Solving PDE related problems using deep-learning

Adar Kahana
Tel Aviv University

Under the supervision of
Eli Turkel (TAU) and
Dan Givoli (Technion)
Shai Dekel (TAU)

Waves seminar,
UC Merced

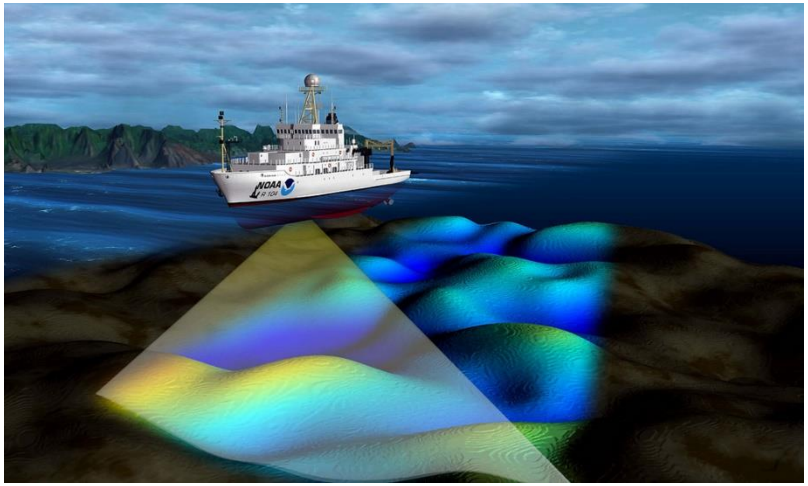
February 4th , 2021

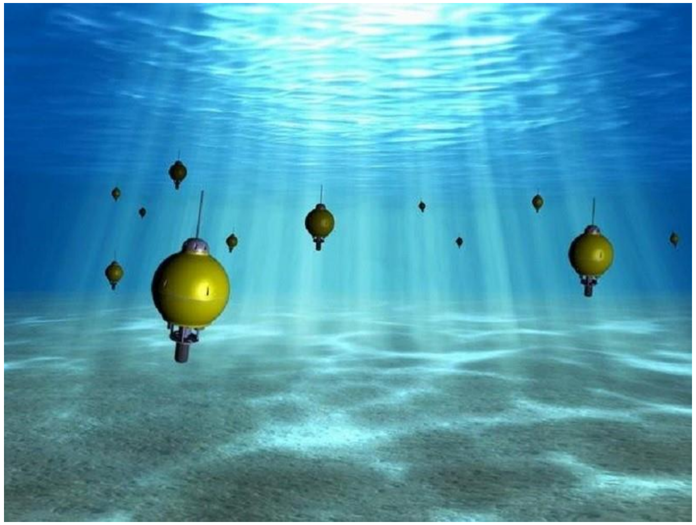
Agenda

- Motivation
- Data driven problems
- Obstacle identification and deep-learning
- Dealing with CFL instability using deep-learning

Underwater acoustics

- Sonar imaging





The wave problem

$$\begin{cases} \ddot{u}(\vec{x},t) = div\big(c(\vec{x})^2 \nabla u(\vec{x},t)\big), & \vec{x} \in \Omega, \mathrm{t} \in (0,T] \\ u(\vec{x},0) = u_0(\vec{x}), & \vec{x} \in \Omega \\ \dot{u}(\vec{x},0) = \dot{u}_0(\vec{x}), & \vec{x} \in \Omega \\ u(\vec{x},t) = f(\vec{x},t), & \vec{x} \in \partial\Omega_1, \mathrm{t} \in [0,T] \\ \nabla u(\vec{x},t) = g(\vec{x},t), & \vec{x} \in \partial\Omega_2, \mathrm{t} \in [0,T] \end{cases}$$

$$\partial\Omega_1 \cup \partial\Omega_2 = \partial\Omega$$

where $\dot{u}_0(\vec{x}) = 0$ and $f(\vec{x},t) = g(\vec{x},t) = 0$

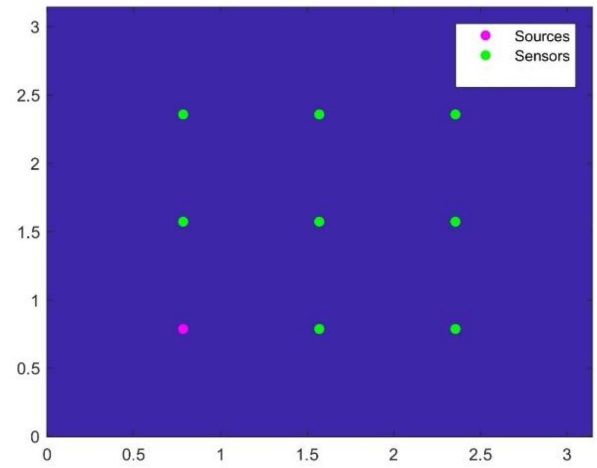
Ill-posed problems

- In an experiment, we store the pressure at a small number of sensors for all time steps
- We wish to find the properties of the source or obstacle from the data stored at these sensors where the **number of sensors << mesh**
- *This is an inverse problem which is highly ill-posed*
- Hence, one cannot usually reconstruct the initial conditions perfectly
- Can we solve these types of ill-posed problems with learning?

Partial information

- "Recording" the solution at a small set of sensors placed in the domain $\{\vec{x}_{s_n}\}_{n=1}^{K} \in \Omega$
- Data –

$$\begin{pmatrix} u(\vec{x}_{s_1}, \mathrm{t}) \\ u(\vec{x}_{s_2}, \mathrm{t}) \\ \vdots \\ u(\vec{x}_{s_n}, \mathrm{t}) \end{pmatrix} + \mathcal{N}(\mu, \sigma^2)$$



