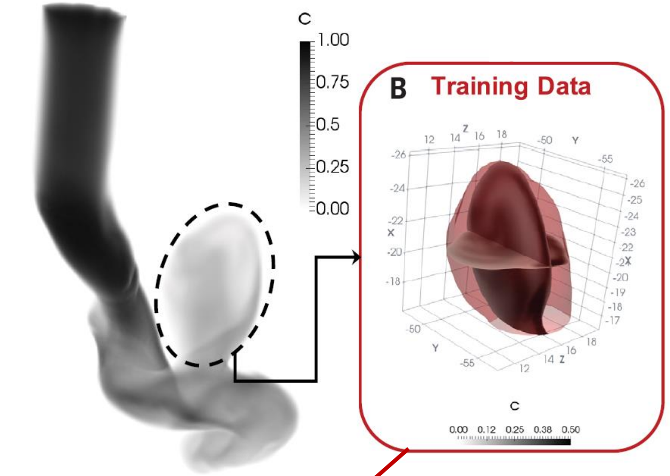


NN outputs (prediction)



3D blood flow

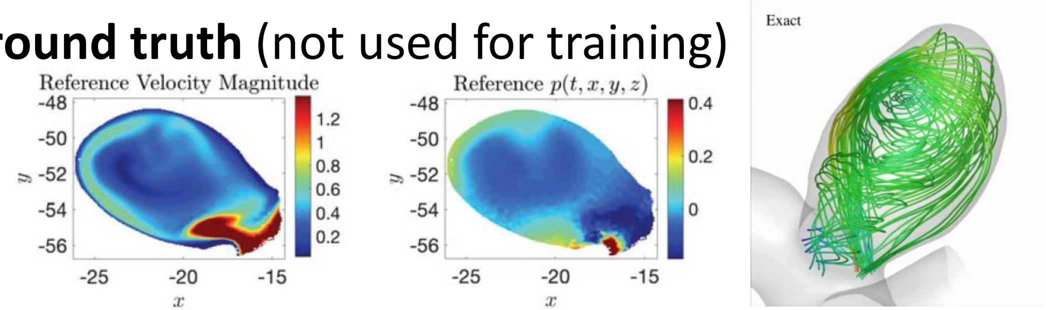
Training data $c(x, y, t)$



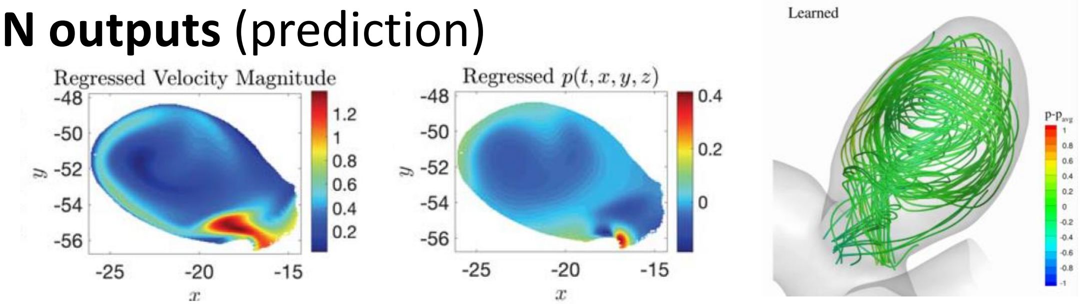
Given training data of $c(x, y, t)$
find $u(x, y, t), v(x, y, t), p(x, y, t)$

