PHYSICS INFORMED NEURAL NETWORKS FOR SOLVING WAVE EQUATION

Harpreet Sethi, 6th November 2021

Colorado School of Mines, Email: hsethi@mines.edu

ACKNOWLEDGEMENTS

NVIDIA Energy team

- Ken Hester, Doris Pan, Pavel Dimitrov, Gunter Roth

NVIDIA SimNet team

- Sanjay Choudhary, Oliver Hennigh, Kaustabh Sanghali, Zhiwei

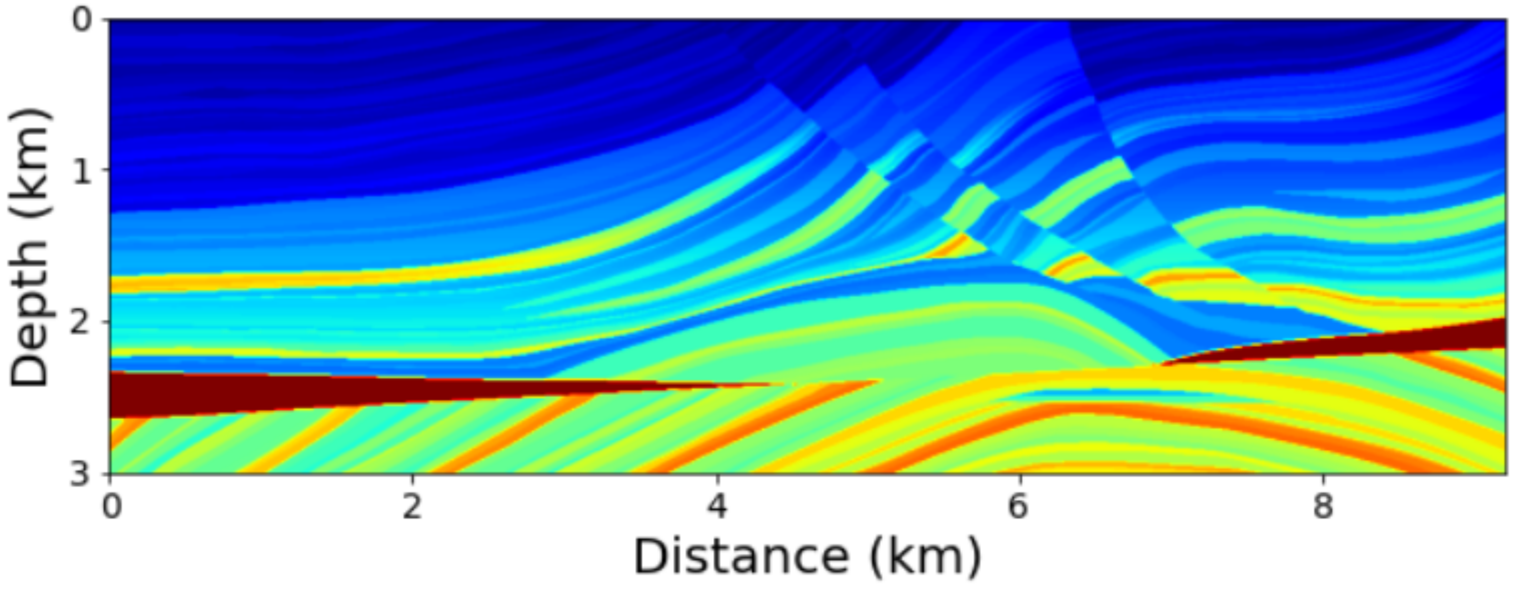
AGENDA

- Motivation
- Theory
- Experimental setup
- Numerical examples
- Conclusion and next steps

AGENDA

- Motivation
- Theory
- Experimental setup
- Numerical examples
- Conclusion and next steps

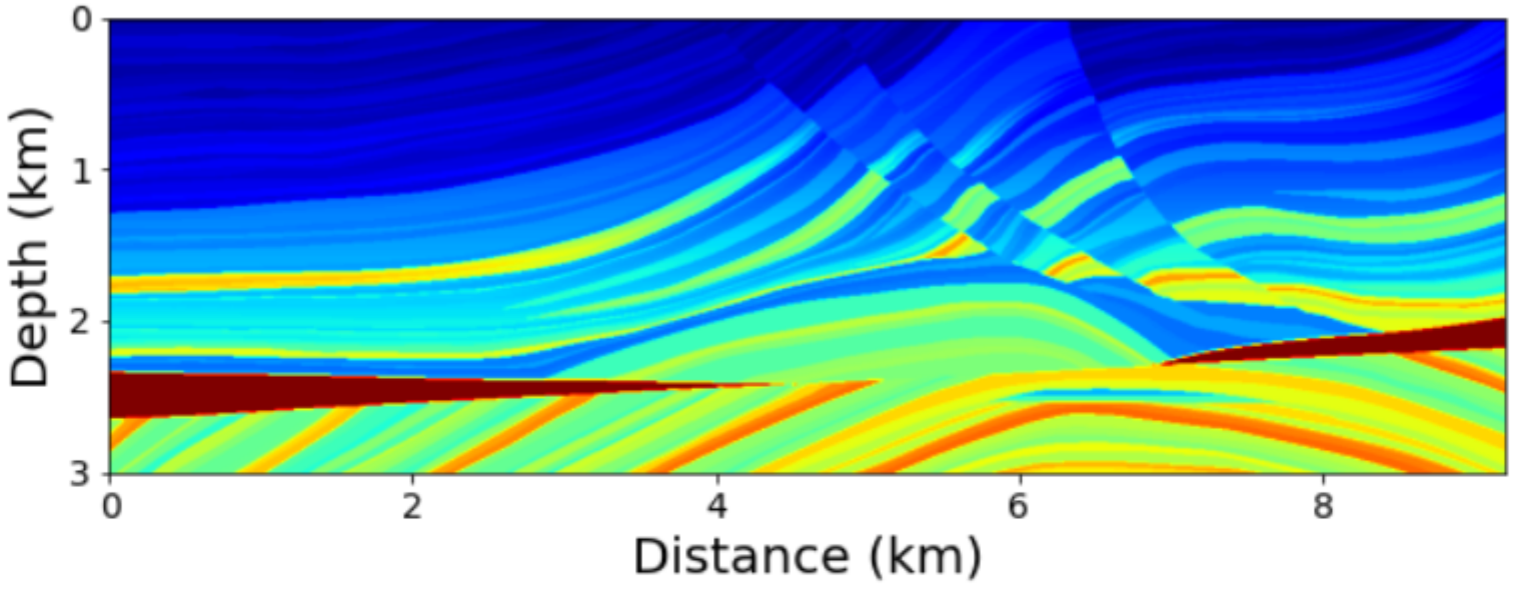
NUMERICAL SOLVERS



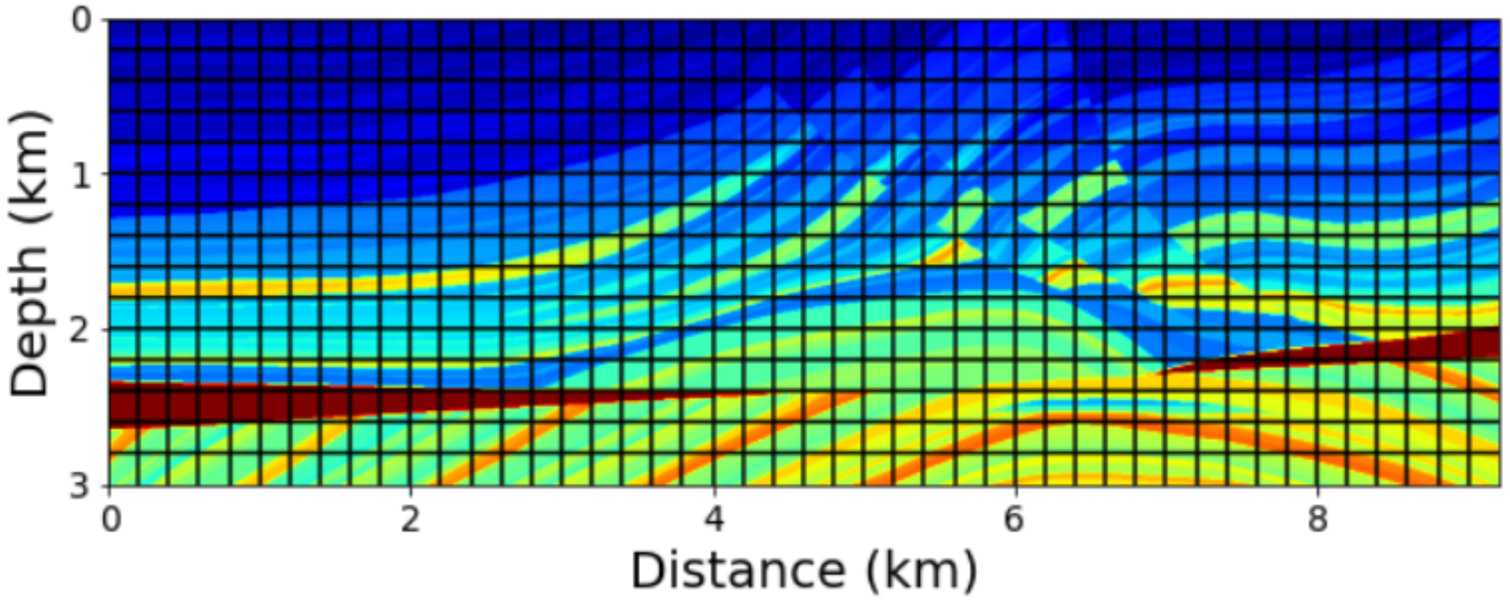
Model domain

Finite-difference (FD), Finite-element (FE) , Finite-volume, Pseudo-spectral methods etc.

NUMERICAL SOLVERS



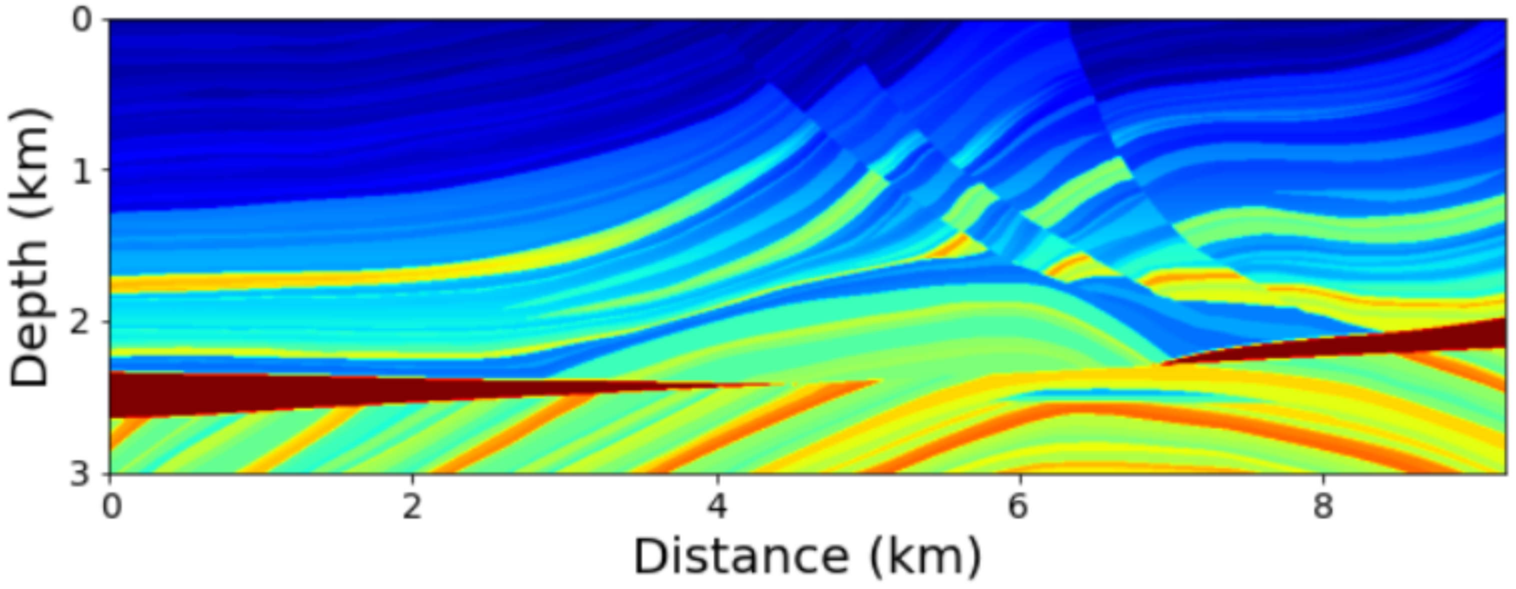
Model domain



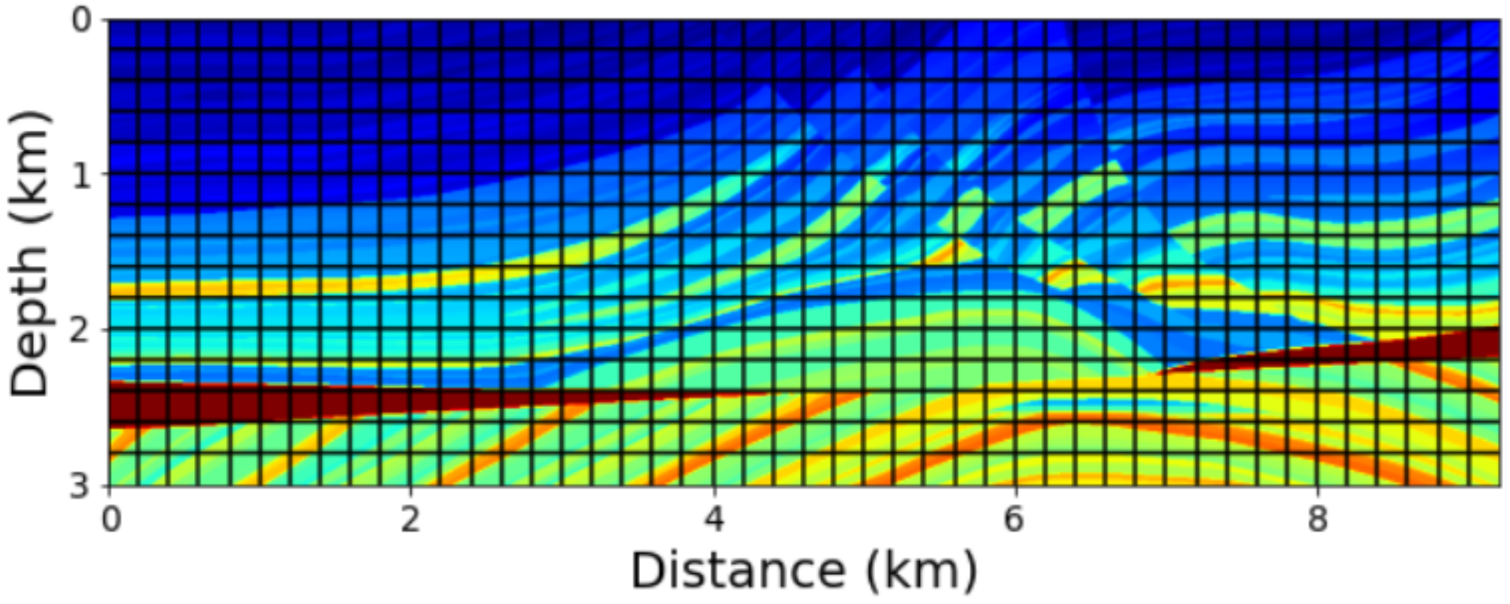
Discretize model/generate a mesh

Finite-difference (FD), Finite-element (FE) , Finite-volume, Pseudo-spectral methods etc.

NUMERICAL SOLVERS



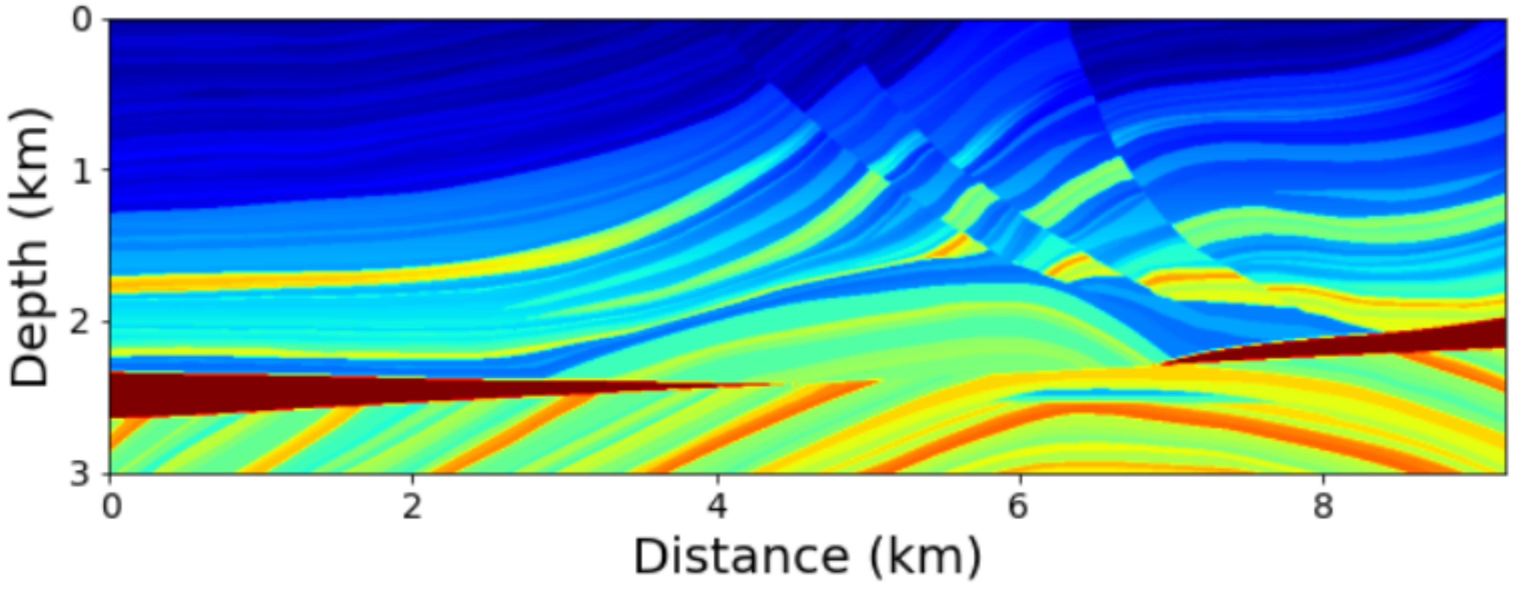
Model domain



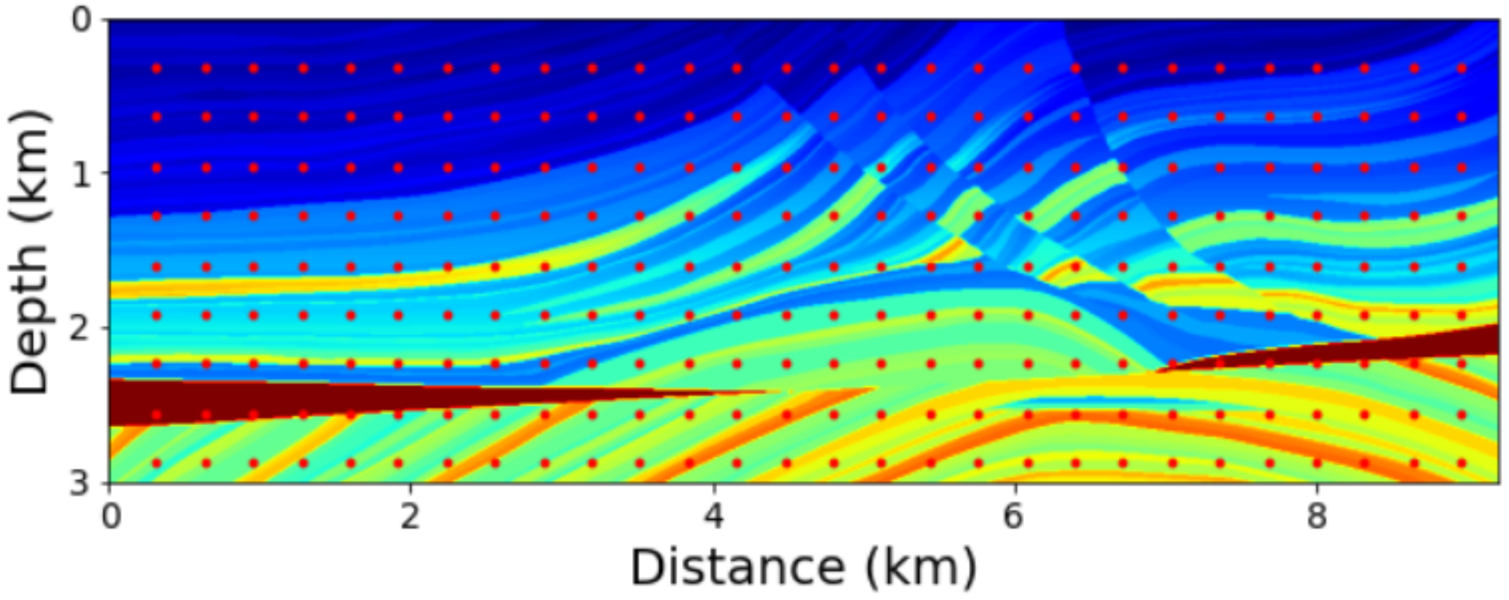
Discretize model/generate a mesh

- Numerical errors such as dispersion, grid anisotropy, truncation etc.