- The ill-posedness raises sensitivity to noise at the sensors

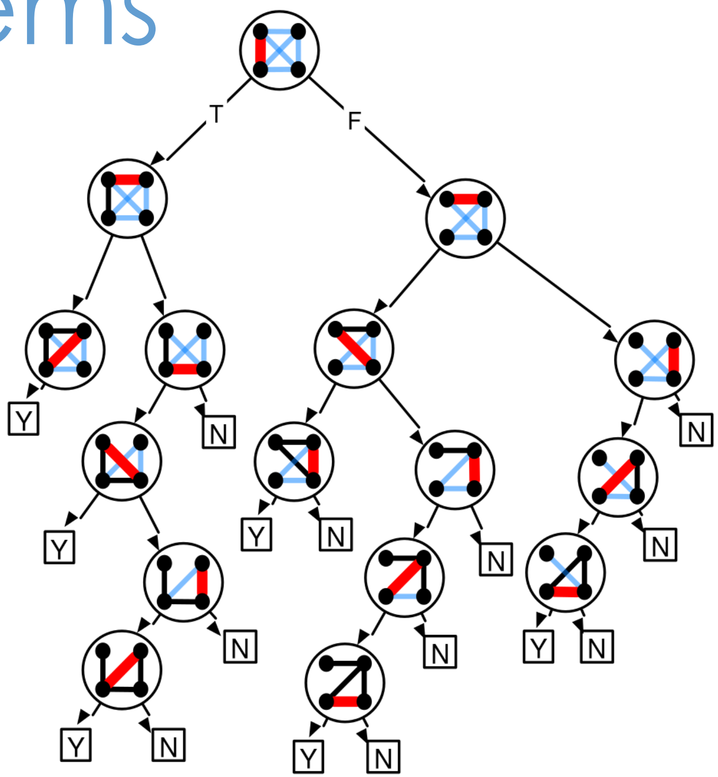
Agenda

- Motivation ✓
- Data driven problems
- Obstacle identification and deep-learning
- Dealing with CFL instability using deep-learning

Data driven problems

- Supervised learning
  - Input data
  - Output labels
- Training
- Prediction (testing)

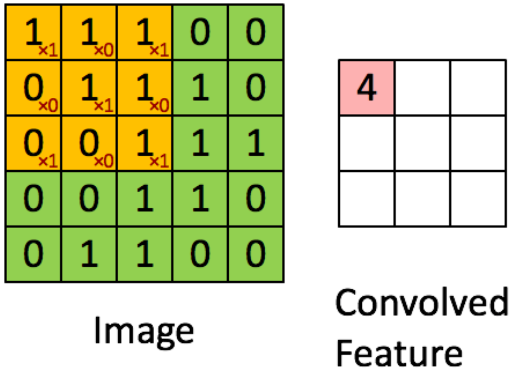
- Drawback - sensitivity

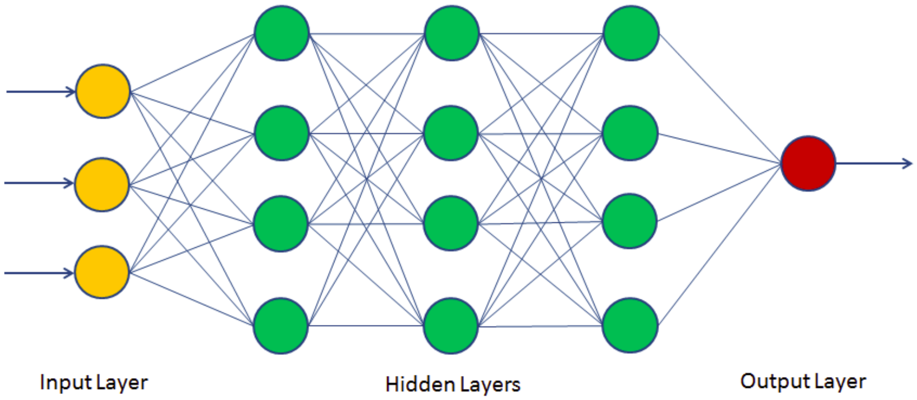


Deep-learning

- Training "weights" to learn connections in the data
- Hidden multi-dimensional embeddings
- Convolutions, Fully connected
- "Deep" and non-linear
- Loss

$$\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 1\end{pmatrix} \rightarrow \begin{pmatrix}w_1 & w_2 & w_3\\ w_4 & w_5 & w_6\\ w_7 & w_8 & w_9\end{pmatrix}$$





Physically-informed NN

- **Input**: set of points from the initial and boundary conditions
- **Output**: solution in the domain
- **Loss**: the problem

$$ih_t + 0.5h_{xx} + |h|^2 h = 0, \quad x \in [-5, 5], \quad t \in [0, \pi/2],$$
$$h(0, x) = 2\ \mathrm{sech}(x),$$
$$h(t, -5) = h(t, 5),$$
$$h_x(t, -5) = h_x(t, 5),$$

$$f := ih_t + 0.5h_{xx} + |h|^2 h,$$

$$MSE = MSE_0 + MSE_b + MSE_f,$$

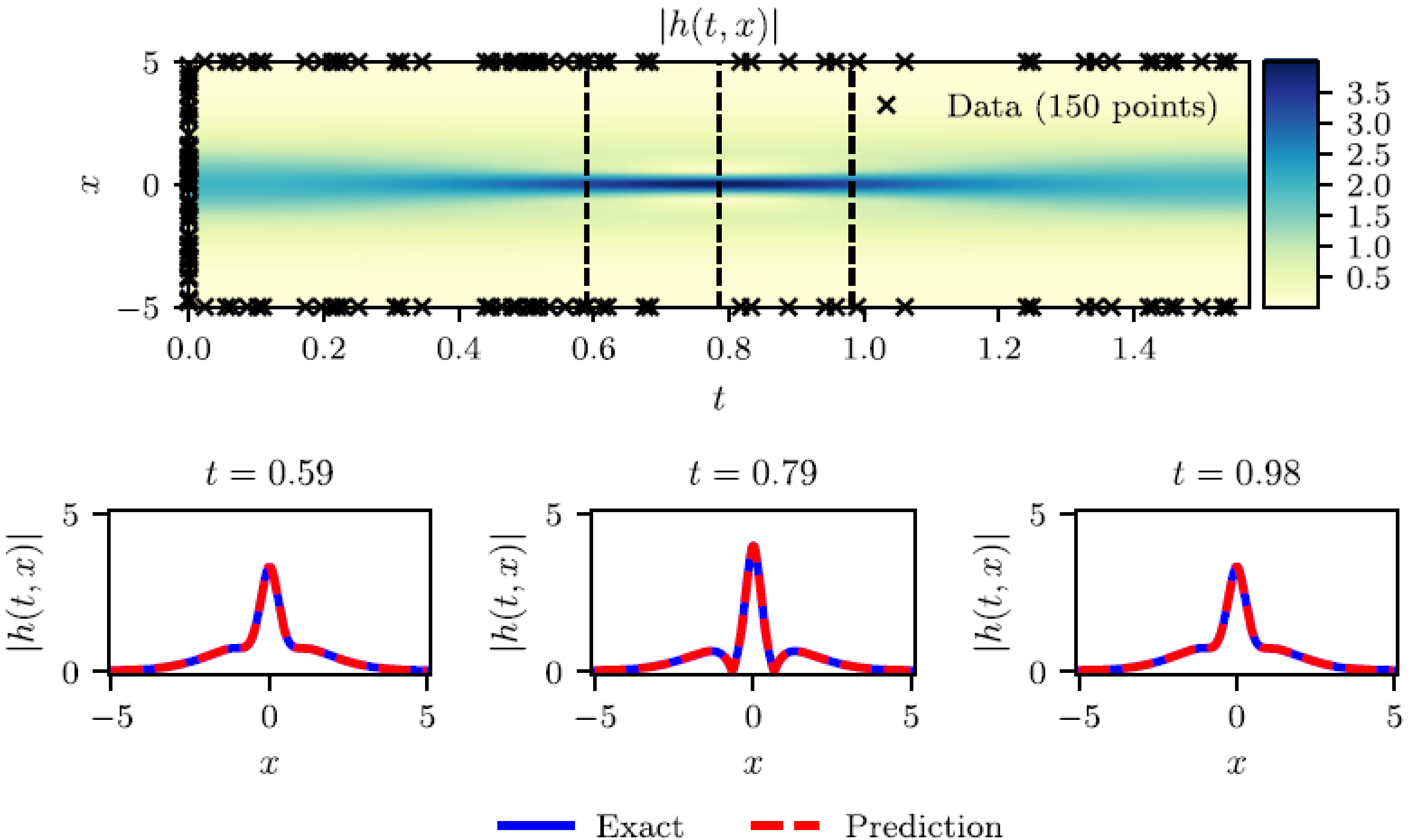
$$MSE_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} |h(0, x_0^i) - h_0^i|^2,$$