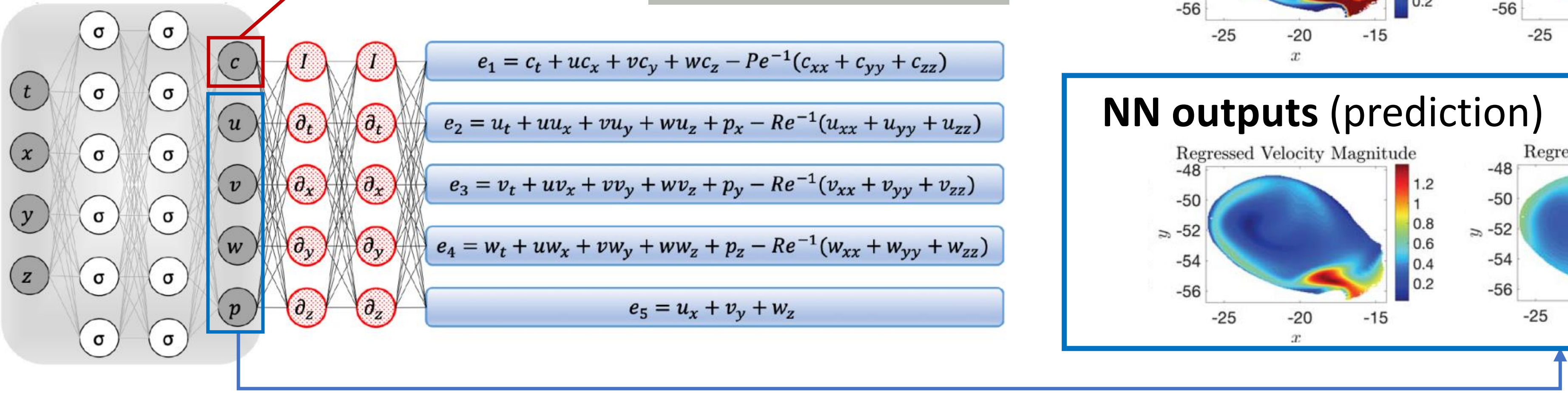
NN input: x, y, x, t
NN output: c, u, v, w, p
NN architecture: 10 layers 50 neurons per layer

Ground truth (not used for training)



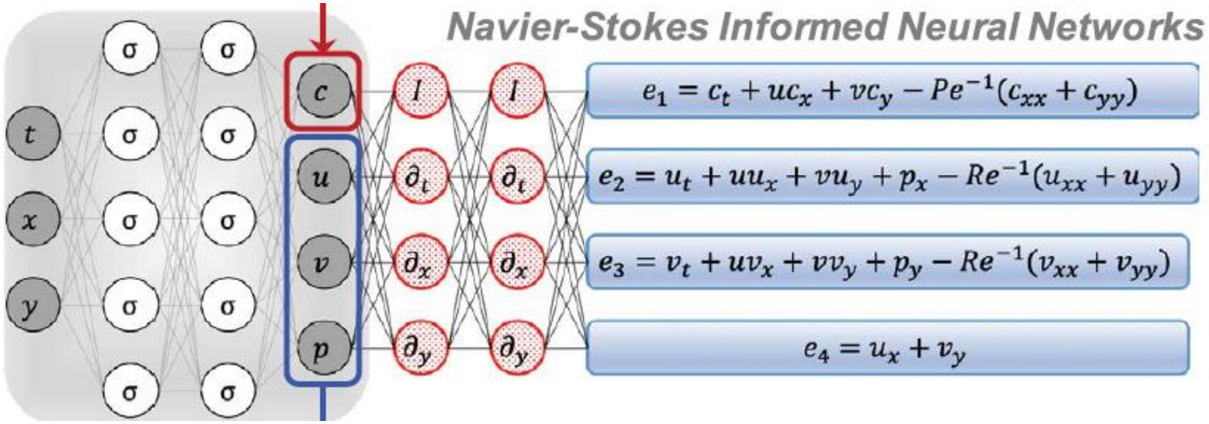
NN outputs (prediction)