PHYSICS INFORMED NEURAL NETWORK (PINN) SOLVER



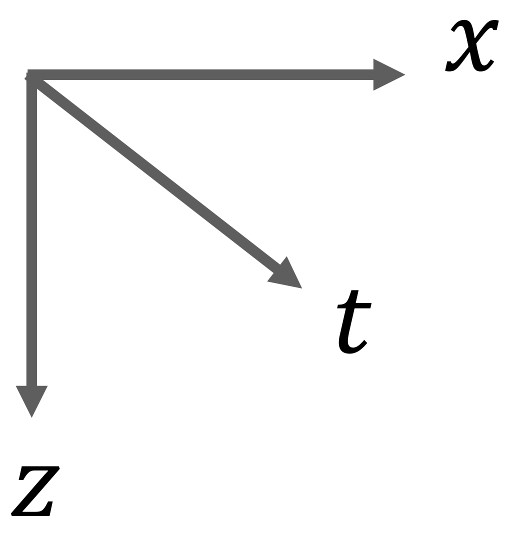
Model domain

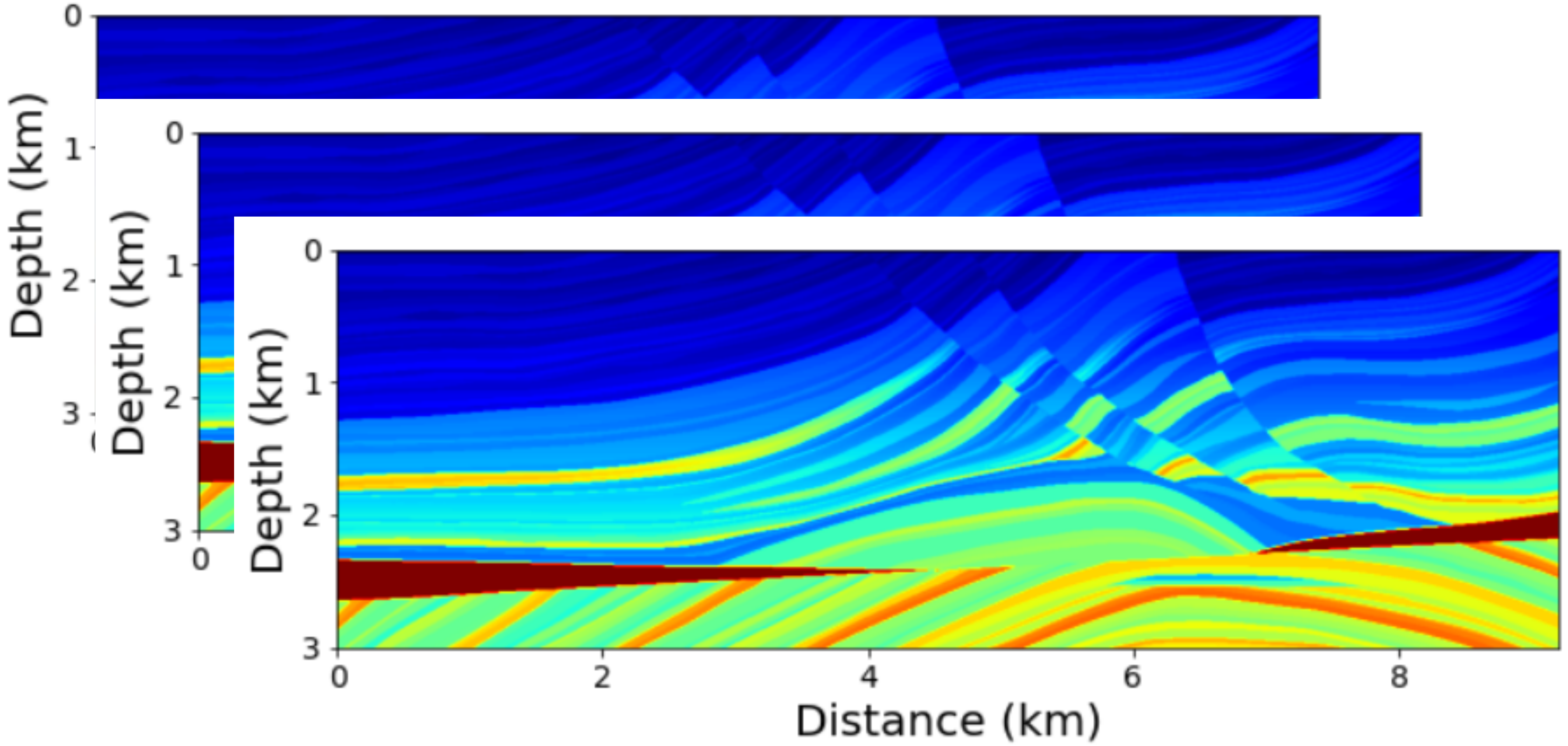


Point cloud representation

- No mesh generation
- Less prone to numerical errors

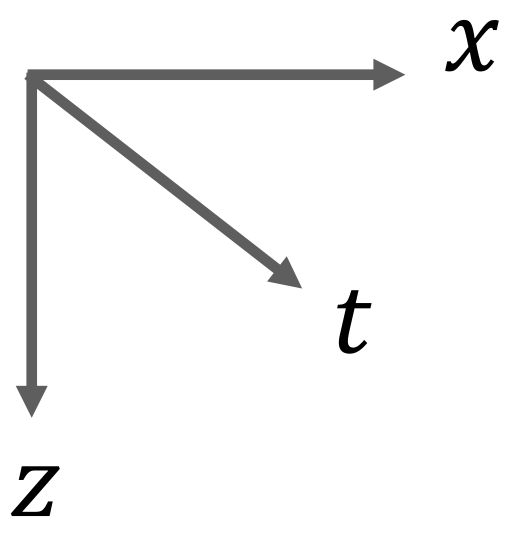
NUMERICAL SOLVERS

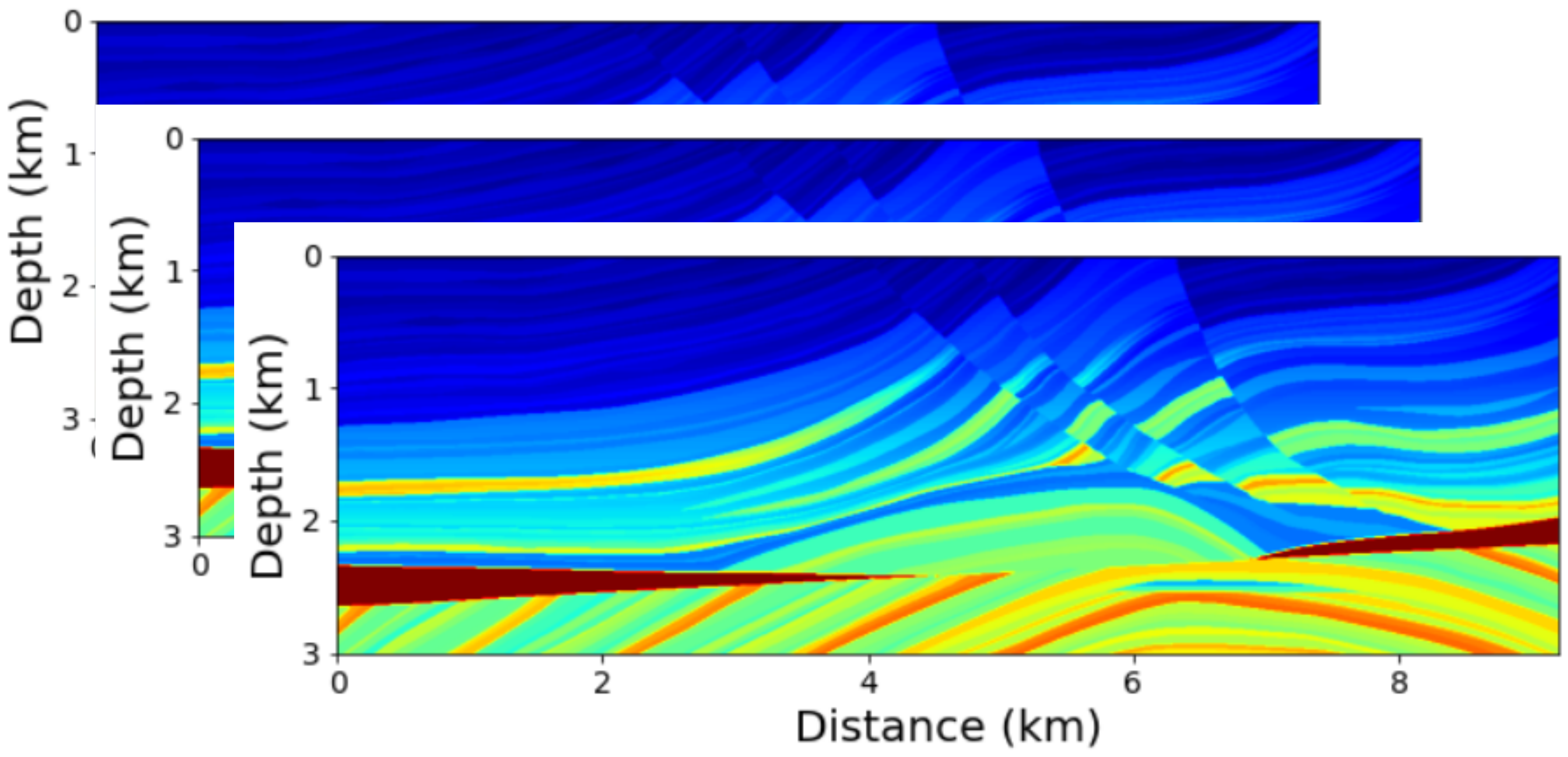




Need to solve for every time step

PINN SOLVER



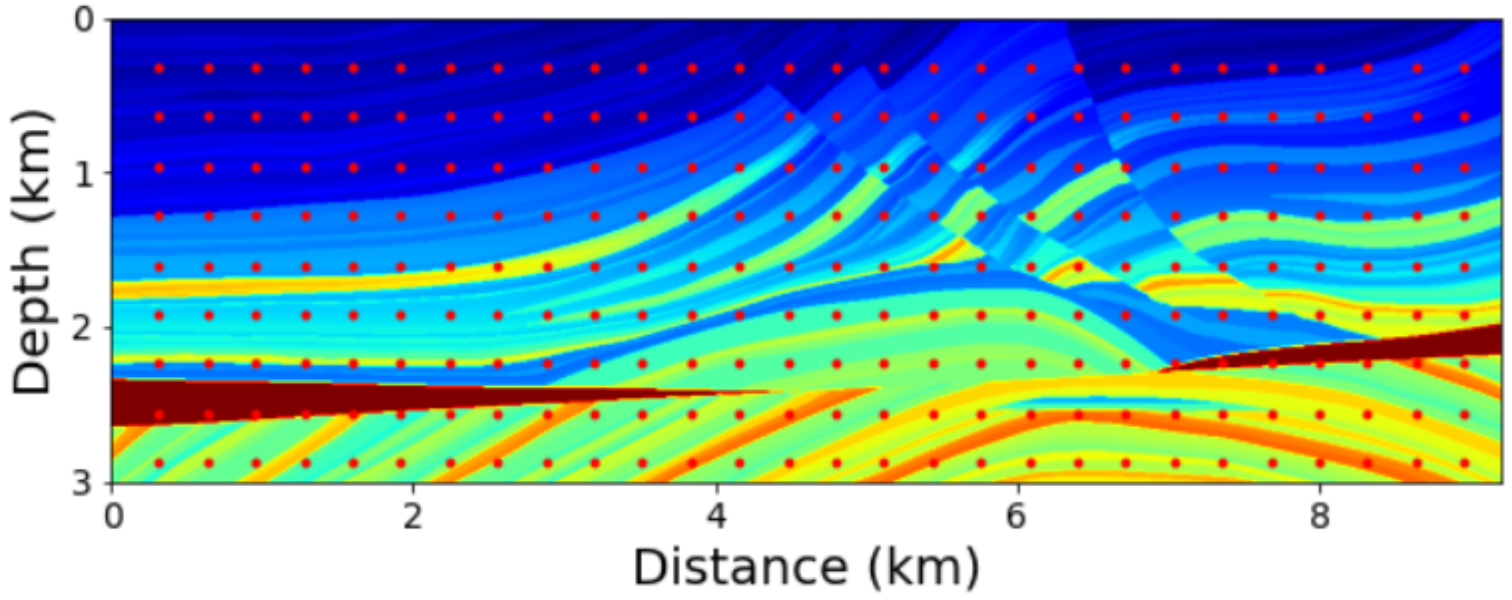


Resolve complete spatial-temporal domain in single step

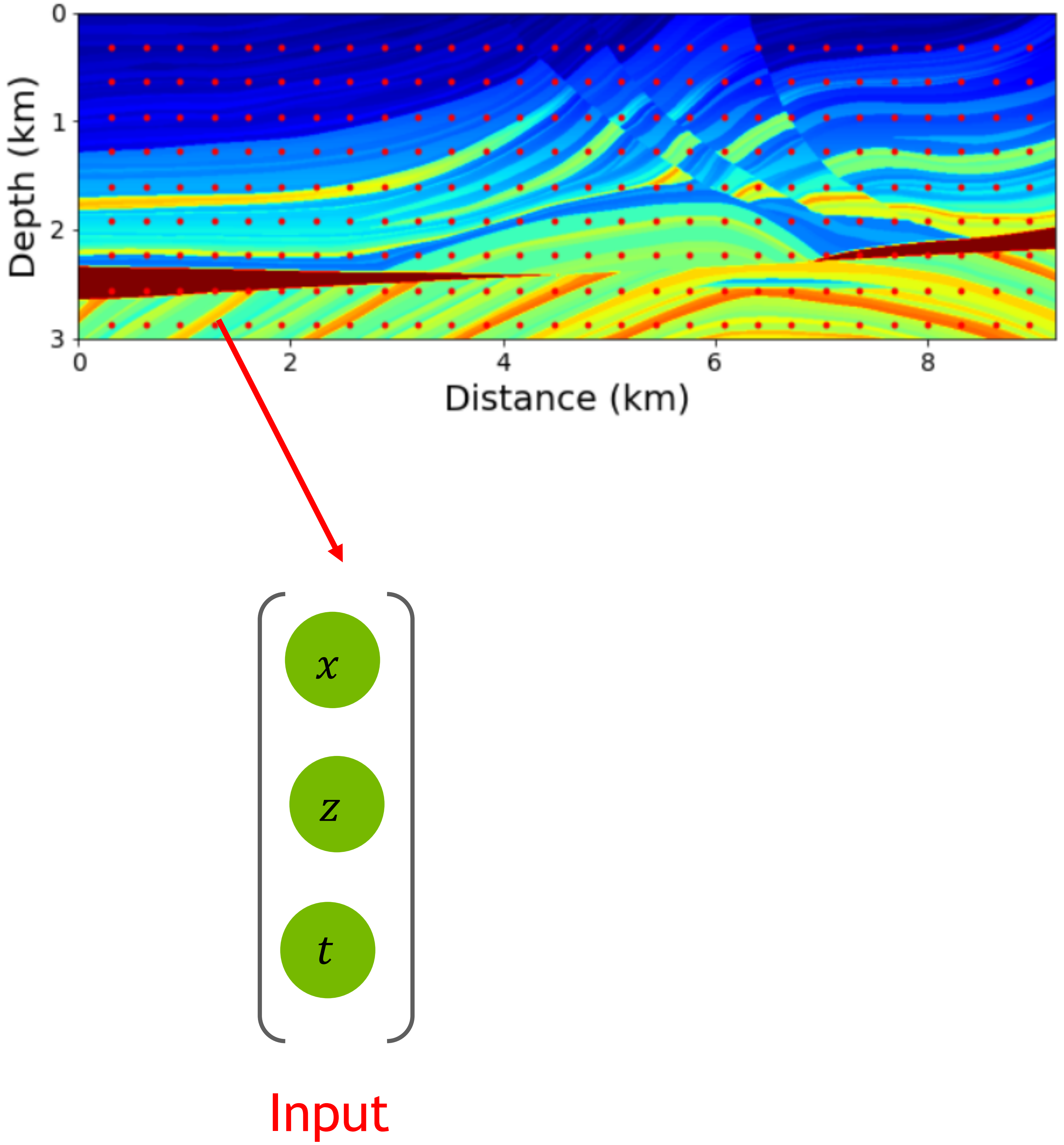
AGENDA

- Motivation
- Theory
- Experimental setup
- Numerical examples
- Conclusion and next steps