$$MSE_b = \frac{1}{N_b} \sum_{i=1}^{N_b} \left(|h^i(t_b^i, -5) - h^i(t_b^i, 5)|^2 + |h_x^i(t_b^i, -5) - h_x^i(t_b^i, 5)|^2 \right),$$

$$MSE_f = \frac{1}{N_f} \sum_{i=1}^{N_f} |f(t_f^i, x_f^i)|^2.$$

M. Raissi, P. Perdikaris, G.E. Karniadakis, Journal of computational Physics, 2018

Results

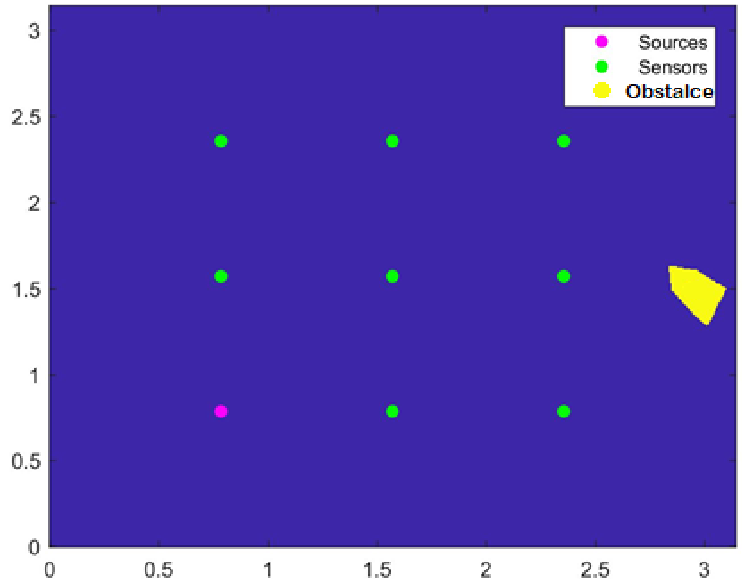

|h(t, x)|
Data (150 points)
x
t
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
5
0
−5
3.5
3.0
2.5
2.0
1.5
1.0
0.5
t = 0.59
t = 0.79
t = 0.98
|h(t, x)|
Exact
Prediction

Agenda

- Motivation ✓
- Data driven problems ✓
- Obstacle identification and deep-learning
- Dealing with CFL instability using deep-learning

Problem definition

Given the position of the source/s and data at a few sensors but many time slices **find the location, size and shape of the unknown scatterers**



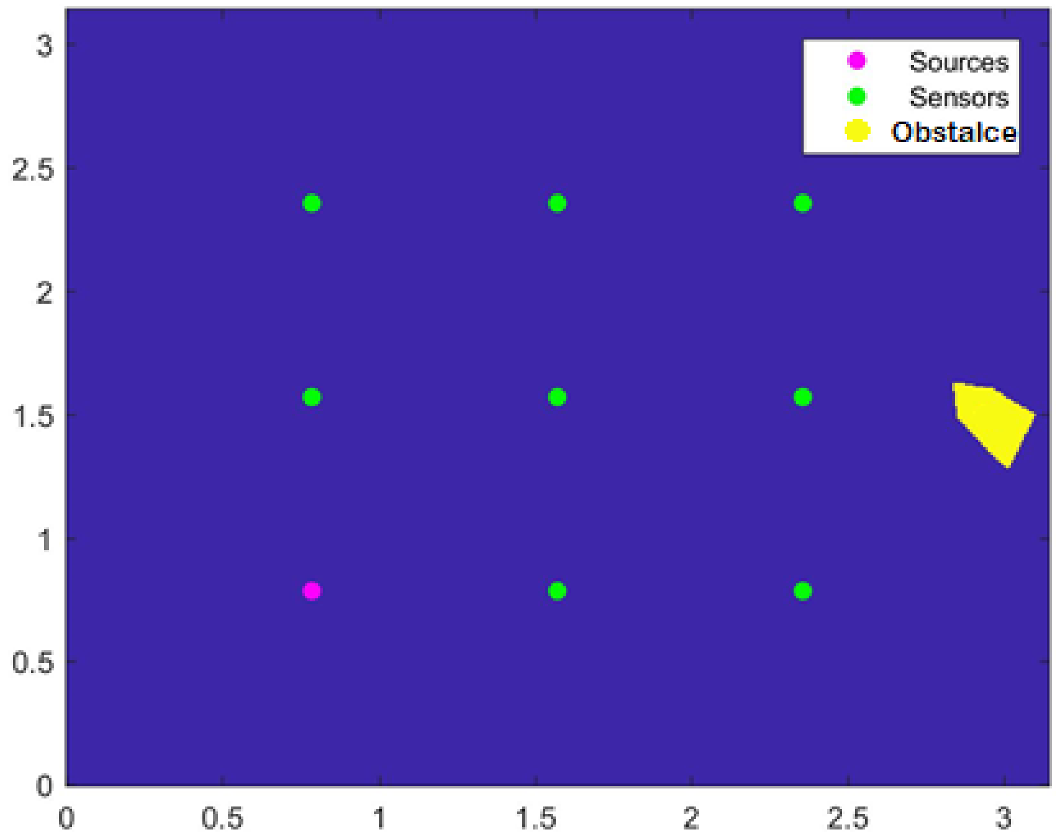
Input: Sensors recordings ($N_{samples} \times N_{tsteps} \times N_{sensors}$)

Output: Obstacle?