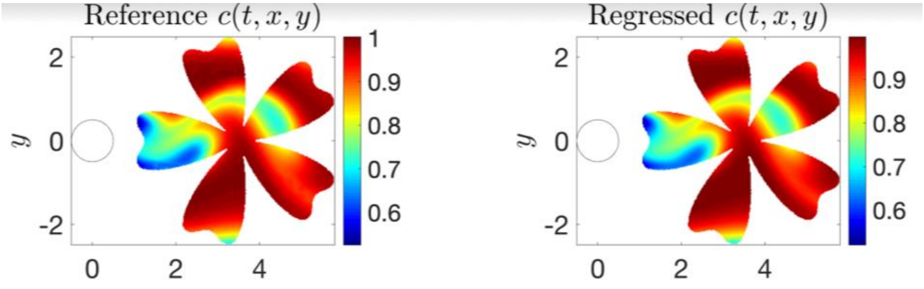


Cylinder arbitrary domain coding exercise

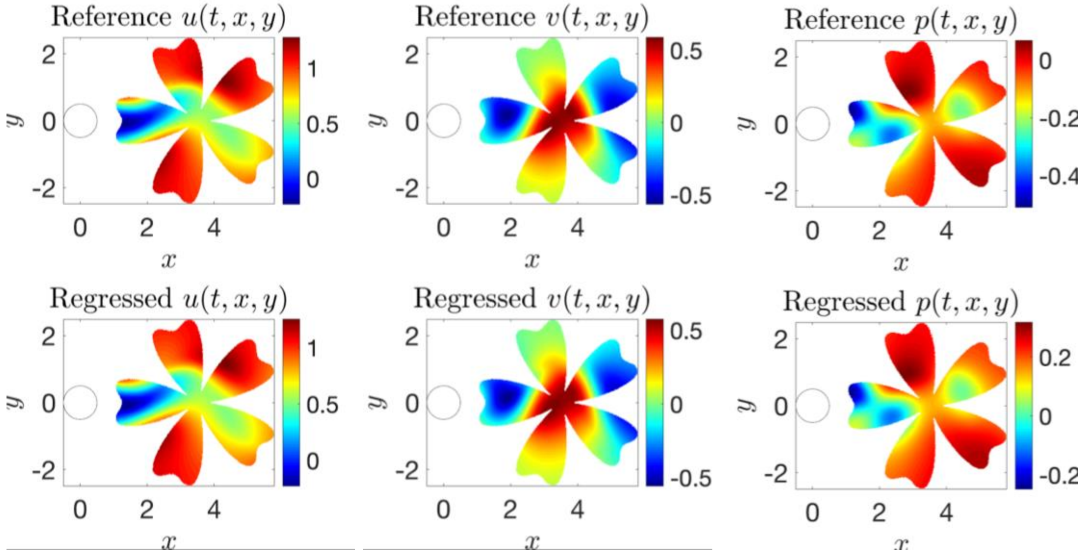
- TF1.14



Training data



Prediction



Is $c(x, y, t)$ sufficient to result in a unique velocity and pressure fields $u(x, y, t), v(x, y, t), p(x, y, t)$?

- Normally there are **no guarantees for unique solutions** unless **proper boundary conditions** are explicitly imposed on the domain boundaries **(well posed).**

- However, as shown in the paper, an informed selection of the training boundaries in the regions where there are sufficient gradients in $c(x,y,t)$ could possibly eliminate the requirement of imposing velocity and pressure boundary conditions.

PINN

Utilize less data

Utilize more physics

- Application 1: Prediction of solution for a **well-posed problem**
- Application 2: Prediction of solution when data is available within the domain but not at the IC, BC
- Application 3: Data-driven discovery of **unknown constants**
- Application 4: Data-driven discovery of **unknown parameter fields**

PINN gives a good prediction when the training loss is sufficiently low and is close to the testing loss evaluated using different sets of collocation points and test data.

The point of PINN is that its prediction can be generalized to a domain without observatoins!

Open questions

- How deep/wide should the neural network be?
- How much data is really needed?
- Can we improve on initializing the network weights or normalizing the data?
- Are the mean square error and the sum of squared errors the appropriate loss functions?
- Why are these methods seemingly so robust to noise in the data?
- What types of problems can easily trap the model training parameters in local minima?