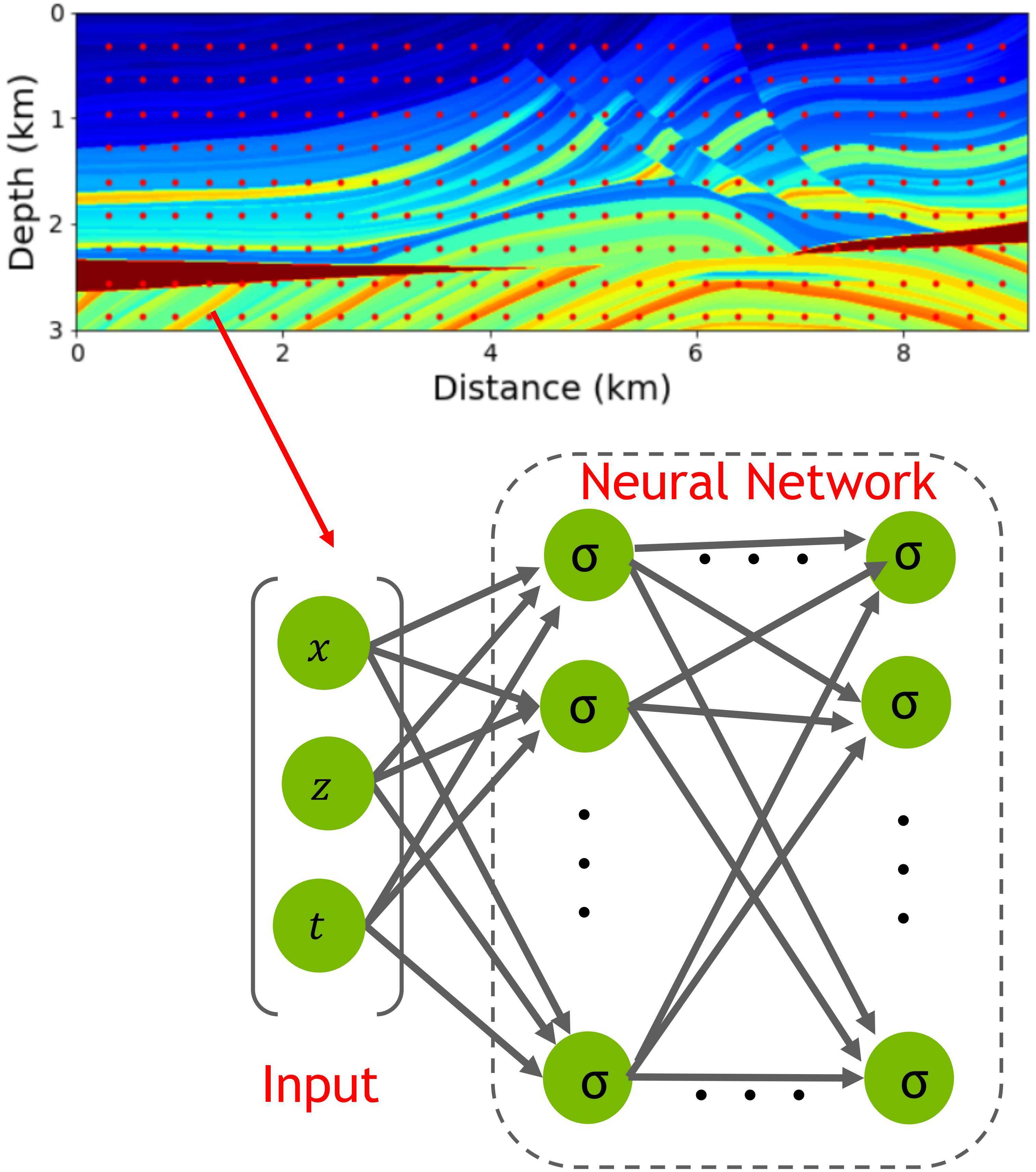
PINN FRAMEWORK



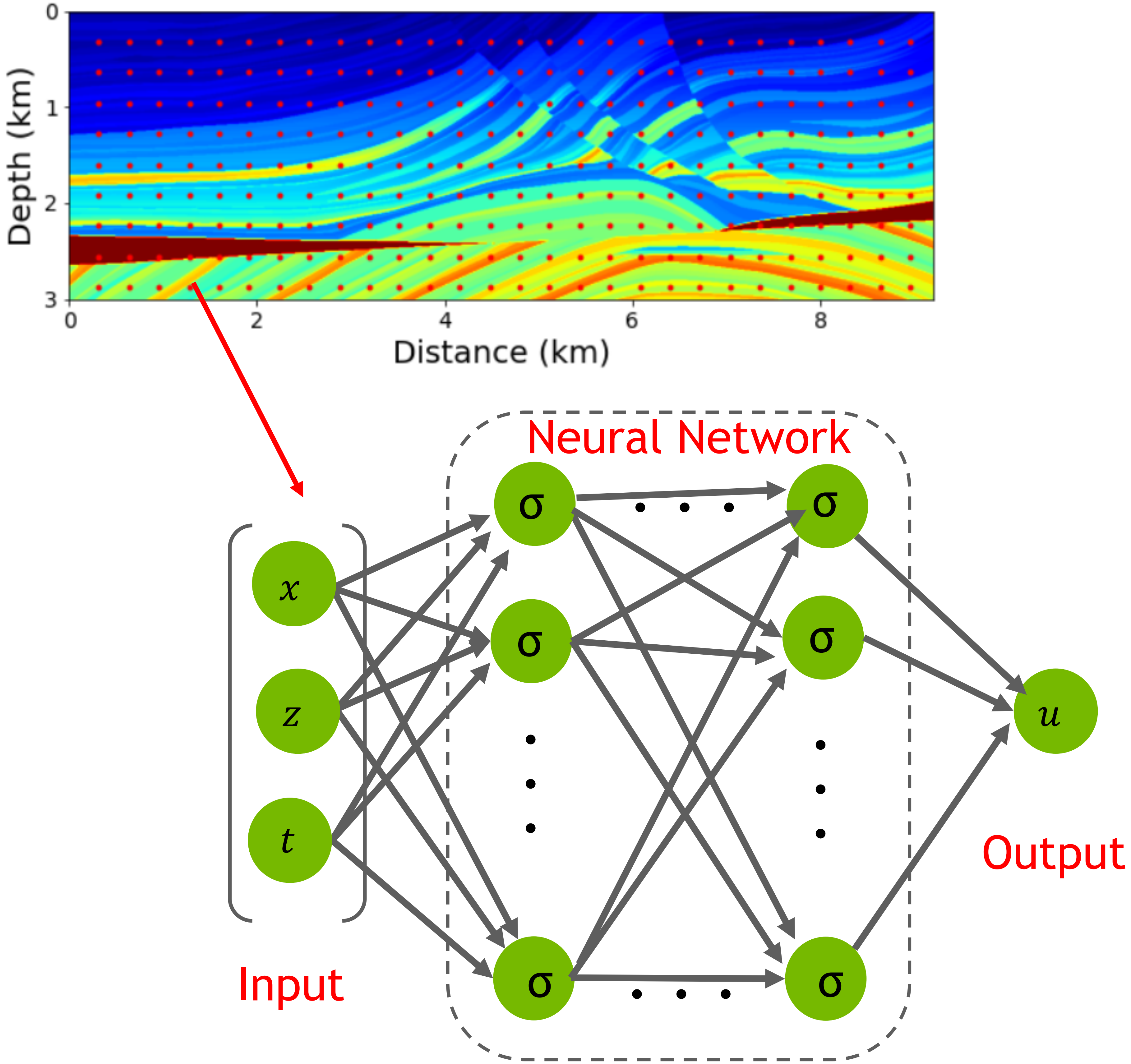
PINN FRAMEWORK



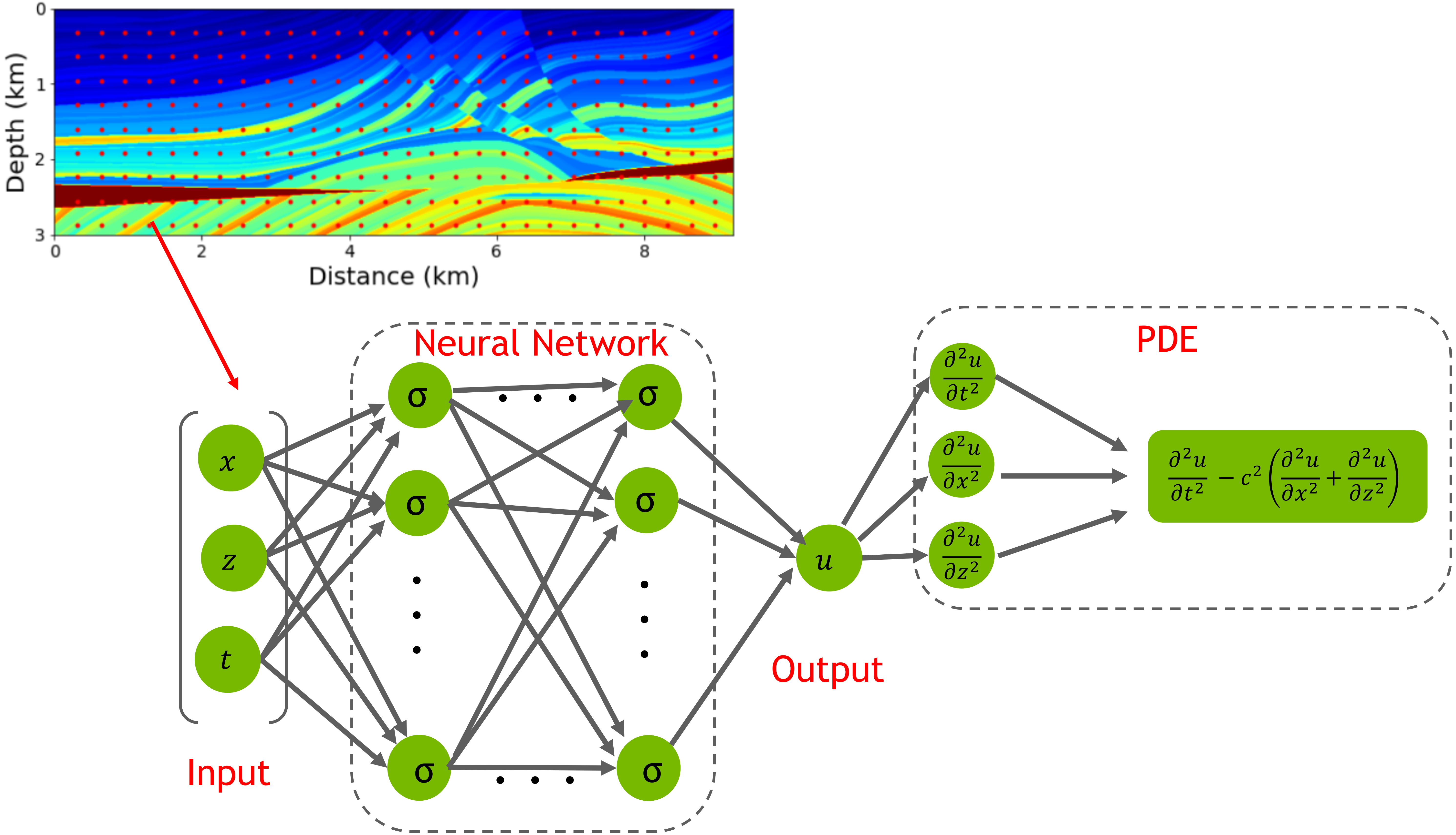
PINN FRAMEWORK



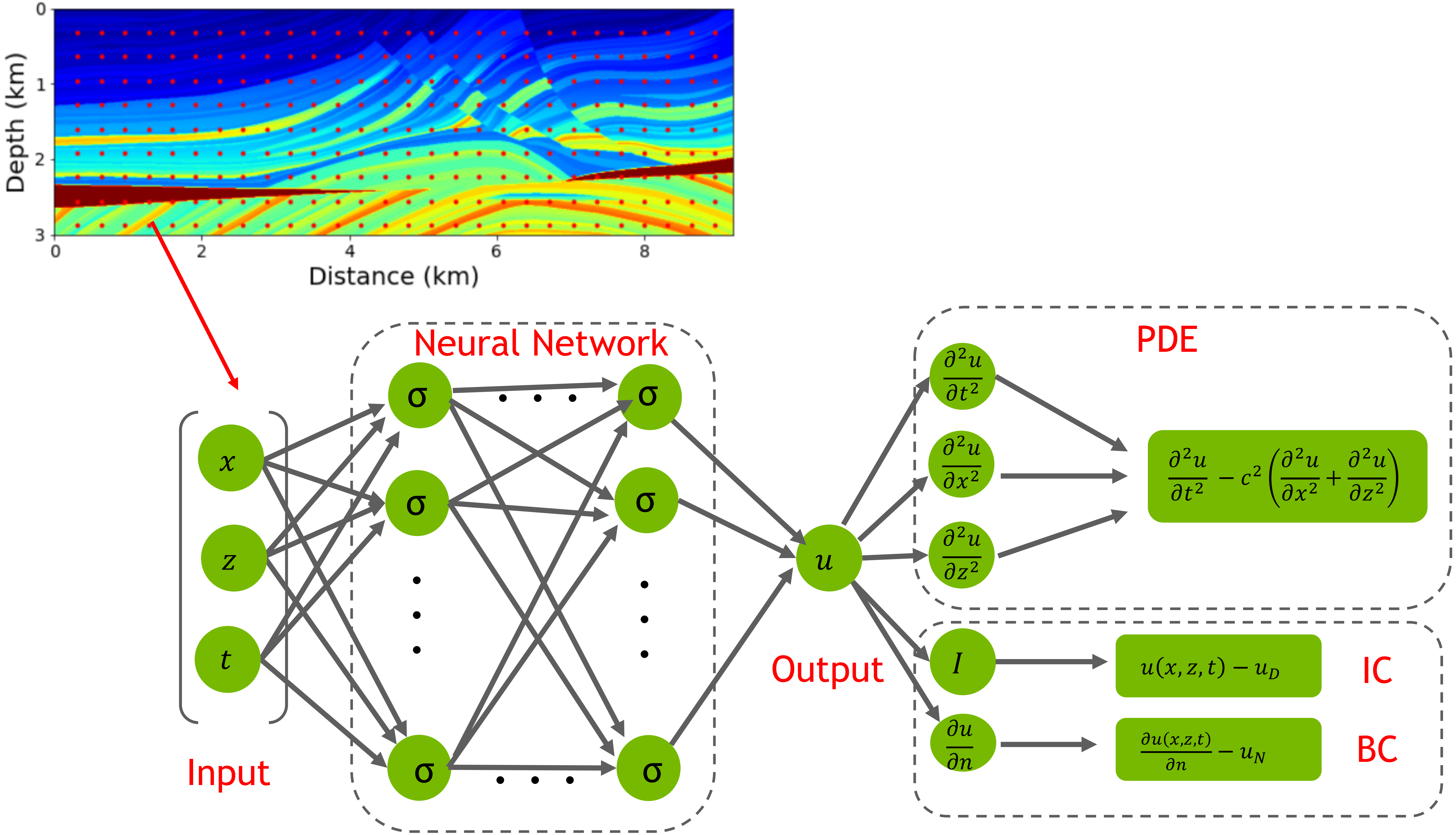
PINN FRAMEWORK



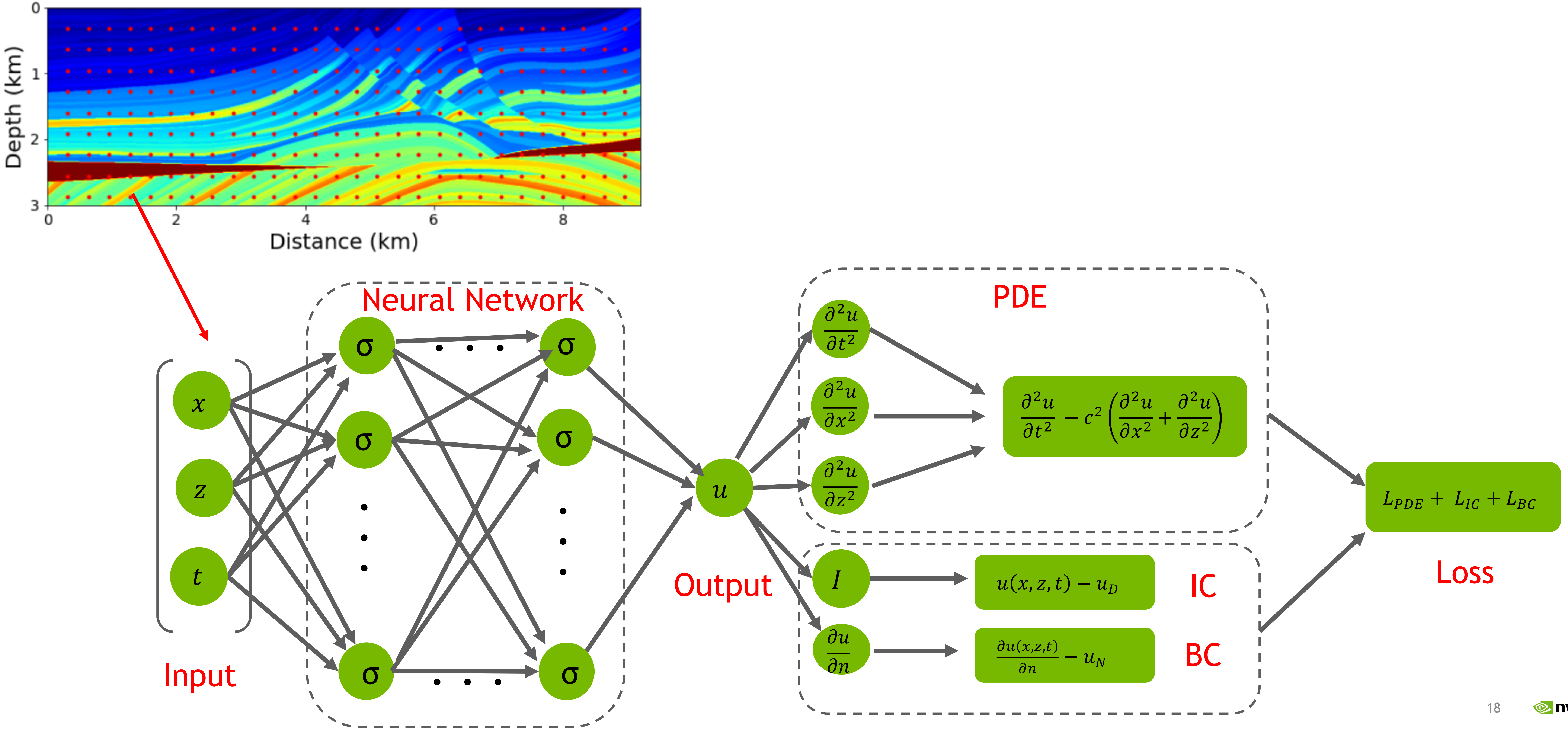
PINN FRAMEWORK



PINN FRAMEWORK

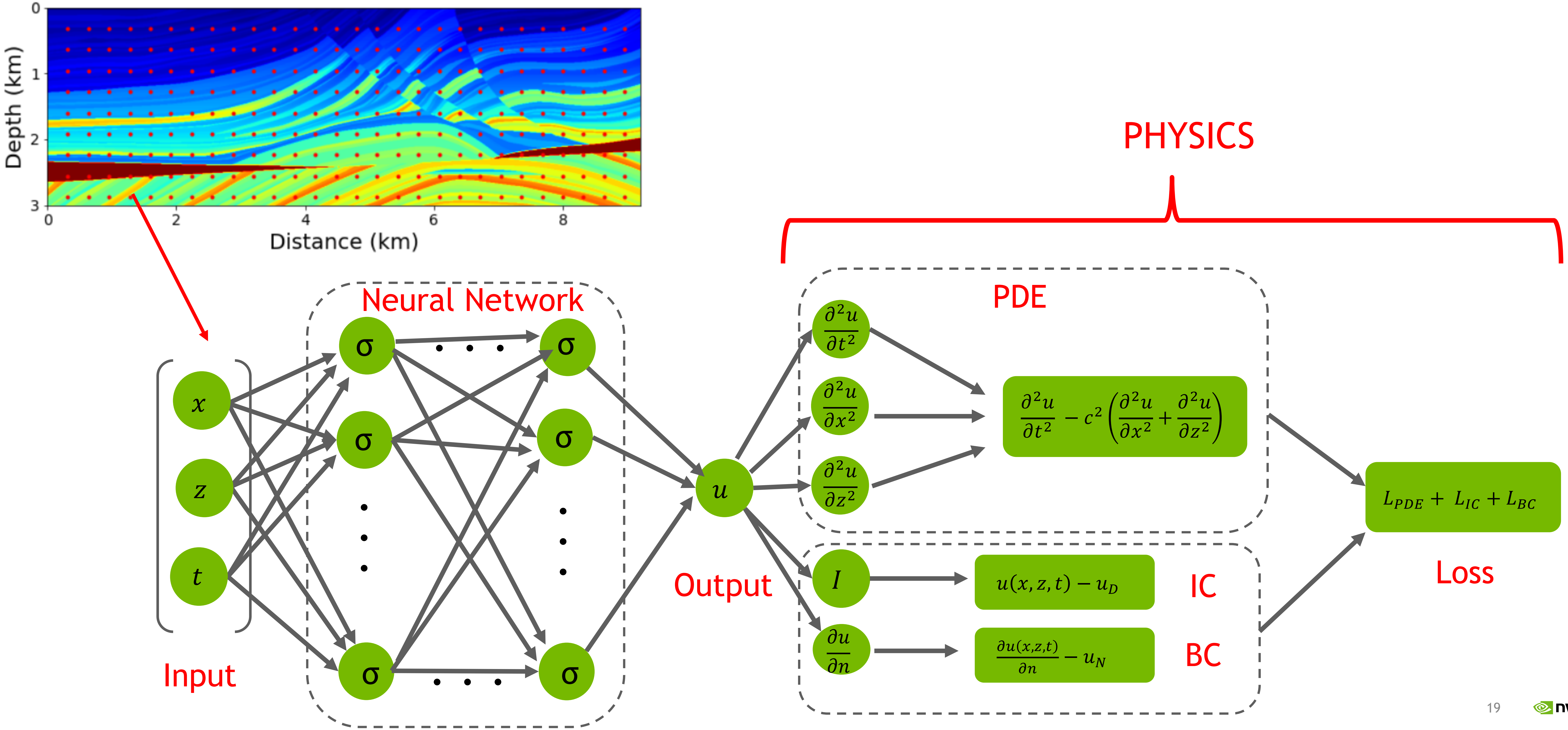


PINN FRAMEWORK