Prior work

- Location $\vec{x}$
- Shape and size –
  - Circles: Radius
  - Rectangles: Height and Width
  - Complex shapes: need to be parametrized
- "Soft" obstacles –
  - Semi-penetrable
  - Multiphysics

Labels solution - segments

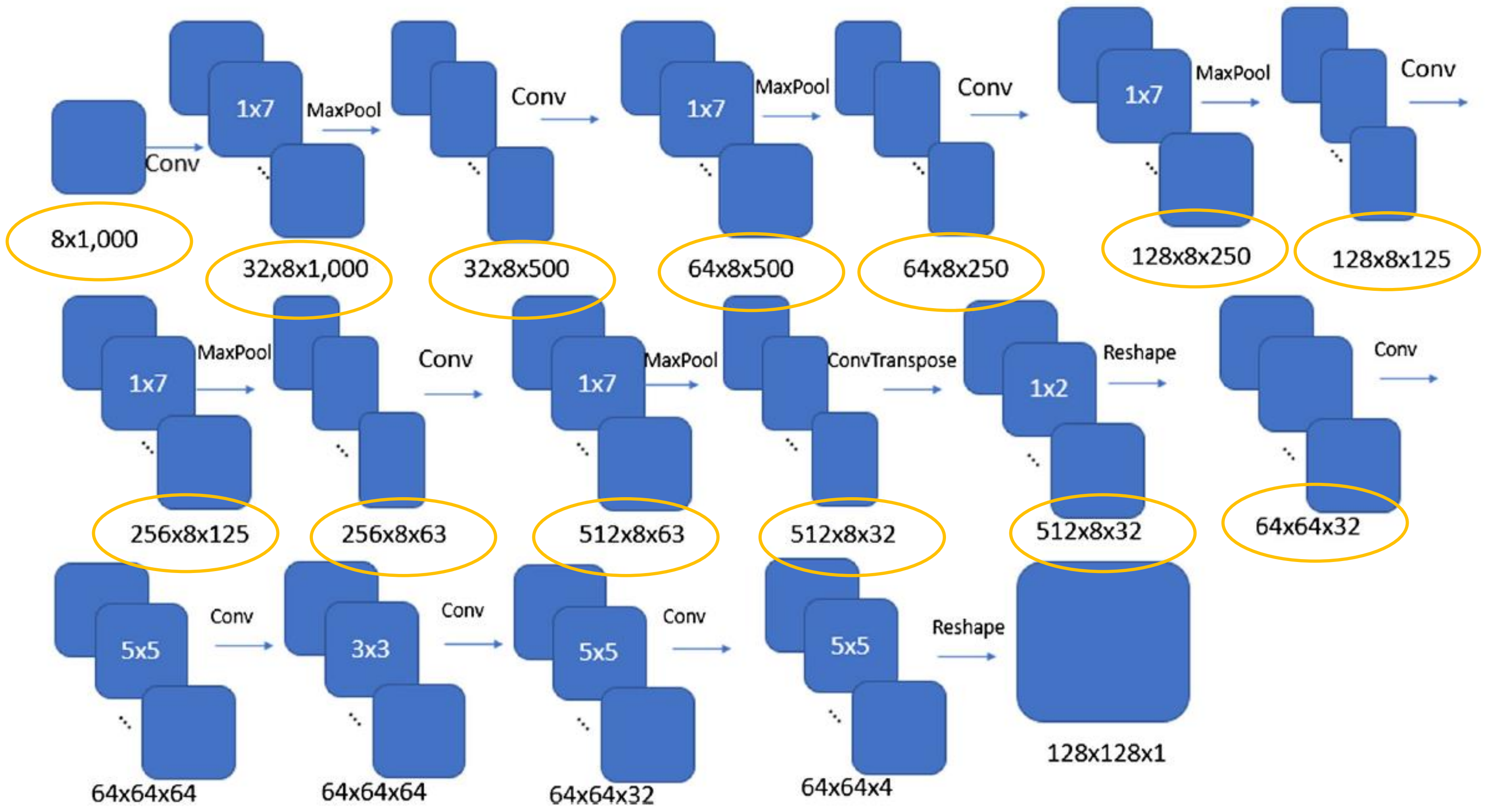


Labels are $m \times n$ **binary** matrices

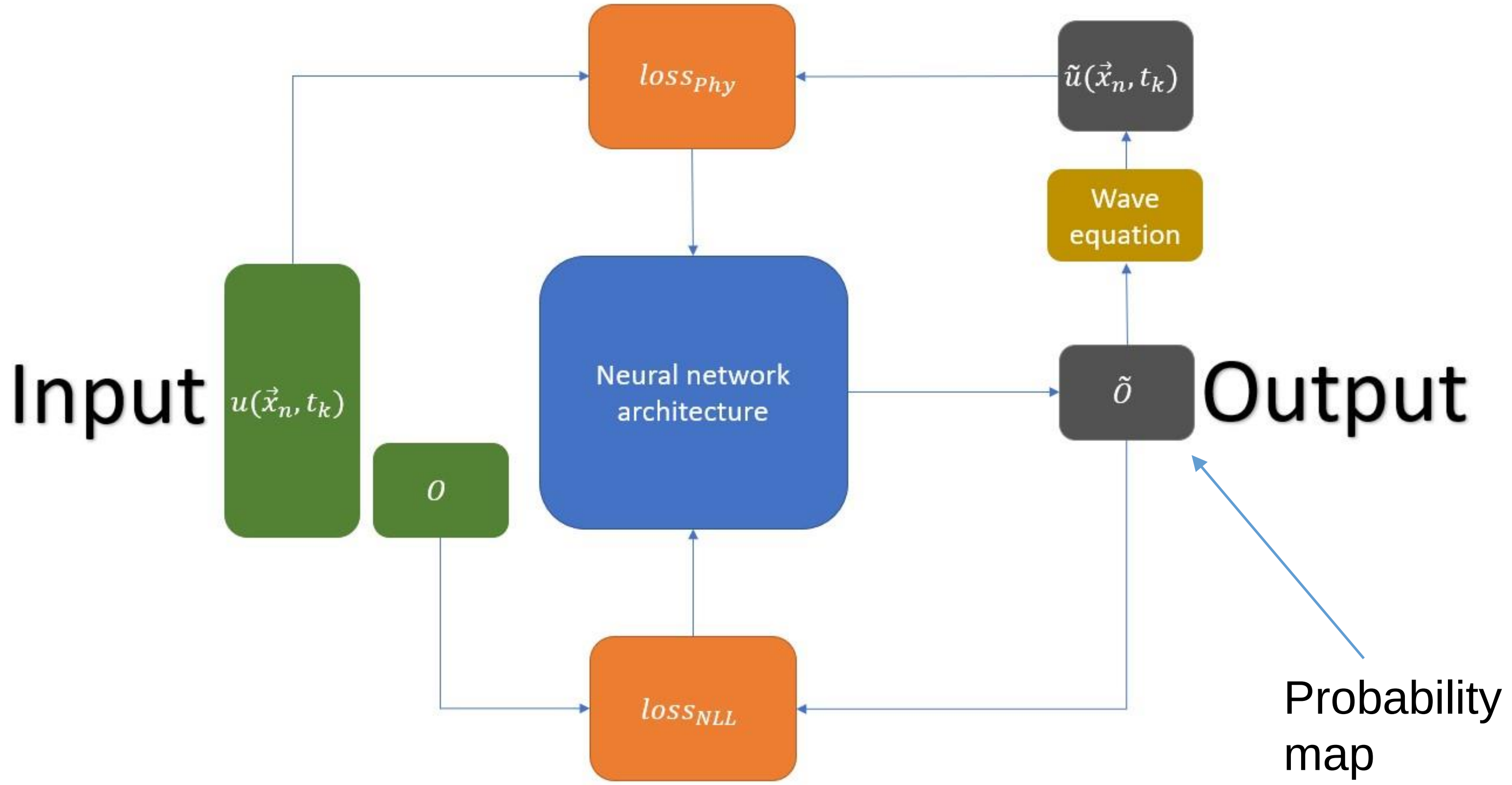
Predictions will be $m \times n$ **probability** matrices