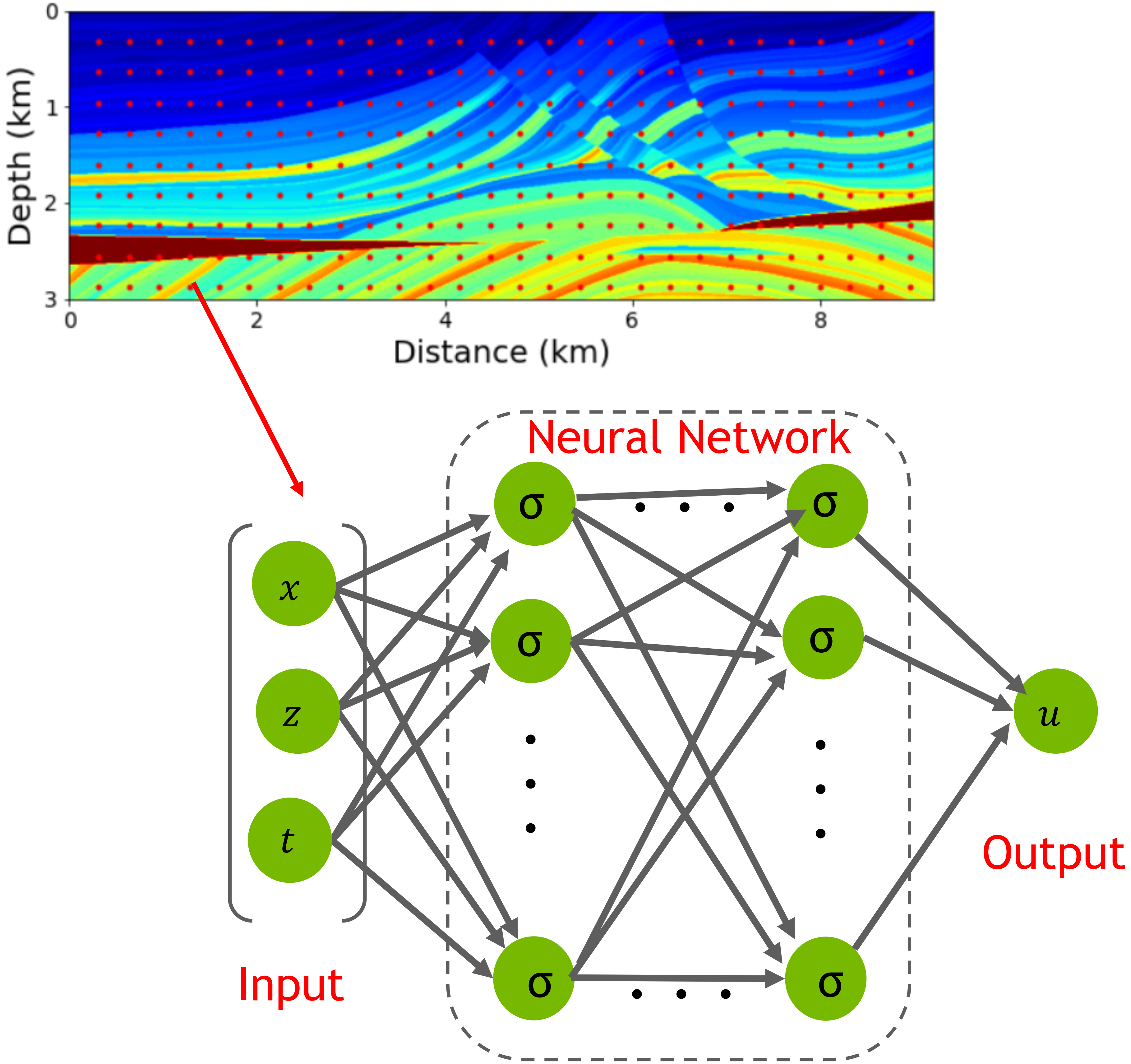


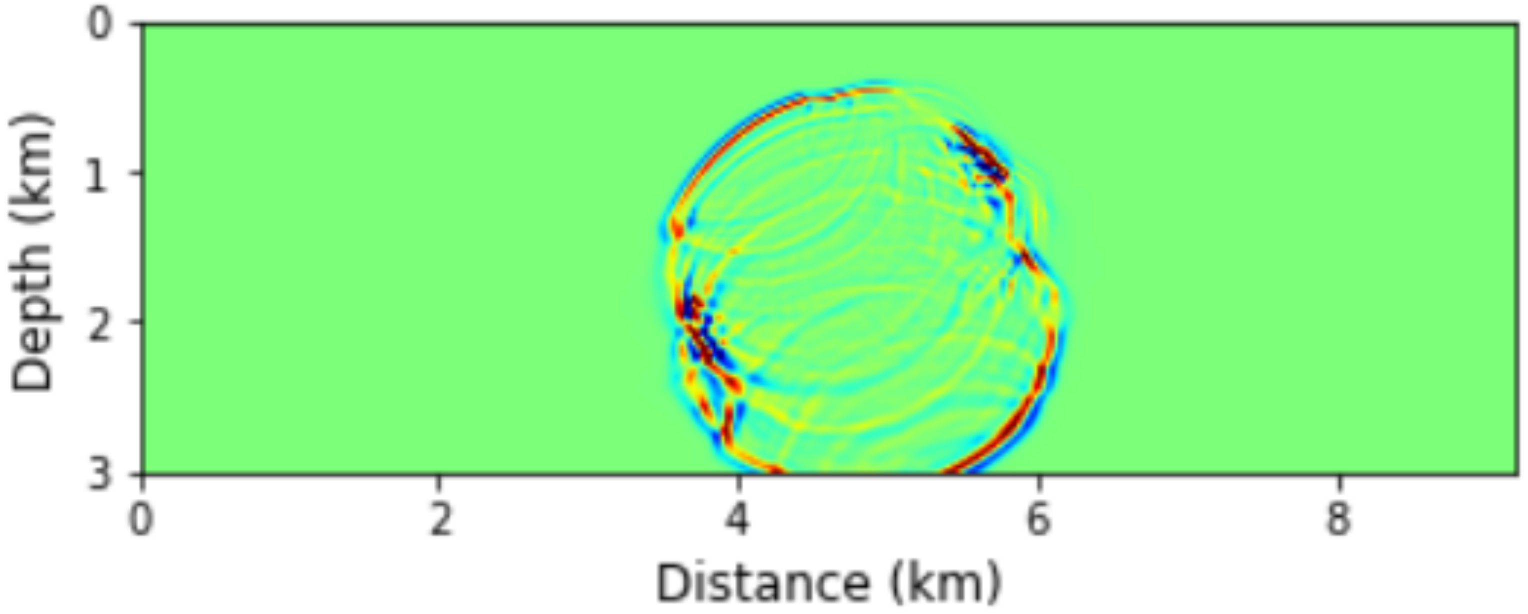
NVIDIA

PINN FRAMEWORK



PINN FRAMEWORK



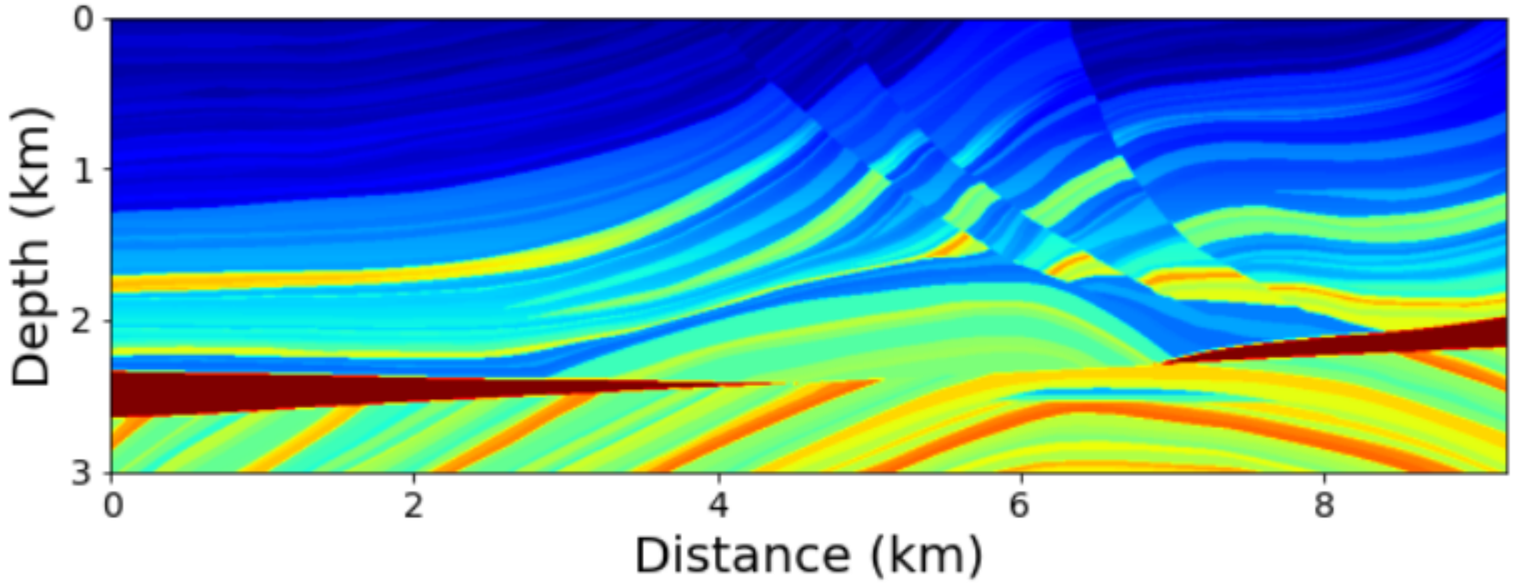


Goal: To predict wavefield solution

AGENDA

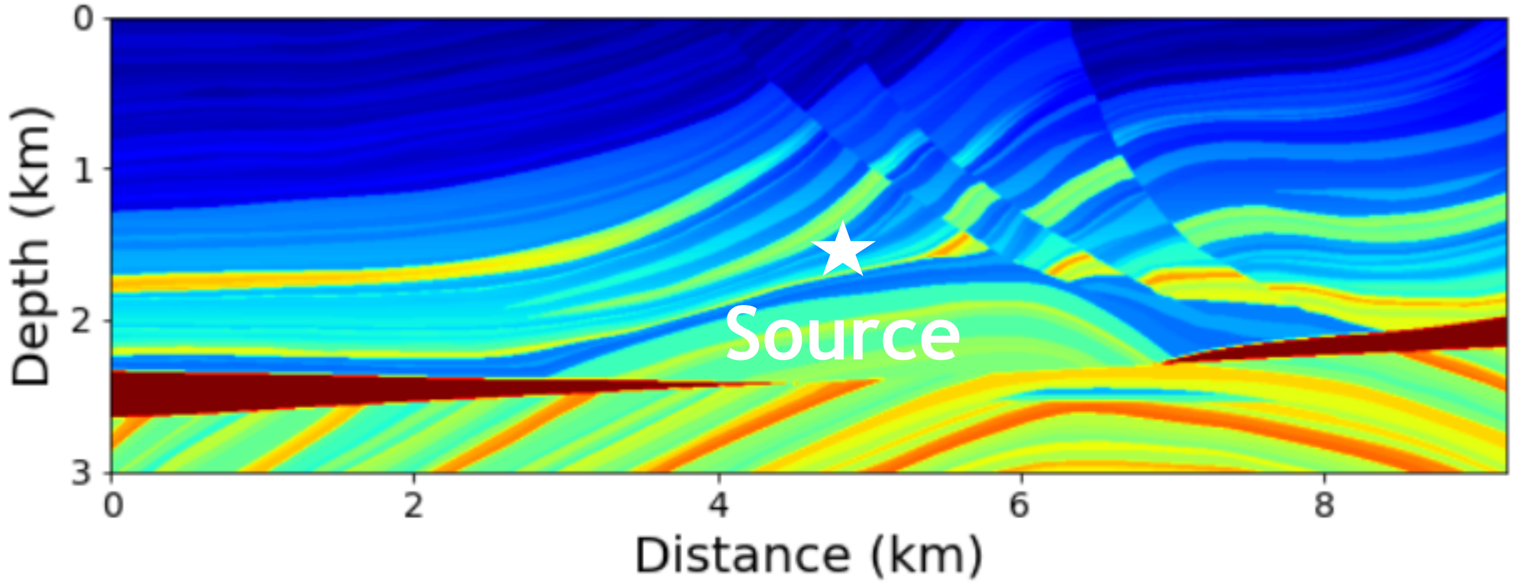
- ➢ Motivation
- ➢ Theory
- ➢ Experimental setup
- ❖ Numerical examples
- ❖ Conclusion and next steps

EXPERIMENT SETUP



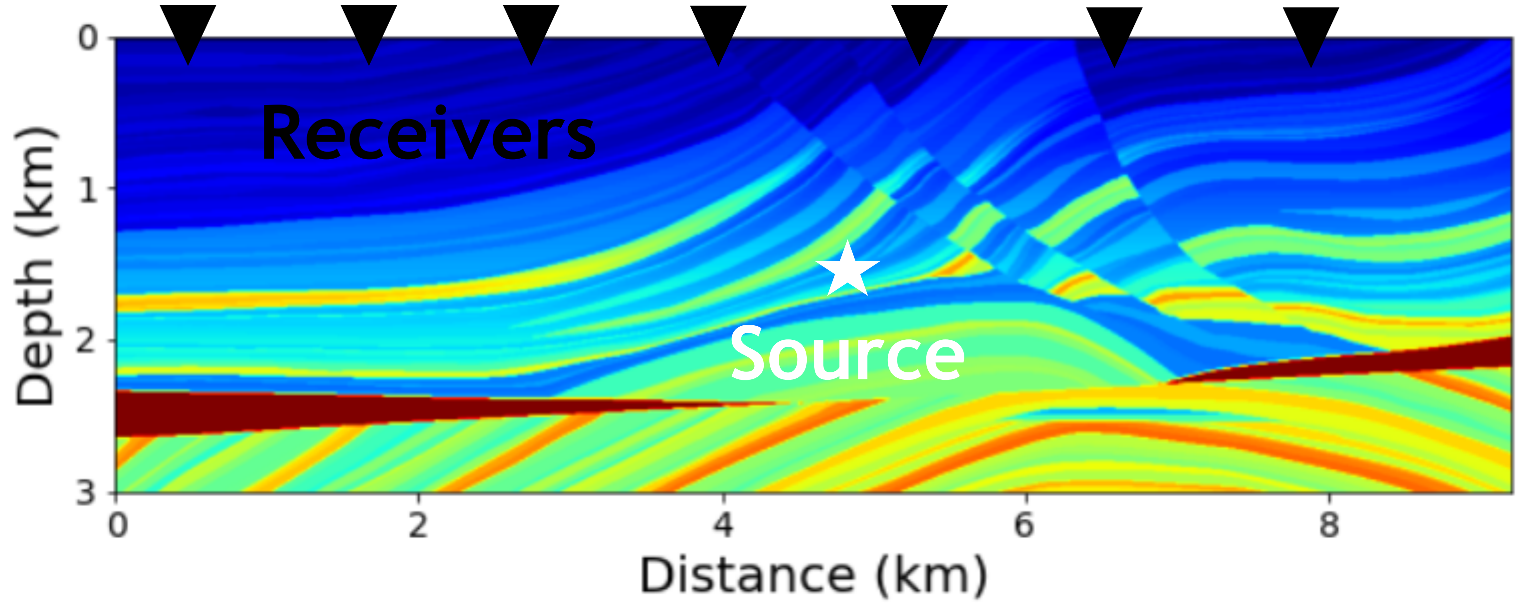
Input velocity model

EXPERIMENT SETUP



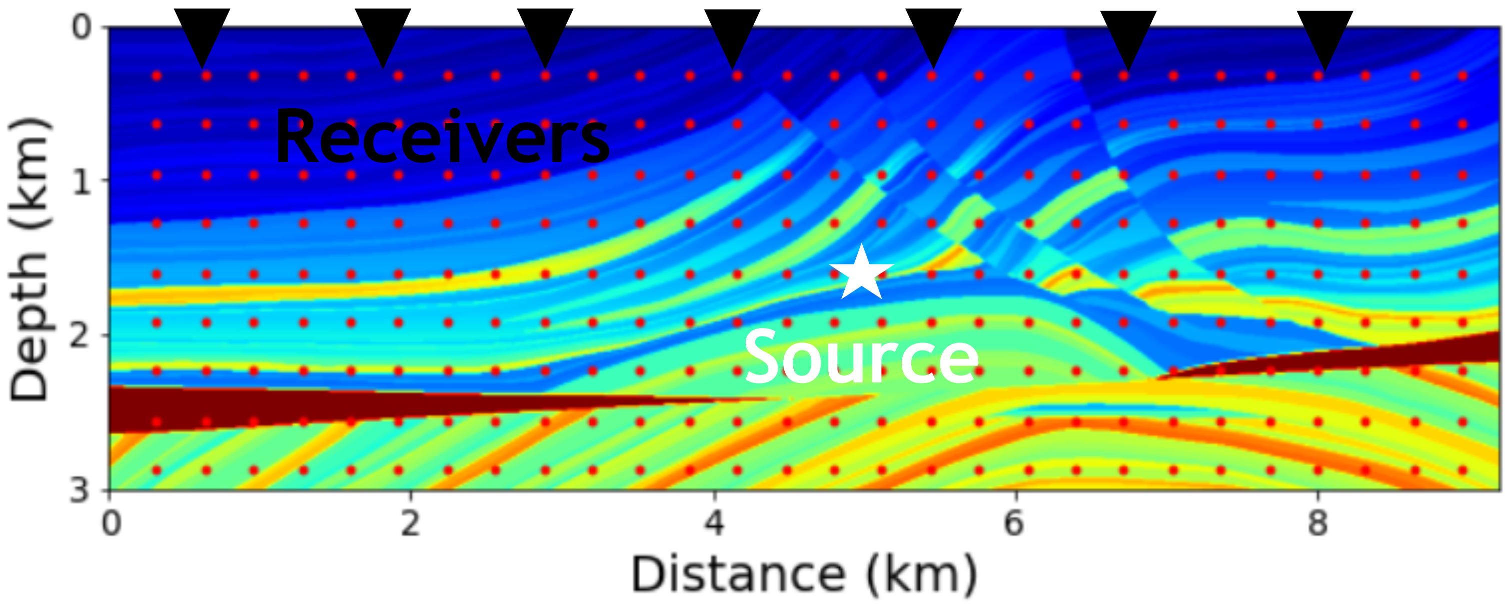
Input velocity model

EXPERIMENT SETUP

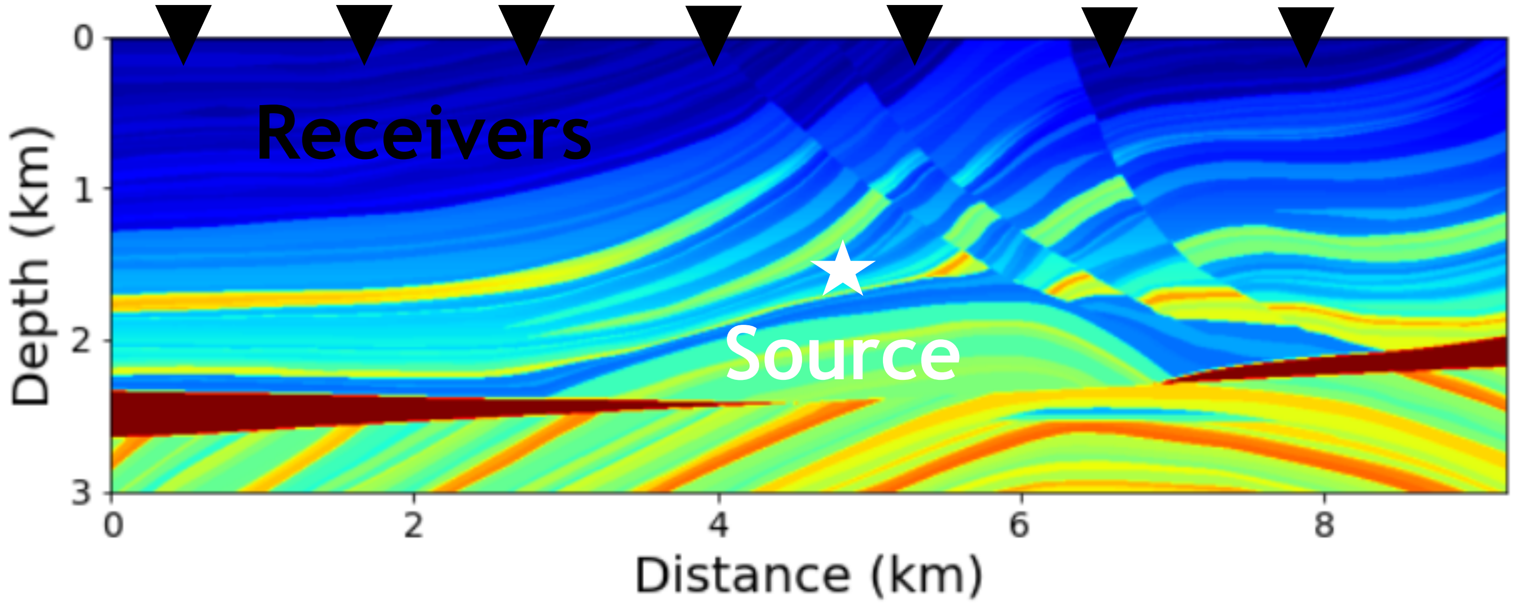


Input velocity model

EXPERIMENT SETUP

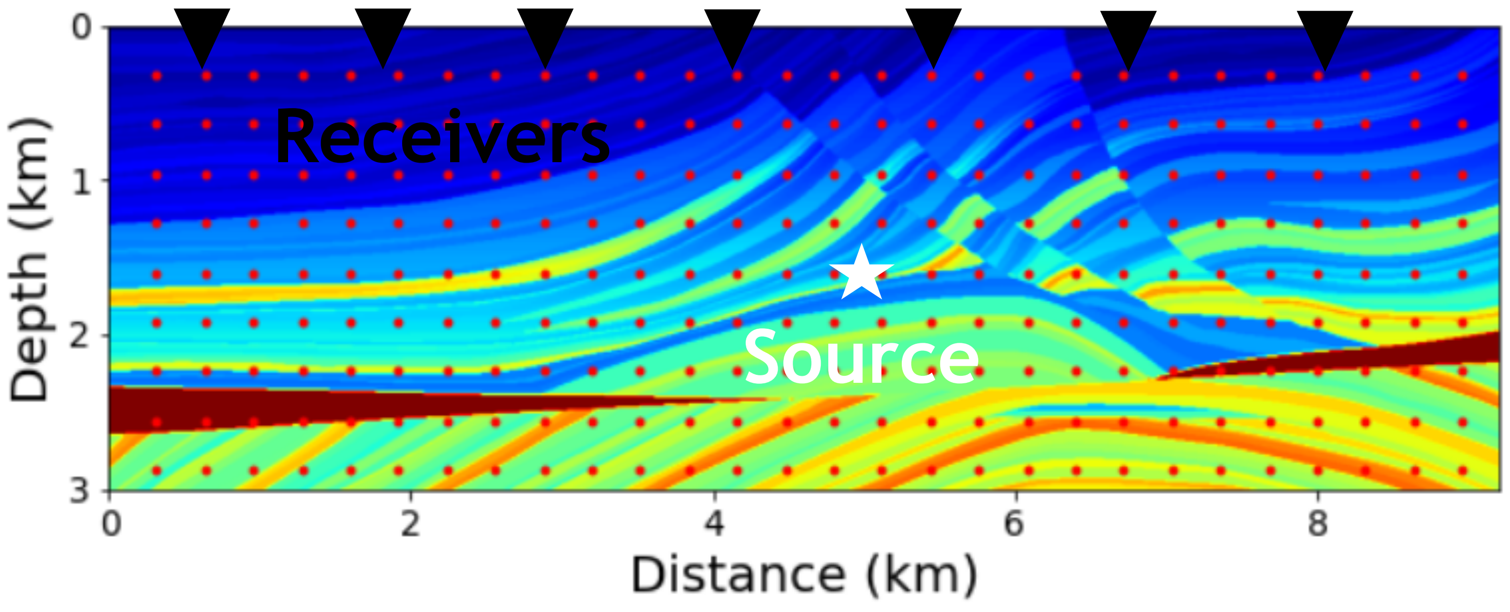


Point cloud representation

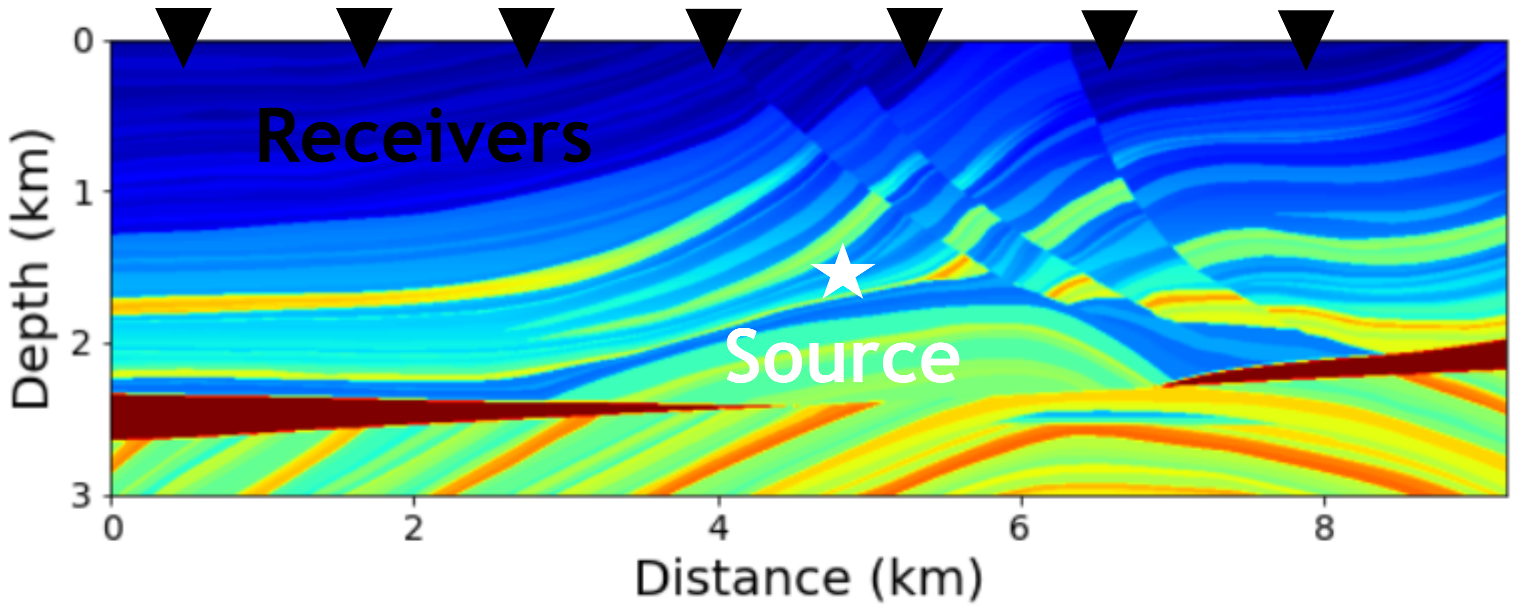


Input velocity model

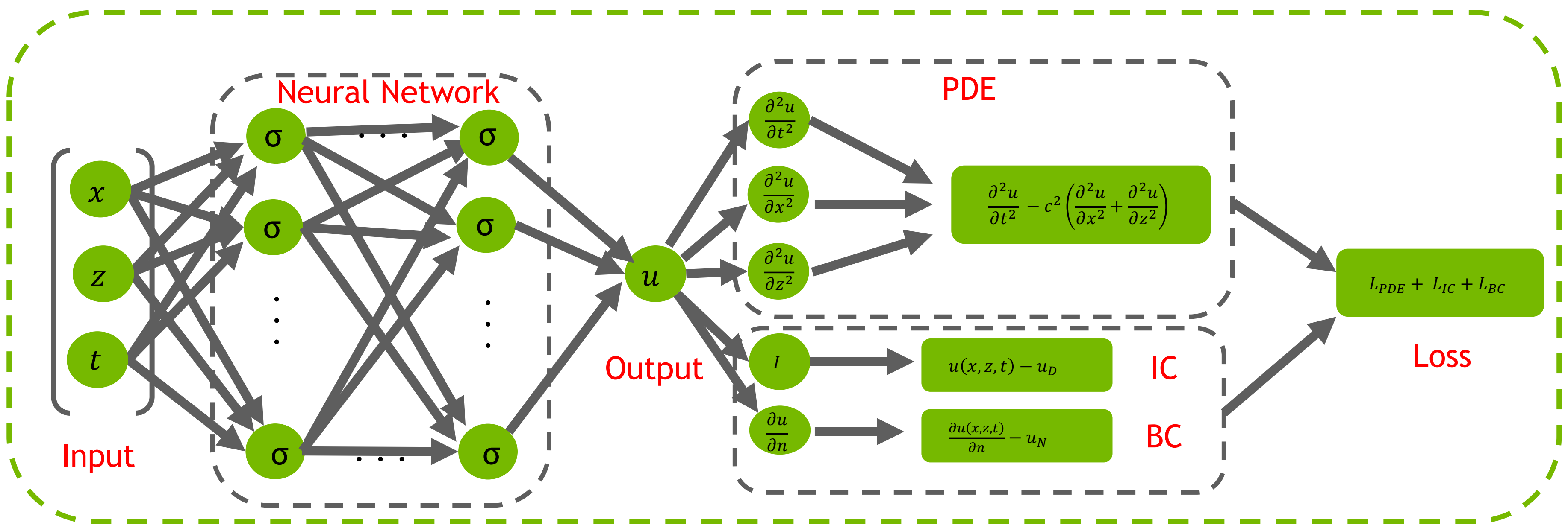
EXPERIMENT SETUP



Point cloud representation

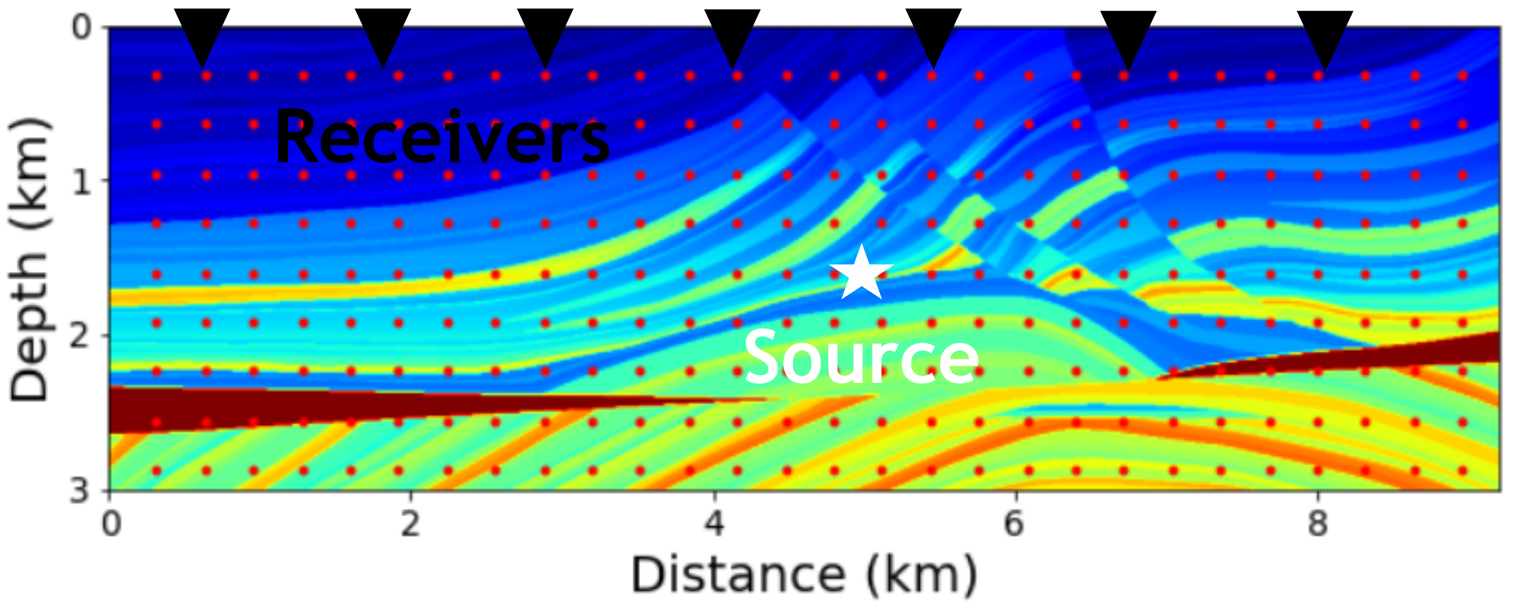


Input velocity model

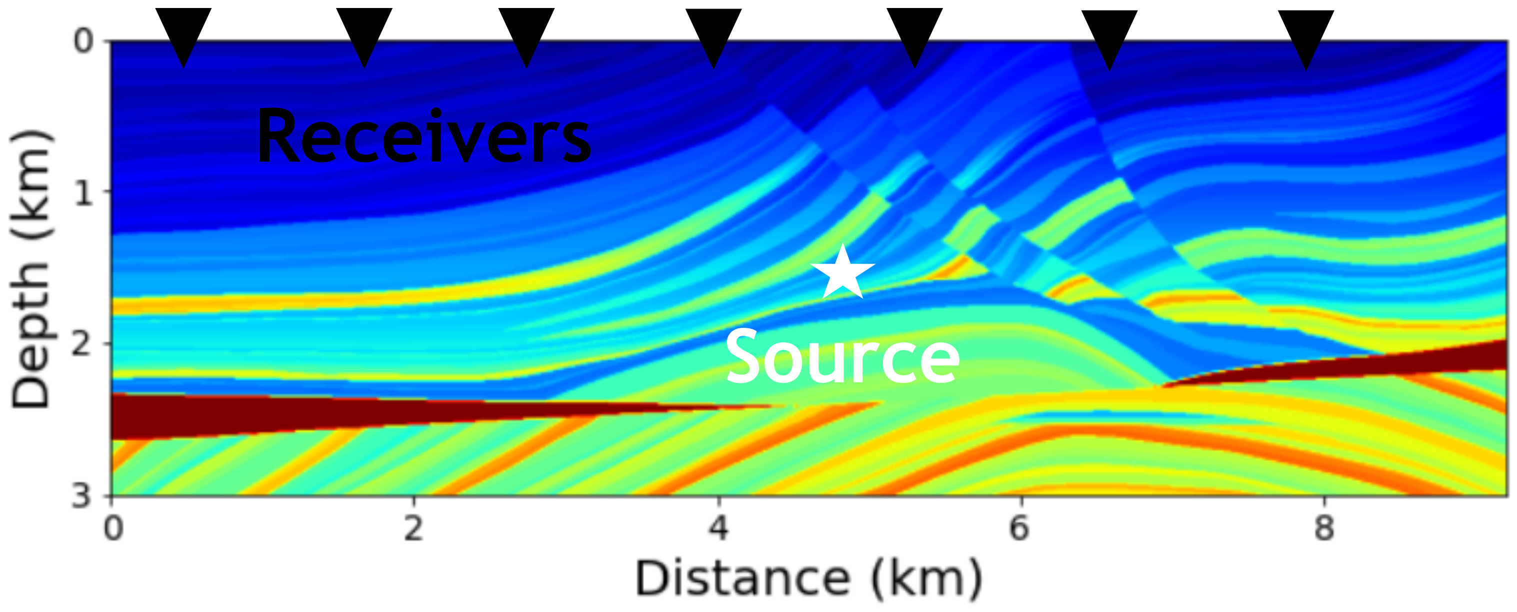


PINN framework

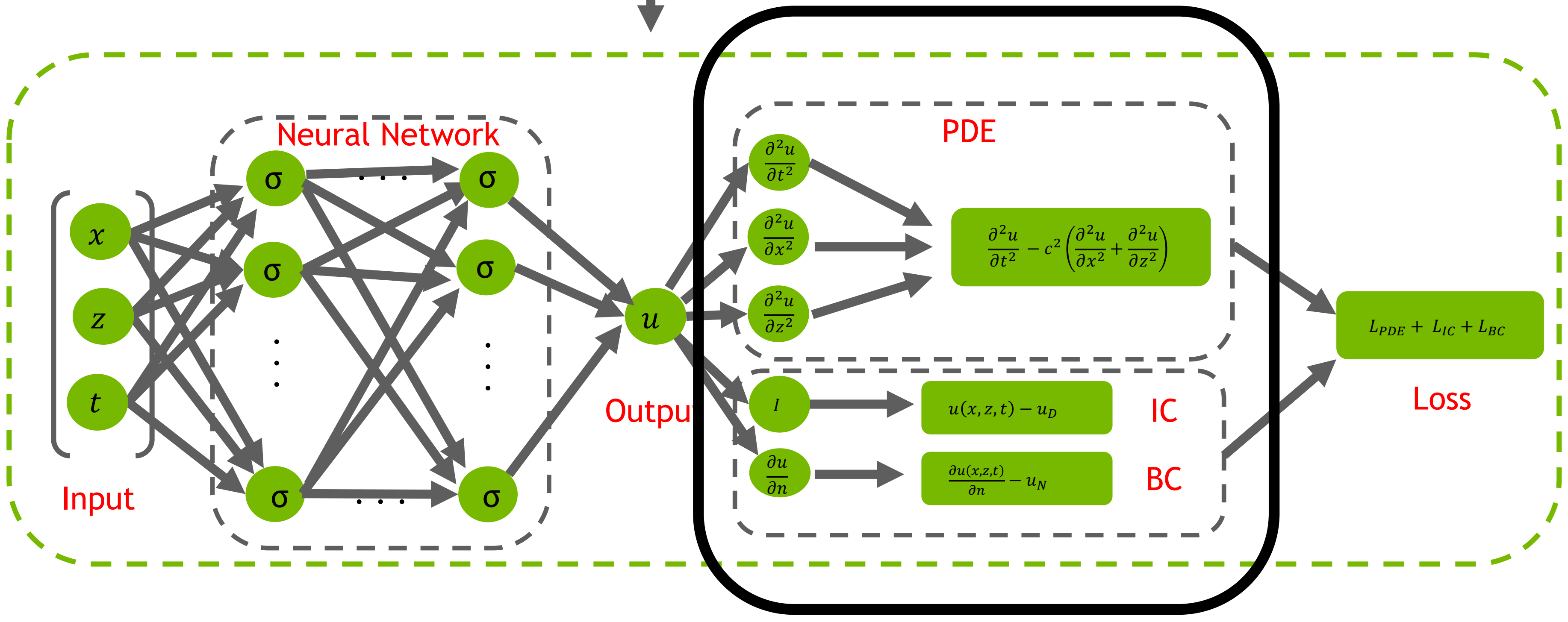
EXPERIMENT SETUP



Point cloud representation

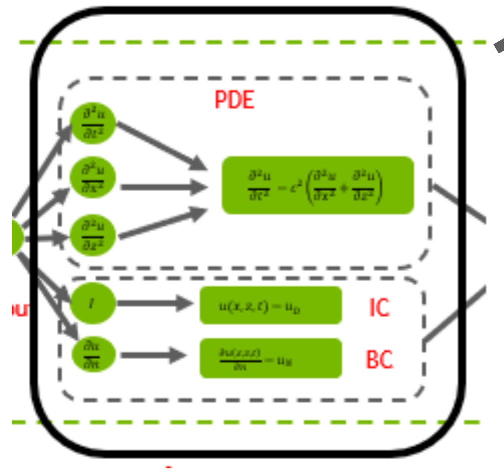


Input velocity model



PINN framework

EXPERIMENT SETUP



PDE : $$\frac{\partial^2 u}{\partial t^2} - c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2}\right) = f(x,z,t)$$

u : wavefield solution

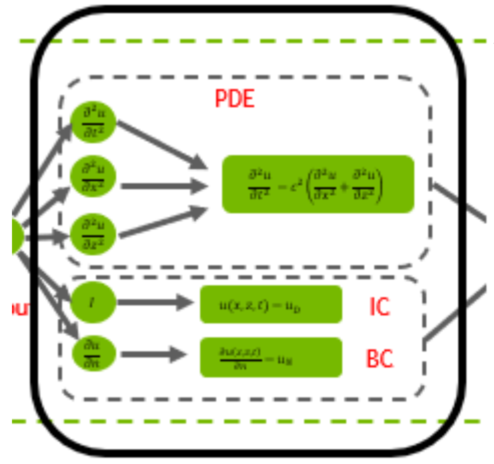
c : velocity

$f(x,z,t)$: source

IC : $$u(x,z,t=0) = 0$$
$$\frac{\partial u(x,z,t=0)}{\partial t} = 0$$

BC : Absorbing boundary conditions (ABC)