Loss: NLL

Spatio-temporal architecture



Loss diagram



Physically informed loss

- Using the segmentation network and output $\tilde{O}$
- Define a loss component based on:
  - Solve: $u_{tt} = \left(\left(1 - \tilde{O}(x)\right) c^2(x)\right) \Delta u$
  - Get sensor data: $\{\tilde{u}_k(\overrightarrow{x_{s_i}})\}_{i=1}^{\#sensors}$ for each sample
  - Calculate MSE between ground truth $\{u_k(\overrightarrow{x_{s_i}})\}_{i=1}^{\#sensors}$ and the prediction as component l_2
- Define the loss function for our network as:

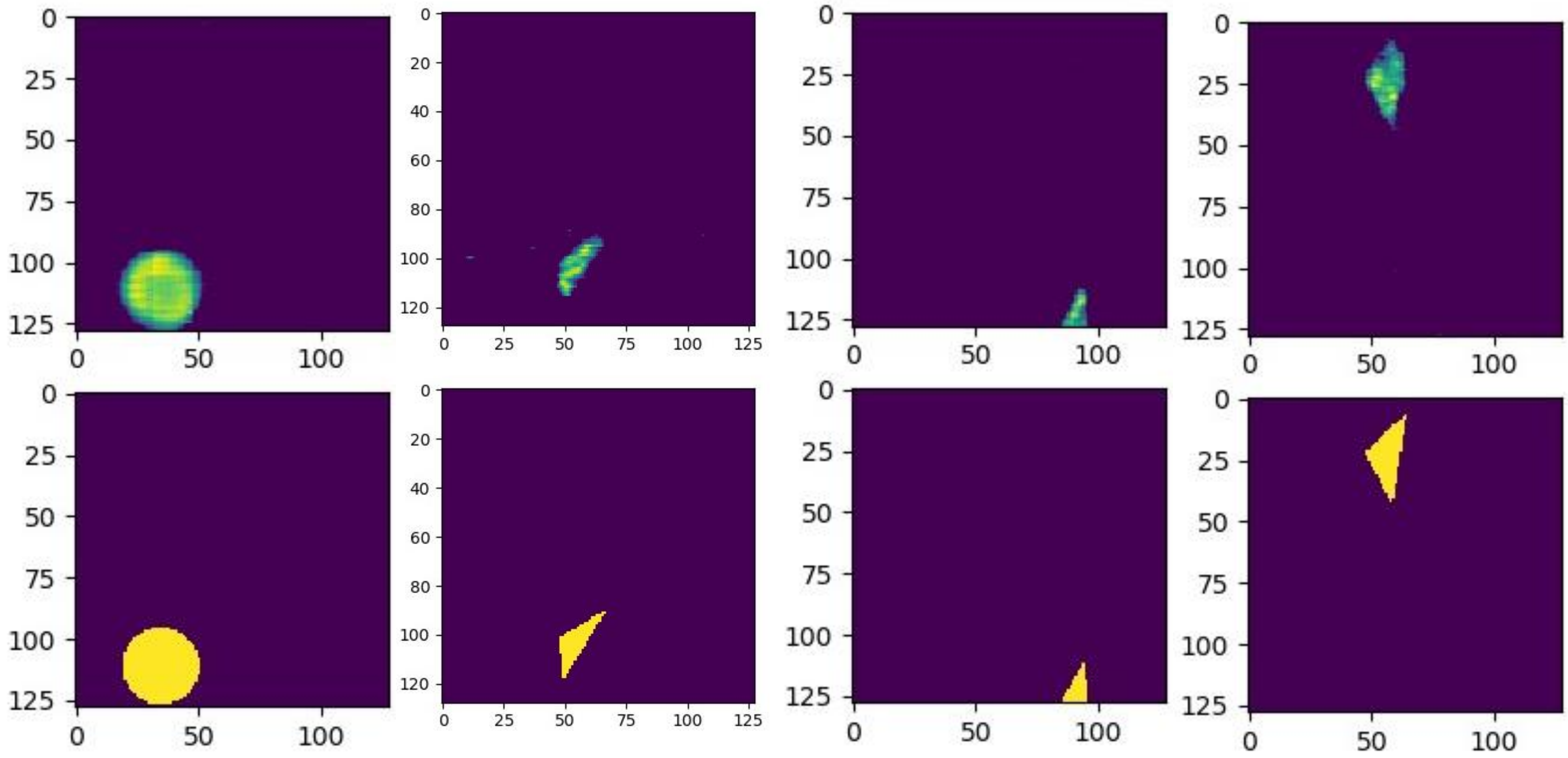
$$\alpha \cdot l_1 + (1 - \alpha) \cdot l_2$$