EXPERIMENT SETUP



PDE : $$\frac{\partial^2 u}{\partial t^2} - c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2}\right) = f(x,z,t)$$

u : wavefield solution

c : velocity

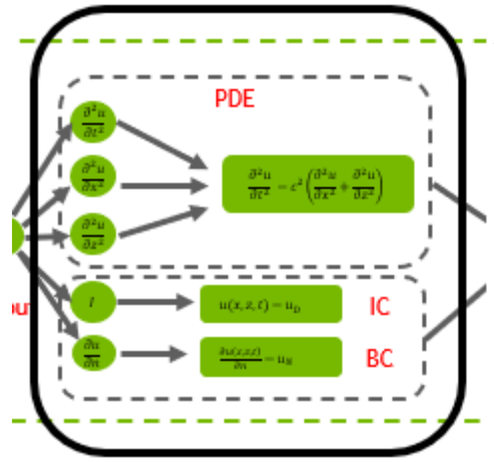
$$f(x,z,t) = \boldsymbol{r(t)} * s(x,z)$$

IC : $$u(x,z,t=0) = 0$$
$$\frac{\partial u(x,z,t=0)}{\partial t} = 0$$

BC : Absorbing boundary conditions (ABC)

NVIDIA

EXPERIMENT SETUP



PDE : $$\frac{\partial^2 u}{\partial t^2} - c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2}\right) = f(x,z,t)$$

u : wavefield solution

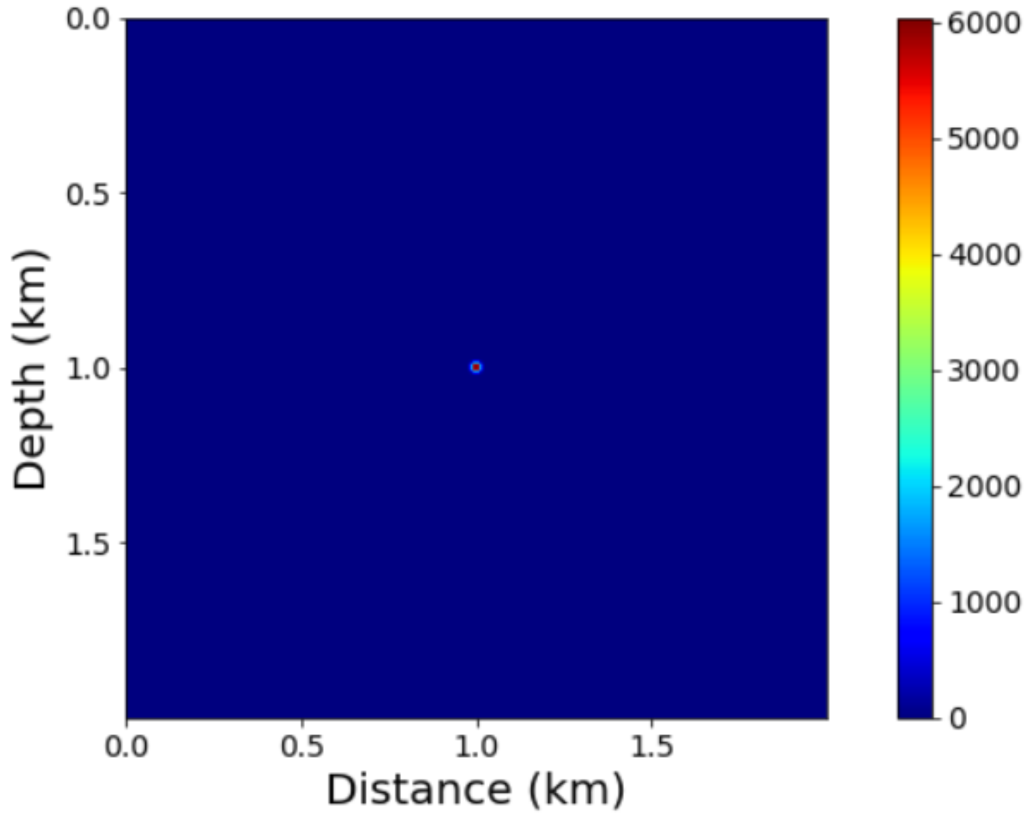
c : velocity

$$f(x,z,t) = r(t) * \boldsymbol{s(x,z)}$$

IC : $$u(x,z,t=0) = 0$$
$$\frac{\partial u(x,z,t=0)}{\partial t} = 0$$

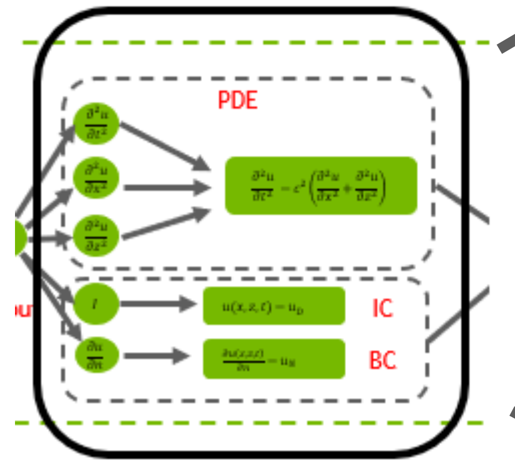
BC : Absorbing boundary conditions (ABC)

$\boldsymbol{s(x,z)}$:



Gaussian spatial function

EXPERIMENT SETUP



PDE : $$\frac{\partial^2 u}{\partial t^2} - c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2}\right) = f(x,z,t)$$

u : wavefield solution

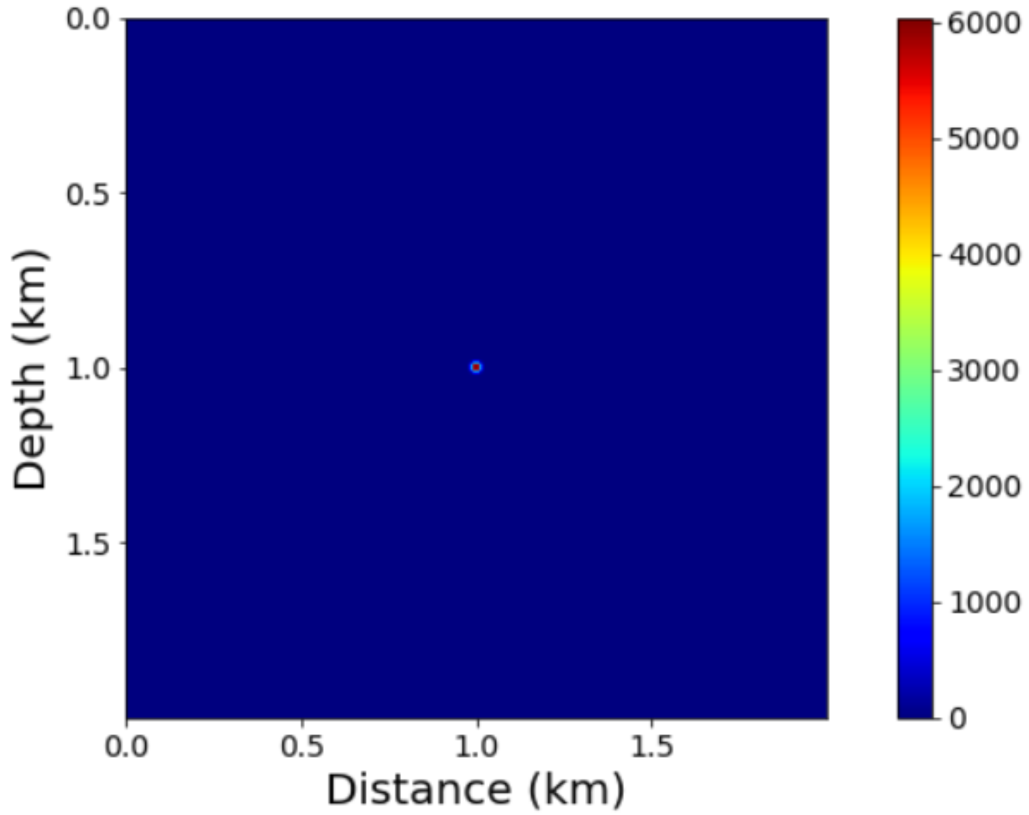
c : velocity

IC : $$u(x,z,t=0) = 0$$
$$\frac{\partial u(x,z,t=0)}{\partial t} = 0$$

BC : Absorbing boundary conditions (ABC)

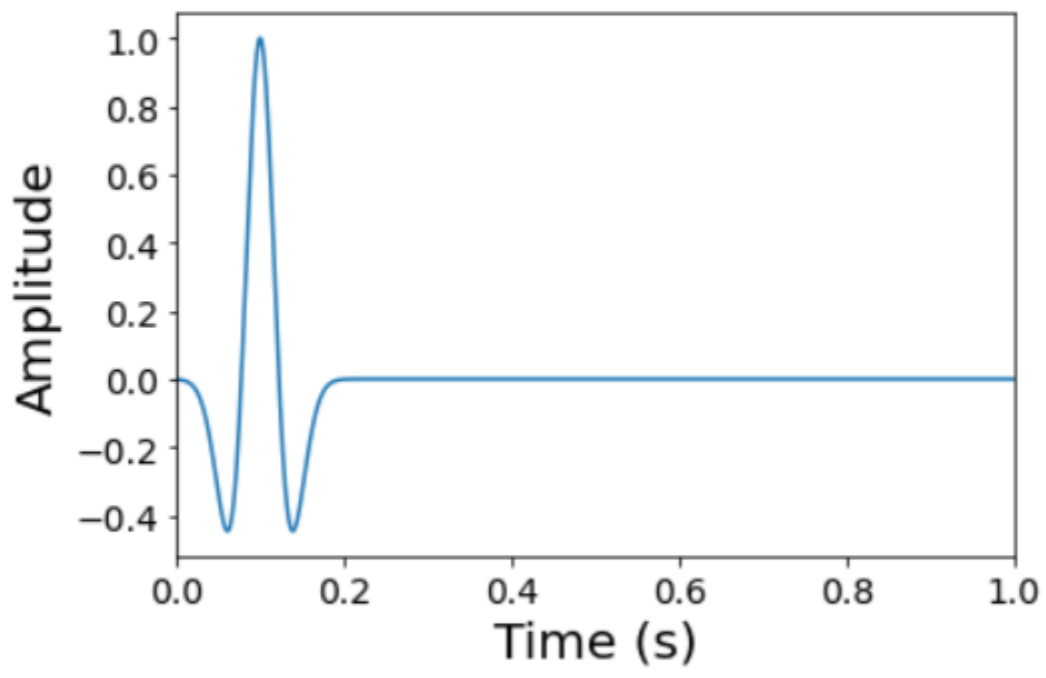
$$f(x,z,t) = \boldsymbol{r(t)} * s(x,z)$$

$s(x,z)$:



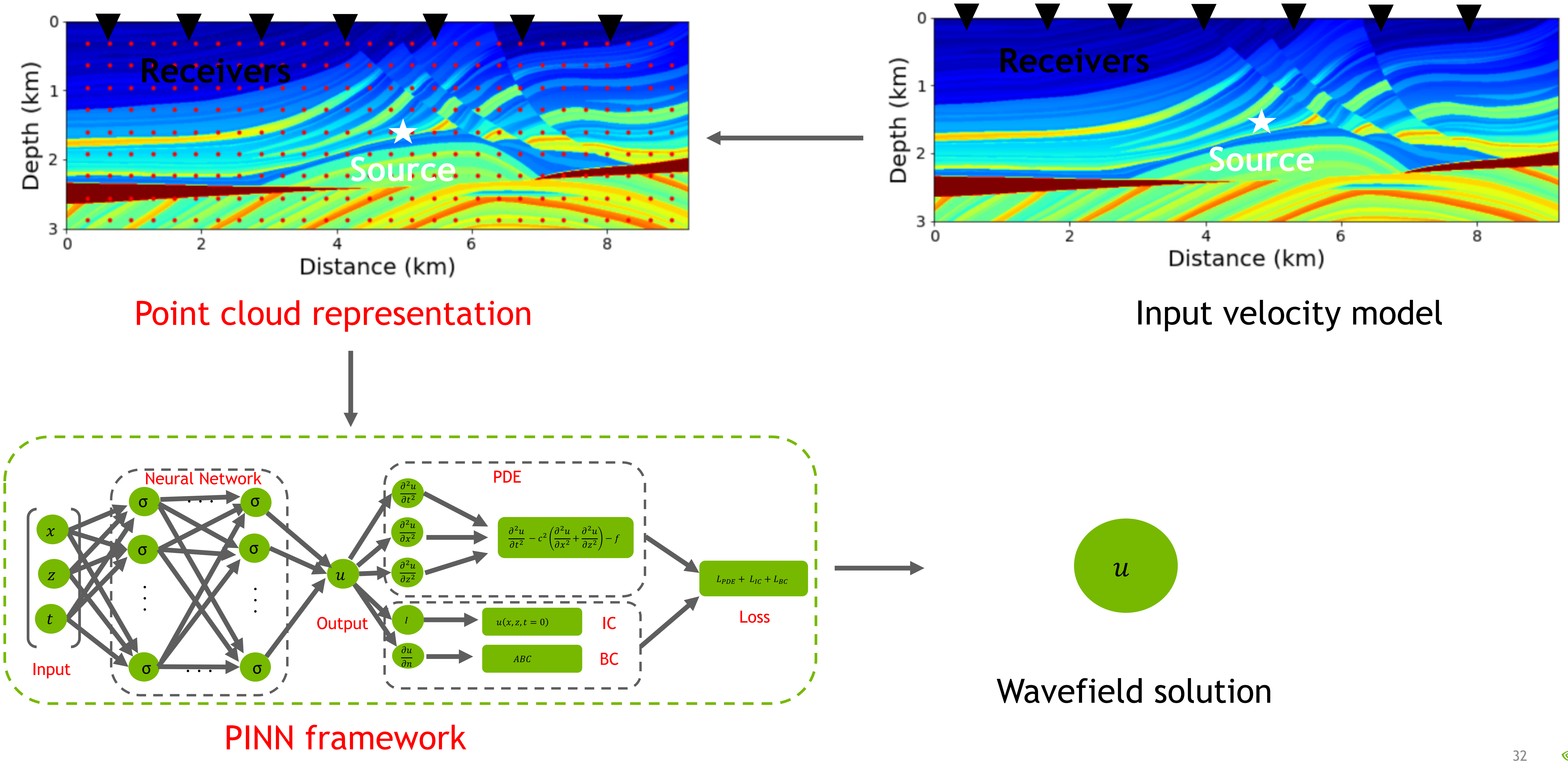
Gaussian spatial function

$\boldsymbol{r(t)}$:

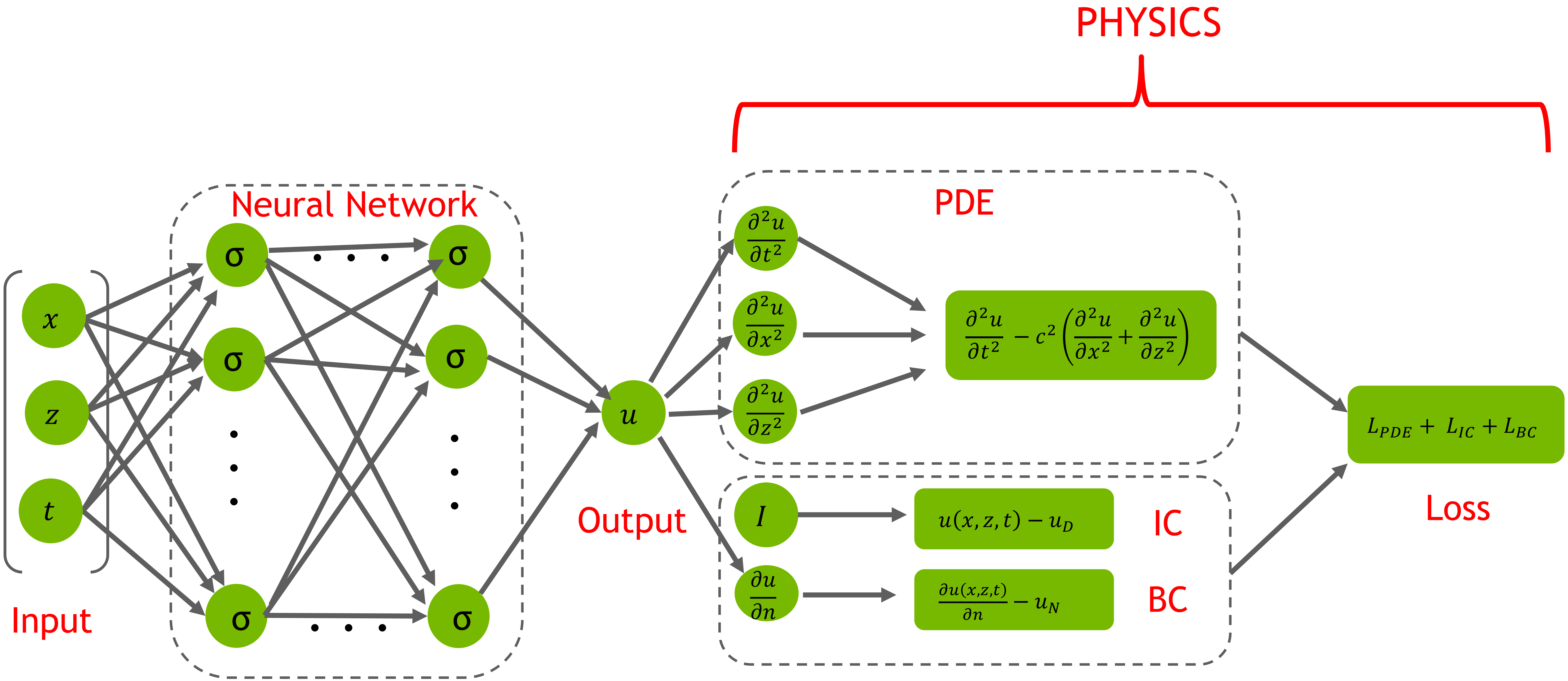


Ricker wavelet

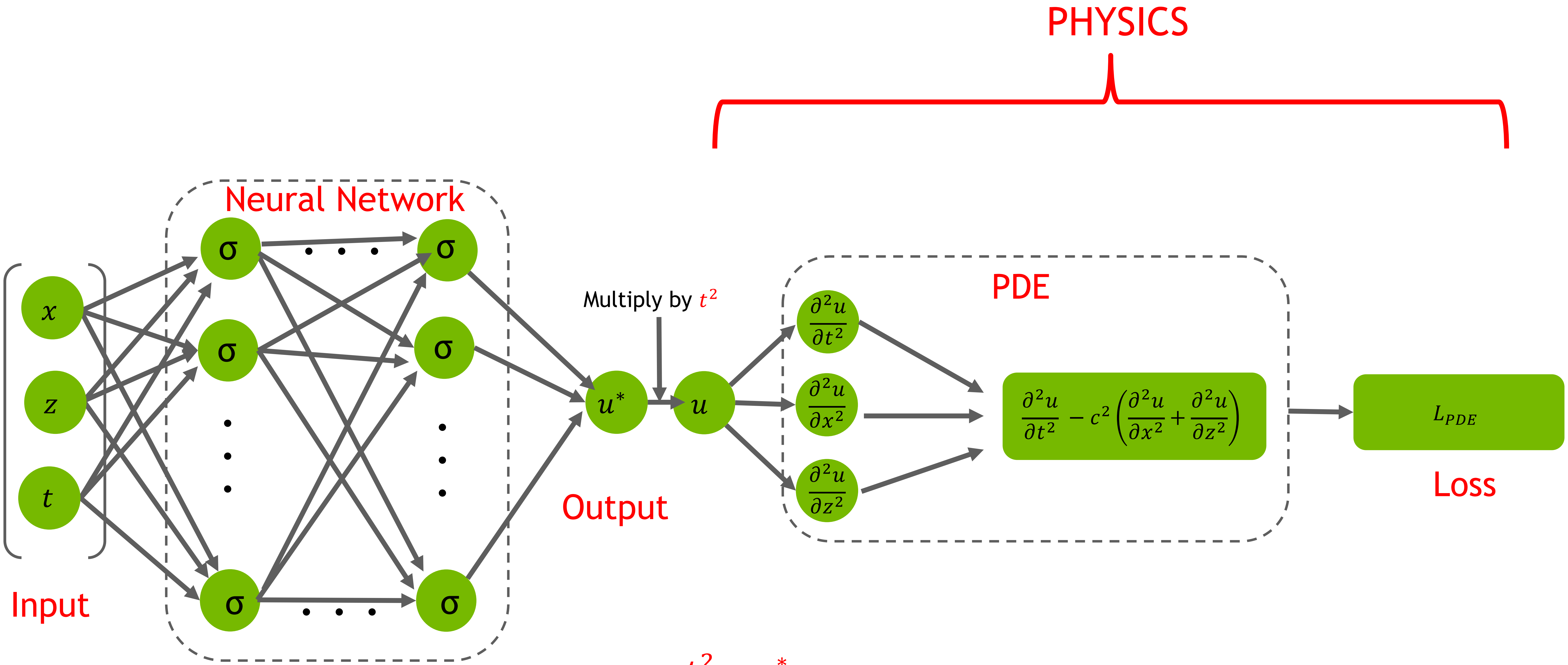
EXPERIMENT SETUP



ENFORCEMENT OF INITIAL CONDITIONS (IC)



ENFORCEMENT OF INITIAL CONDITIONS (IC)



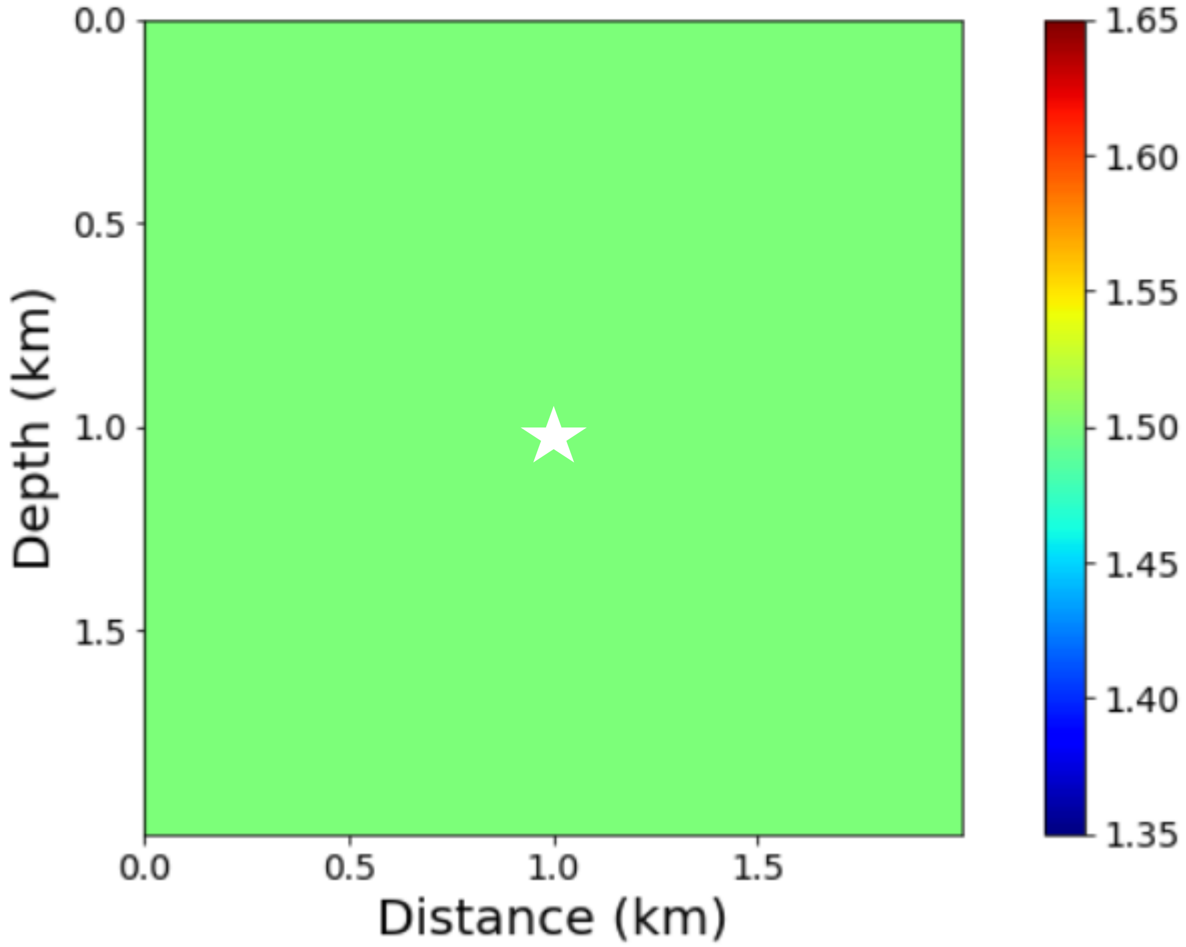
$$u = t^2 * u^*$$

Multiply by t^2 to the Neural Network output u^*

AGENDA

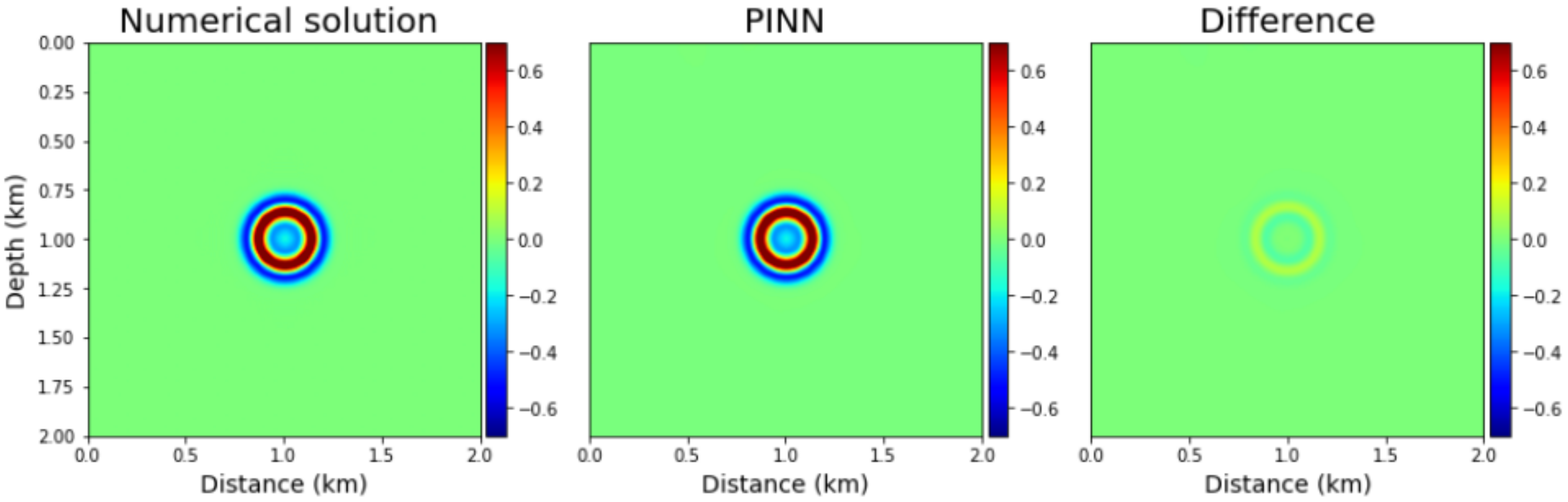
- ➢ Motivation
- ➢ Theory
- ➢ Experimental setup
- ➢ Numerical examples
- ❖ Conclusion and next steps

EXPERIMENT-1: HOMOGENEOUS MODEL



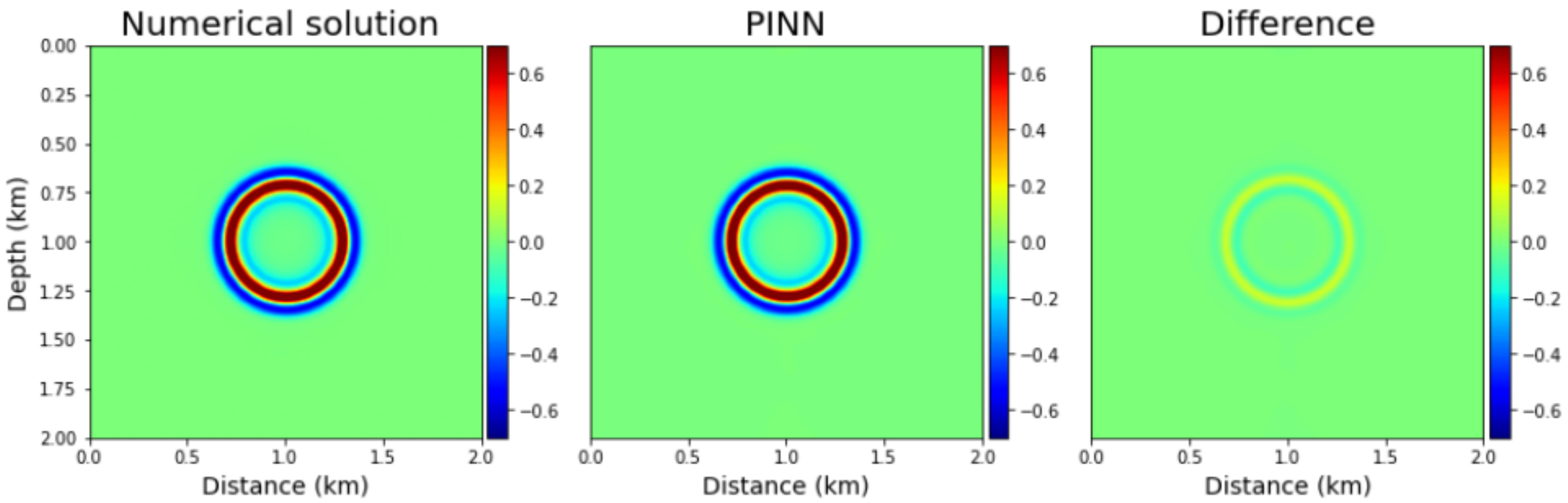
- Ricker wavelet with peak frequency 20 Hz
- Fourier neural network with 20 frequencies