such that l_1 is the NLL loss described earlier

Numerical experiments

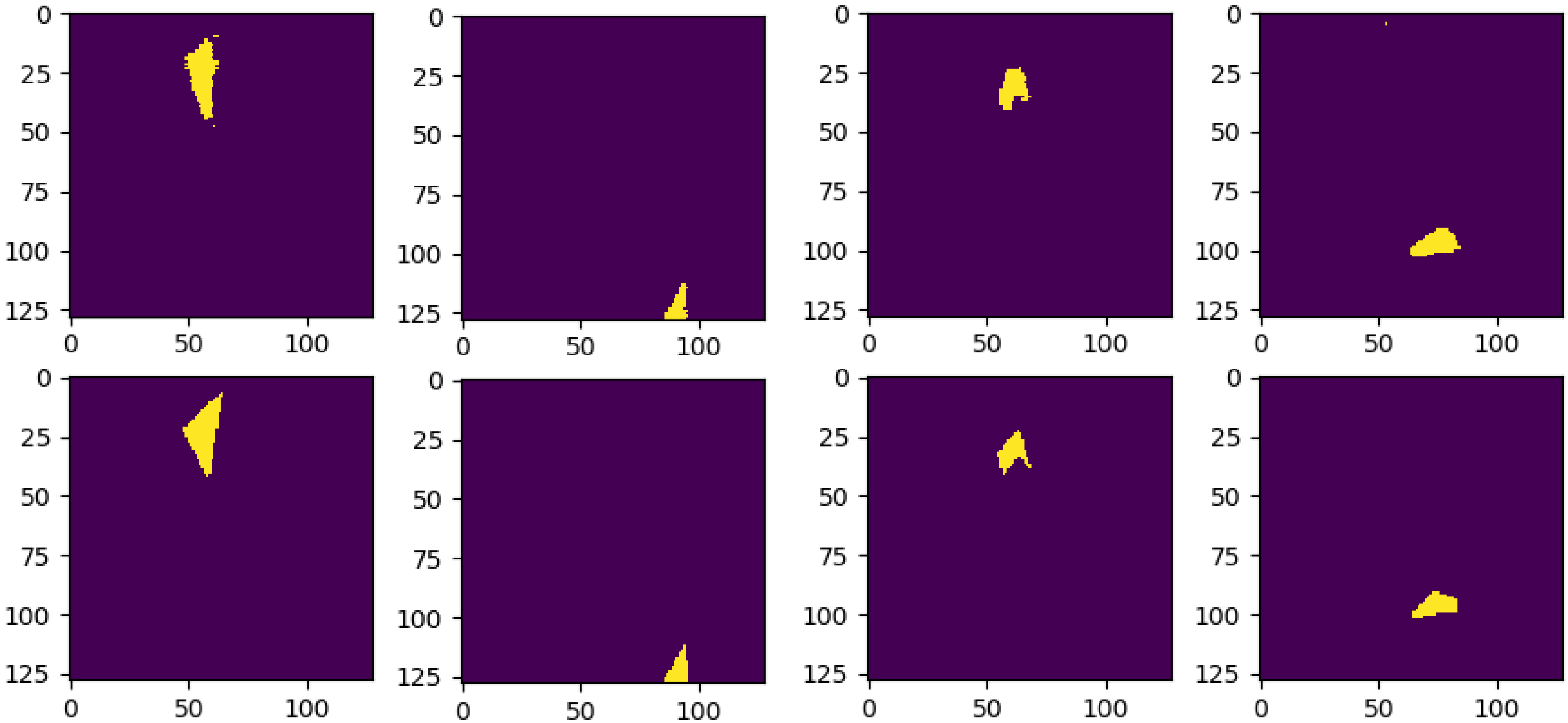
- Dirichlet BC
- Compact Gaussian initial condition
- Arbitrary polygonal obstacles –
  - Generate number of edges
  - Generate edge length and angle
  - Generate location (x_0, y_0, z_0)
  - Enormous samples space
- Generated only 25,000 samples

Probability images



Neural network – results

- Intersection over union: $0 \leq IOU(A,B) = \frac{|A \cap B|}{|A \cup B|} \leq 1$
- Up to 66% IOU score



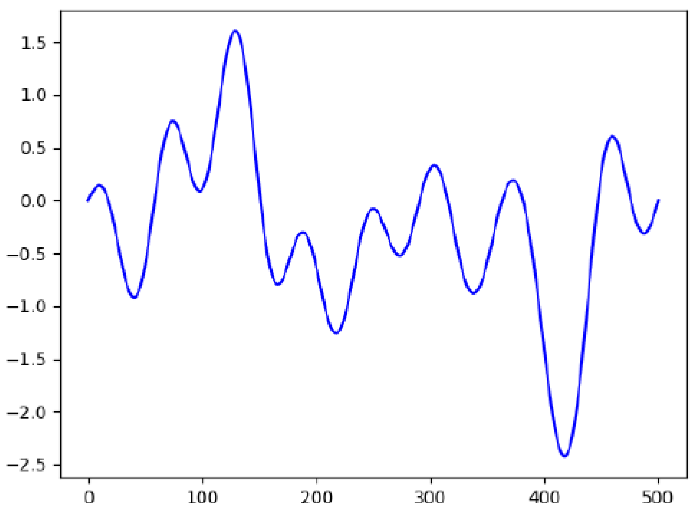
A.K., E. Turkel, D. Givoli, S. Dekel, Journal of computational Physics, 2020

Agenda

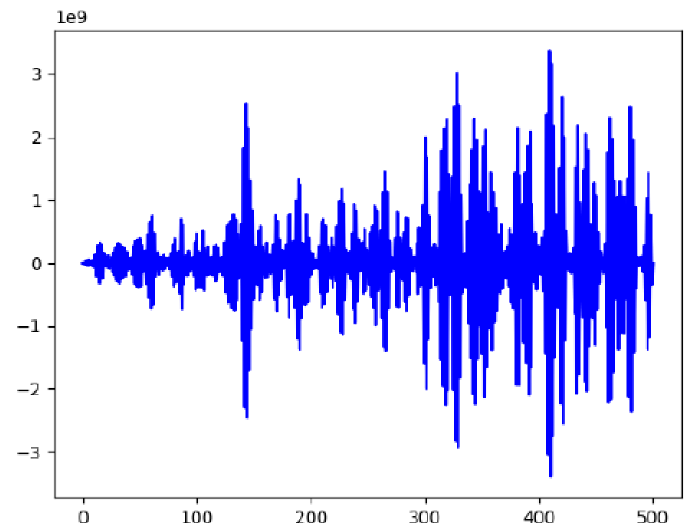
- Motivation ✓
- Data driven problems ✓
- Obstacle identification and deep-learning ✓
- Dealing with CFL instability using deep-learning

Explicit schemes and CFL

- One-dimensional wave equation
- CFL condition for stability: $\alpha = \frac{c\Delta t}{\Delta x} \leq 1$
- FDCD: $u_i^{n+1} = 2u_i^n - u_i^{n-1} + \alpha^2(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$



(a) $\alpha = 0.875$

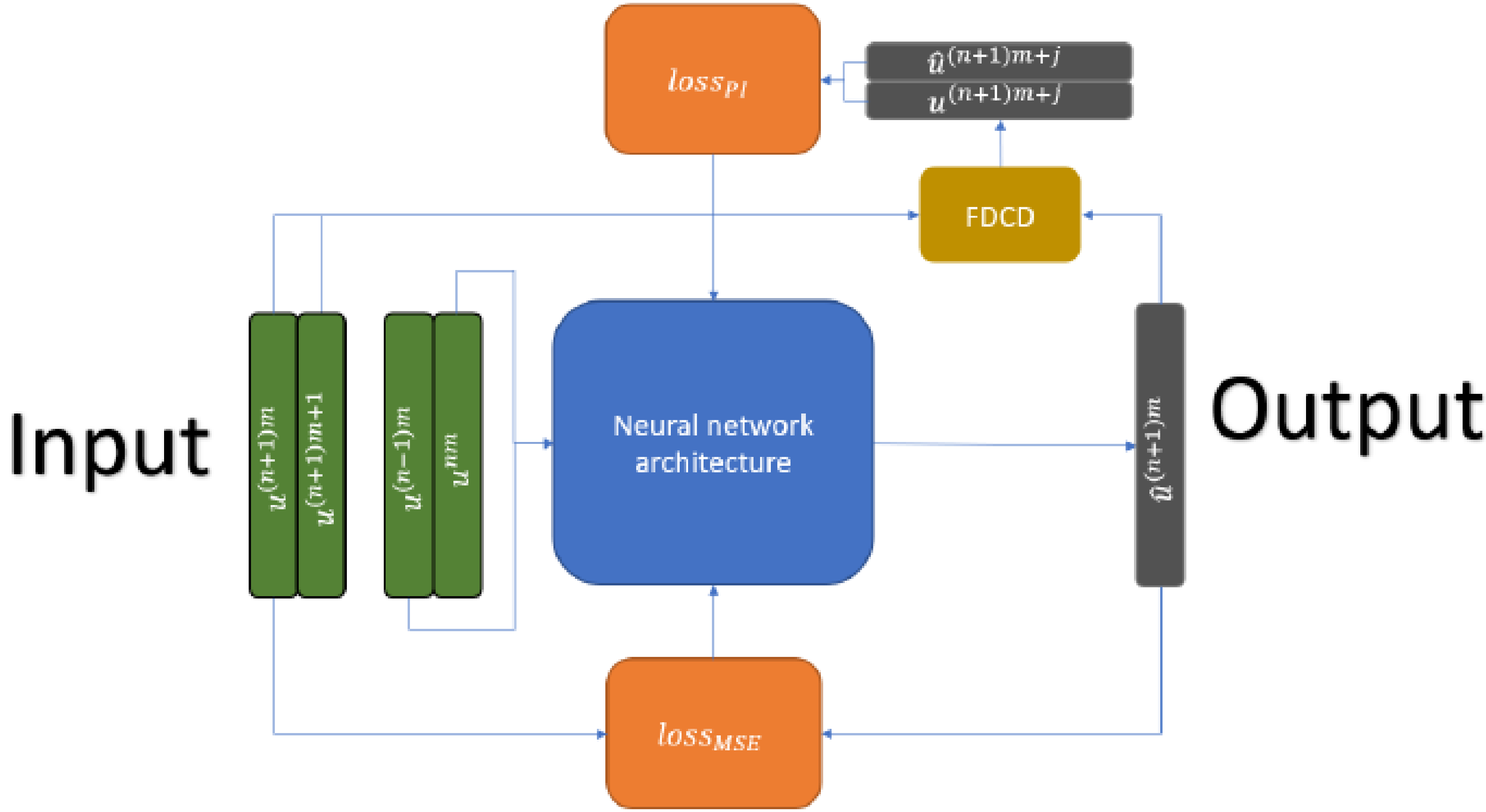


(b) $\alpha = 8.75$

Network architecture

- Input: $u^{(n-1)m}, u^{nm}$
- Output: $u^{(n+1)m}$
- Spatio-temporal architecture
- Non-linear activation (PReLU)
- Loss: MSE between $u^{(n+1)m}$ and $\hat{u}^{(n+1)m}$

Network diagram



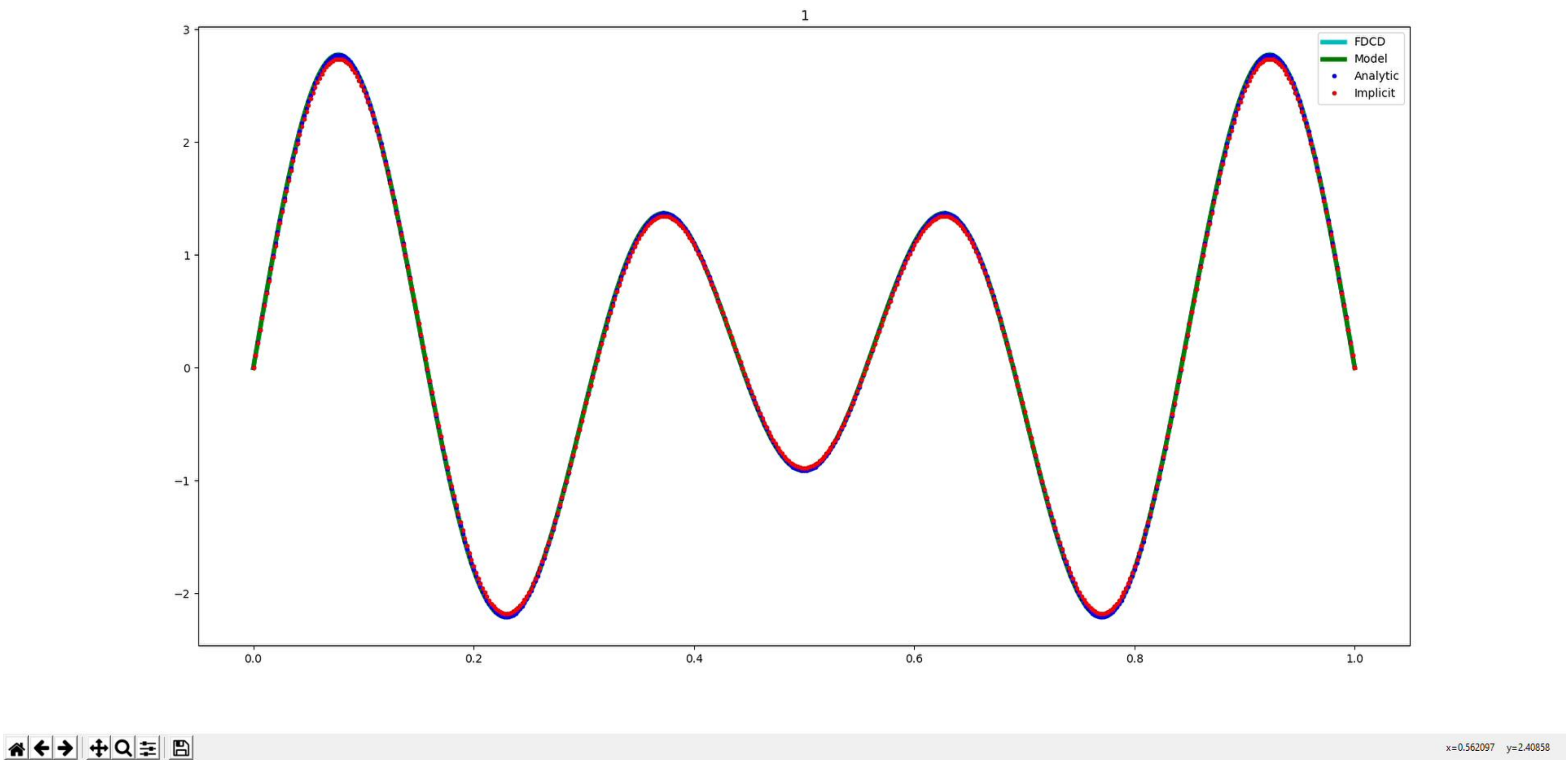
Physics informed loss

- Use $u^{(n-1)m}, u^{nm}$ to predict $\hat{u}^{(n+1)m}$
- Inside the loss:
  - Use $u^{(n+1)m}, u^{(n+1)m+1}$ to calculate $u^{(n+1)m+j}$
  - Use $\hat{u}^{(n+1)m}, u^{(n+1)m+1}$ to predict $\hat{u}^{(n+1)m+j}$
  - Calculate the MSE between $u^{(n+1)m+j}$ and $\hat{u}^{(n+1)m+j}$
- Network loss is the linear combination of the two MSE losses

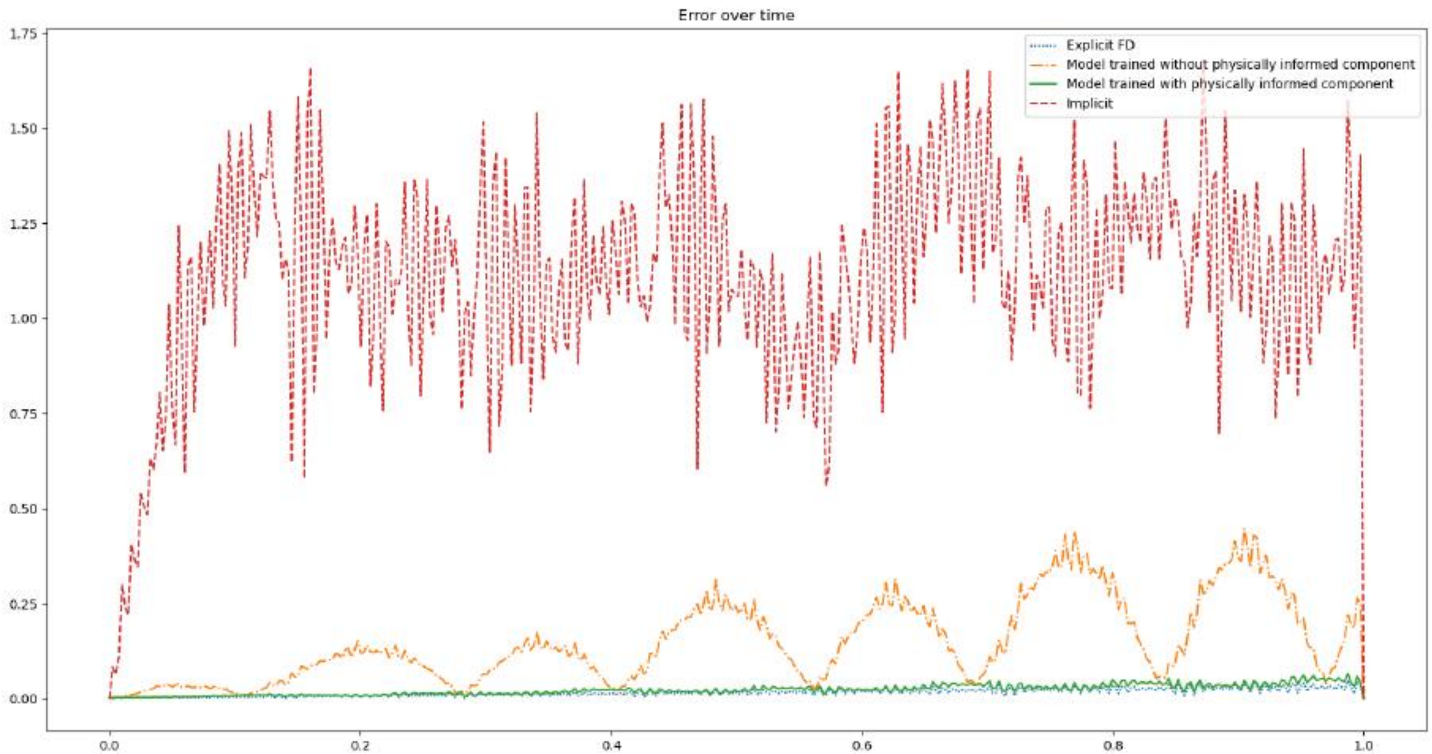
Numerical experiments

- Dirichlet BC
- Data:
  - Linear combinations with random coefficients created from the basis $\{\sin(\pi k x)\}_{k=1}^{20}$
  - 1250 different initial condition and 397 time-steps for each one, total of 496,250 samples
  - Samples created with CFL = 0.875 and only each 10^{th} sample was taken to get CFL = 8.75

Results


1
FDCD
Model
Analytic
Implicit
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
1.0
x=0.562097 y=2.40858

Results



Method	Mean of $\overline{E}$	Mean of $max(E_{ic}(t_n))$	Mean of $median(E_{ic}(t_n))$
Model trained without physically-informed component	0.197140	0.665867	0.153857
Model trained with physically-informed component	**0.021158**	**0.062016**	**0.018769**
Explicit FD	0.011912	0.036059	0.010889
Implicit	1.122231	1.791727	1.155944

O. Ovadia, A. K, E. Turkel, S. Dekel, Journal of computational physics, submitted

Summary and future work

- Obstacle location and identification
  - Investigating source location
  - High measurement noise
- Stability
  - Extending to 2,3 dimensions
  - Dispersion relation problem – optimized kernels
  - Experimental data

Thanks!