EXPERIMENT-1: HOMOGENEOUS MODEL



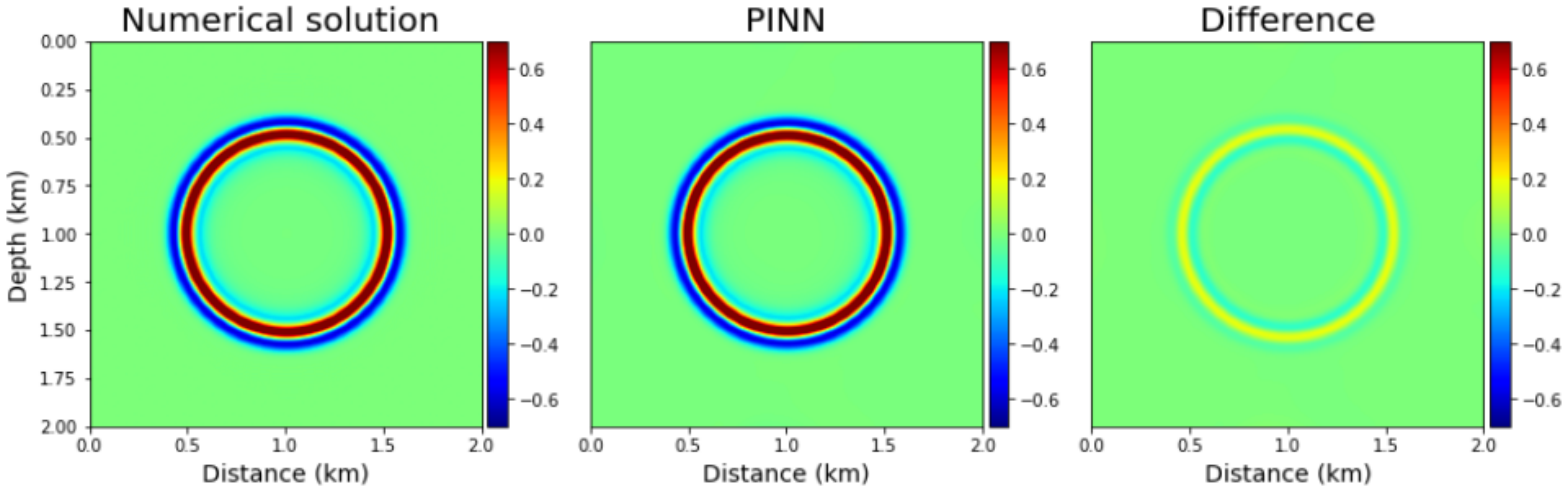
$t = 0.1\ s$

EXPERIMENT-1: HOMOGENEOUS MODEL



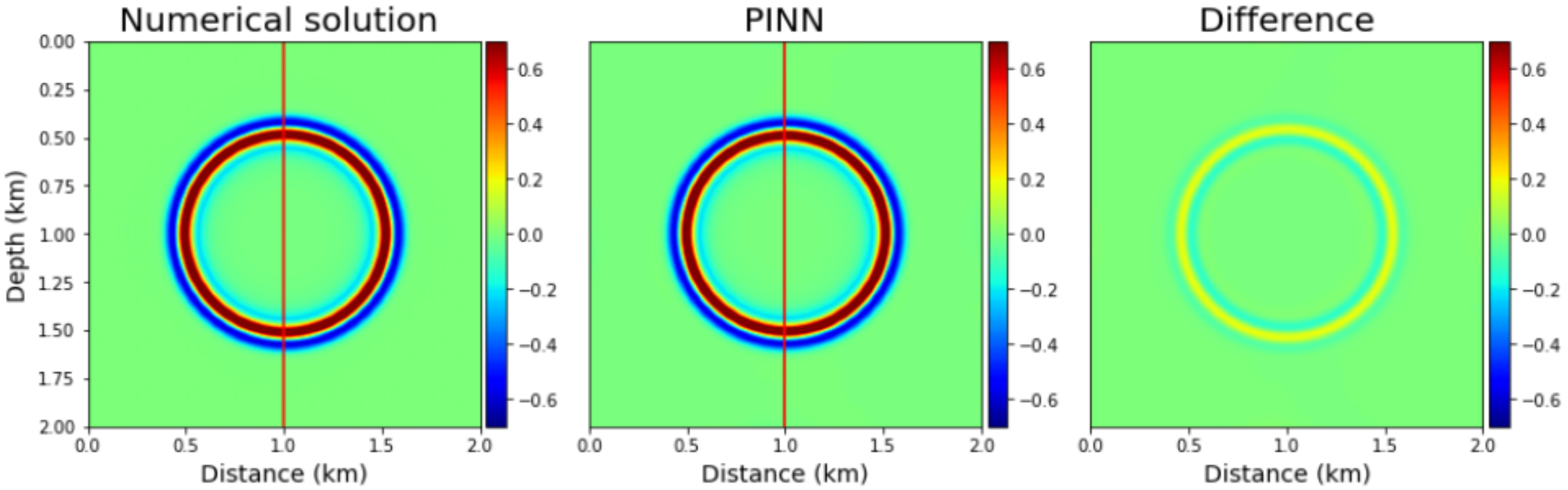
$t = 0.2\ s$

EXPERIMENT-1: HOMOGENEOUS MODEL



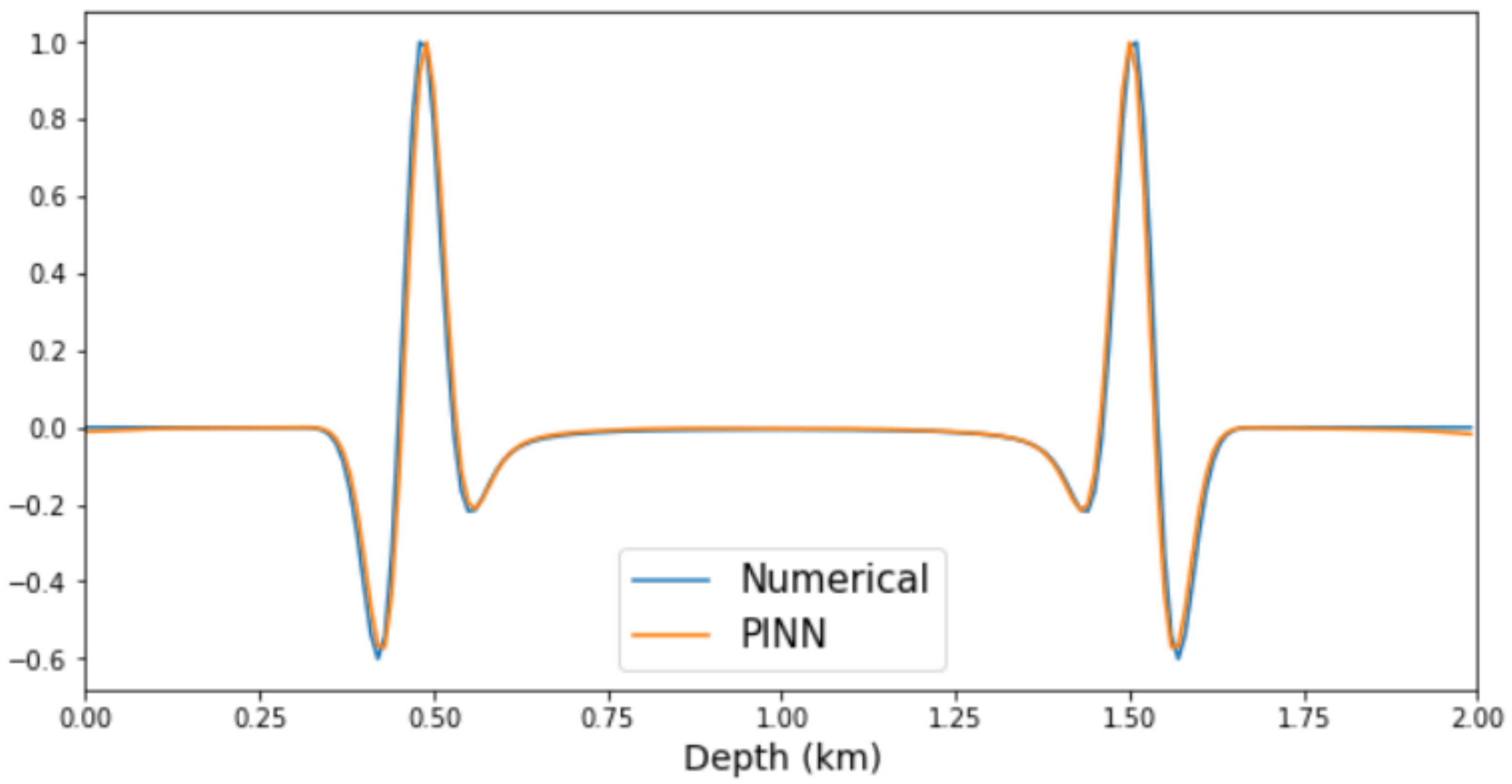
$$t = 0.4\ s$$

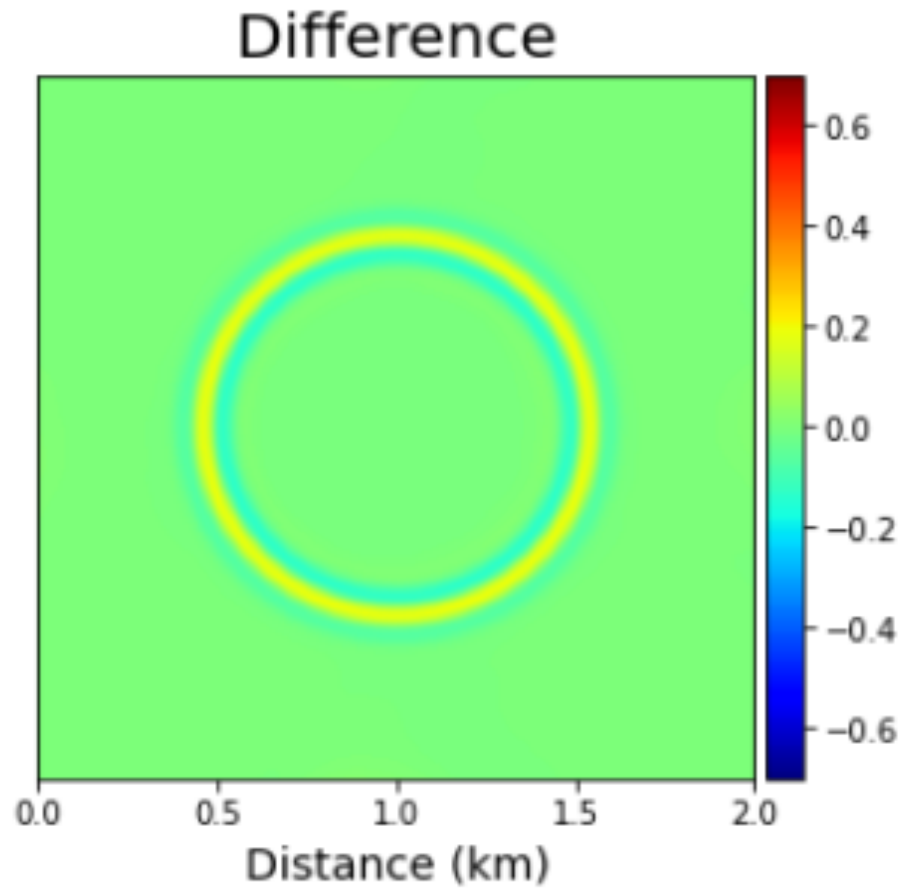
EXPERIMENT-1: HOMOGENEOUS MODEL



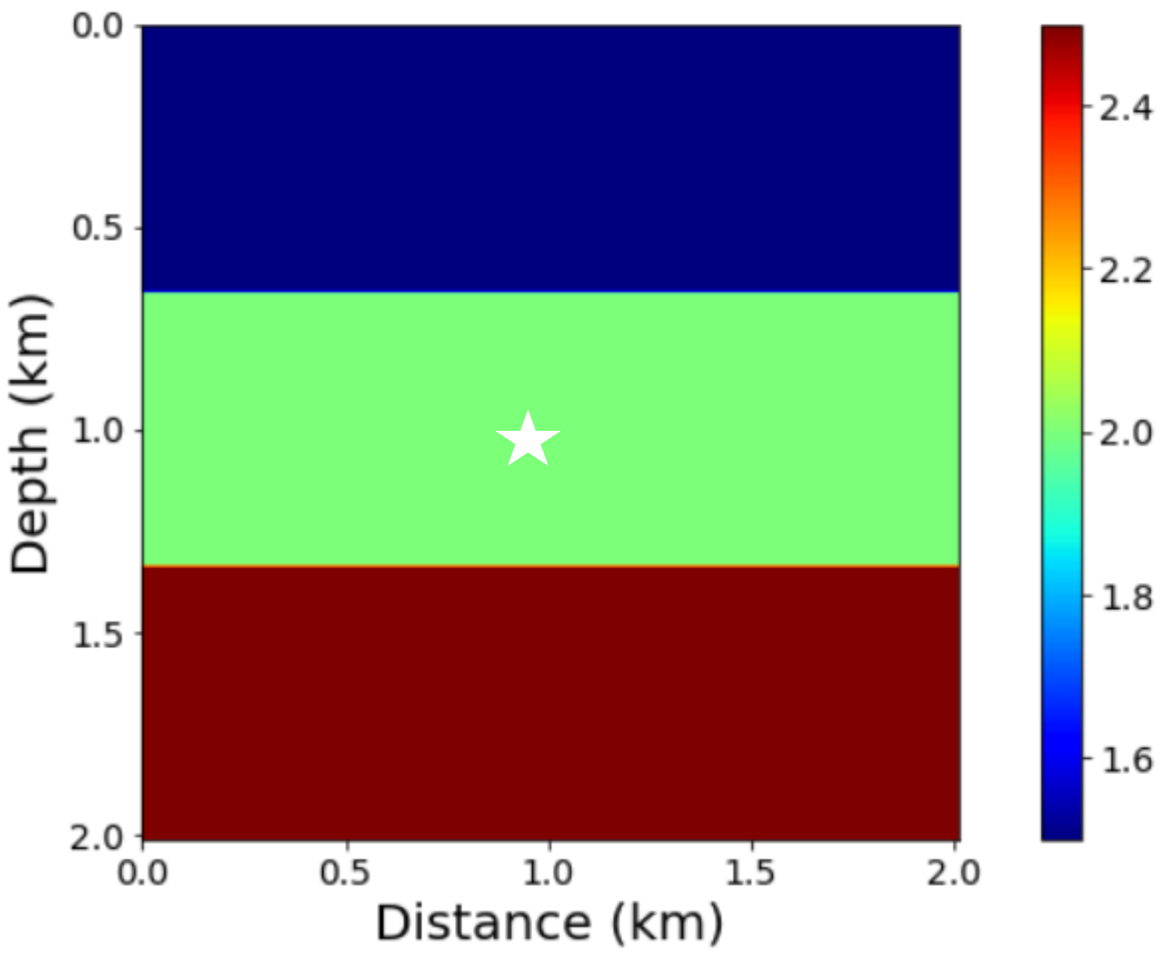
$t = 0.4\ s$

EXPERIMENT-1: HOMOGENEOUS MODEL



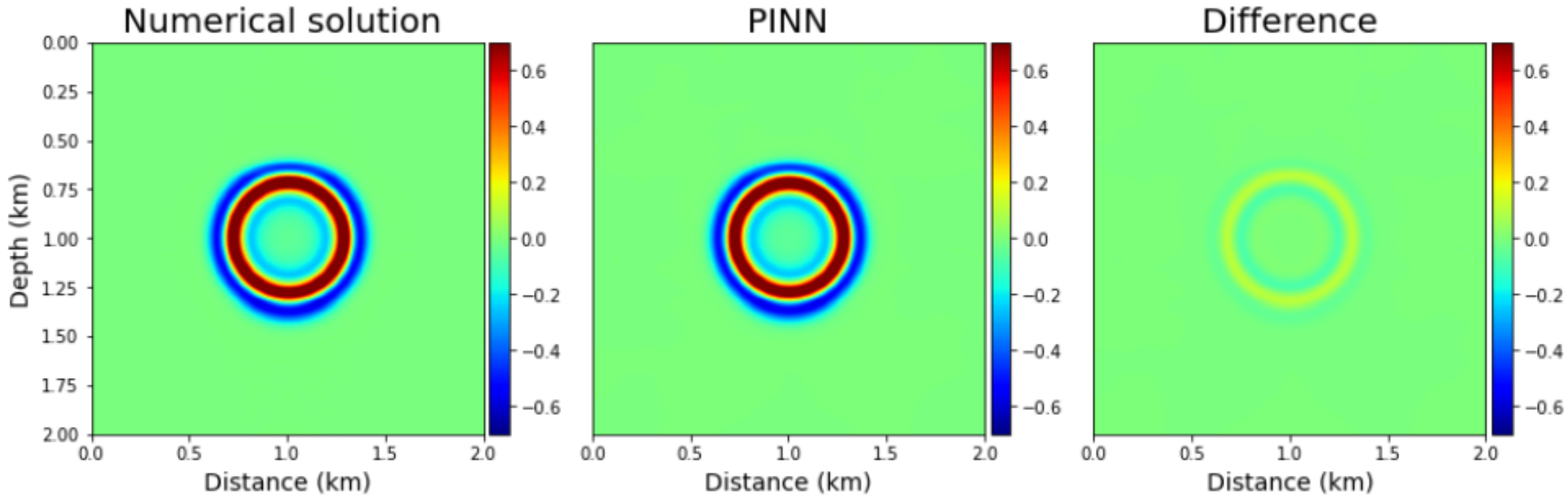


EXPERIMENT-2: LAYERED MODEL



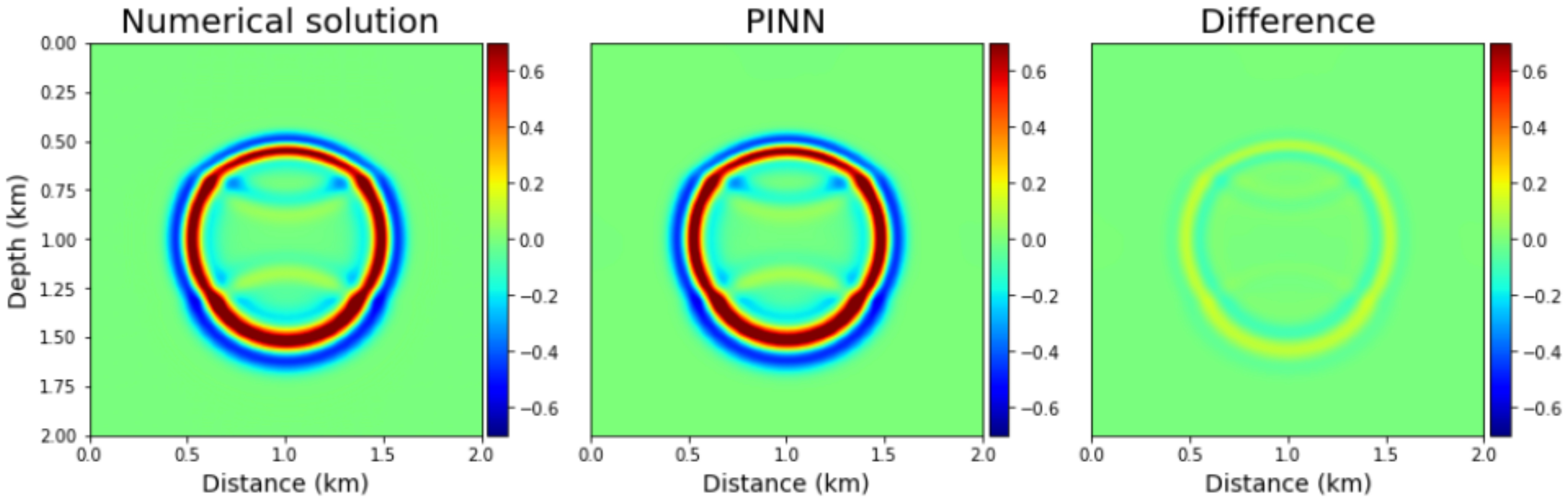
- Ricker wavelet with peak frequency 20 Hz
- Fourier neural network with 20 frequencies

EXPERIMENT-2: LAYERED MODEL



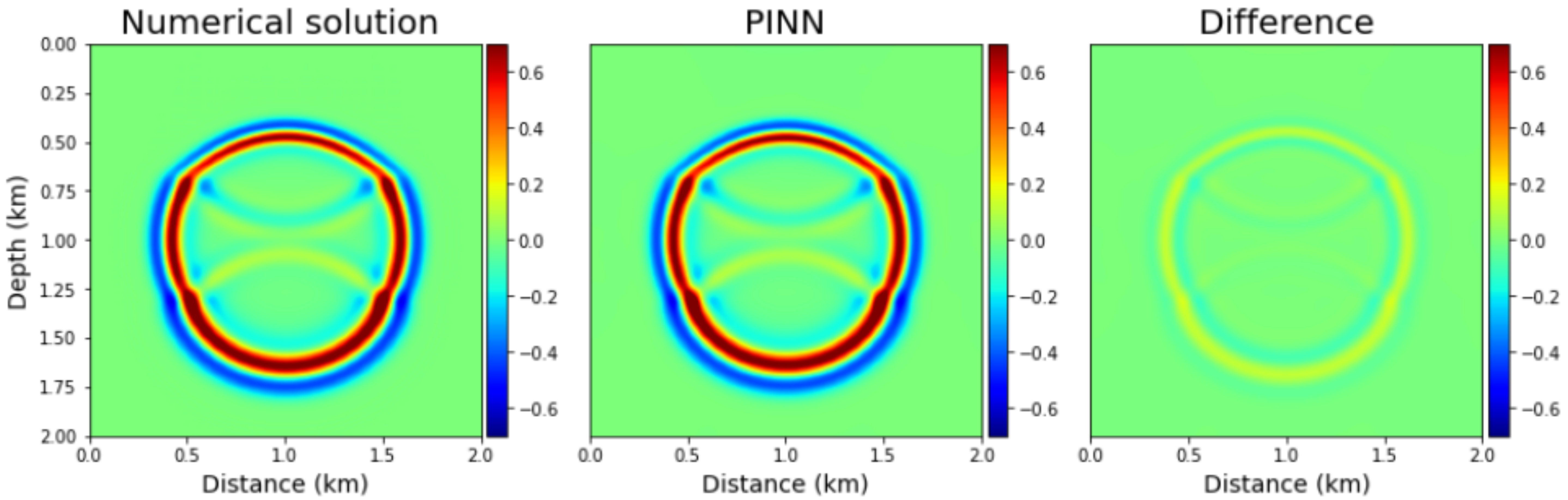
$t = 0.1\ s$

EXPERIMENT-2: LAYERED MODEL



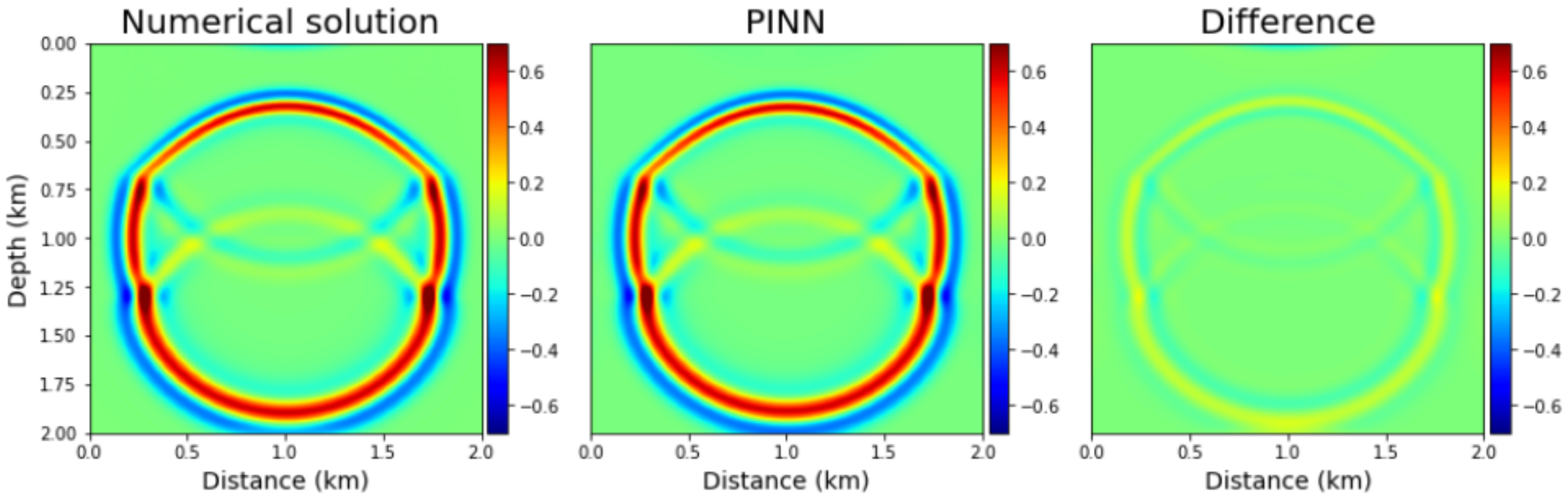
$t = 0.2\ s$

EXPERIMENT-2: LAYERED MODEL



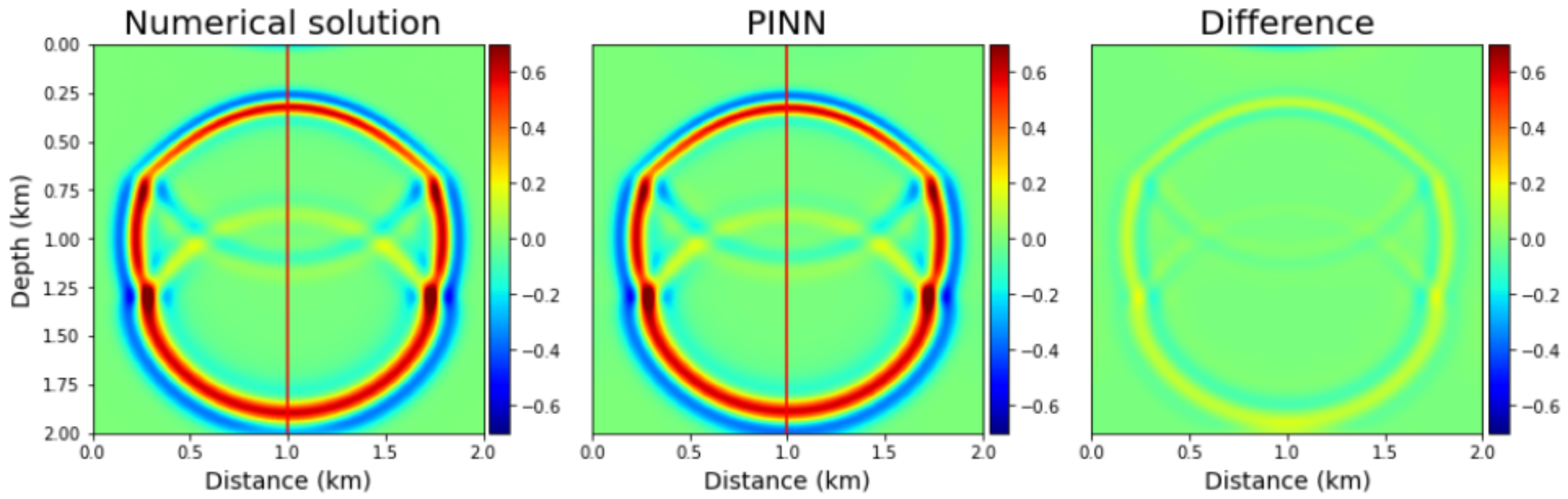
$t = 0.3\ s$

EXPERIMENT-2: LAYERED MODEL



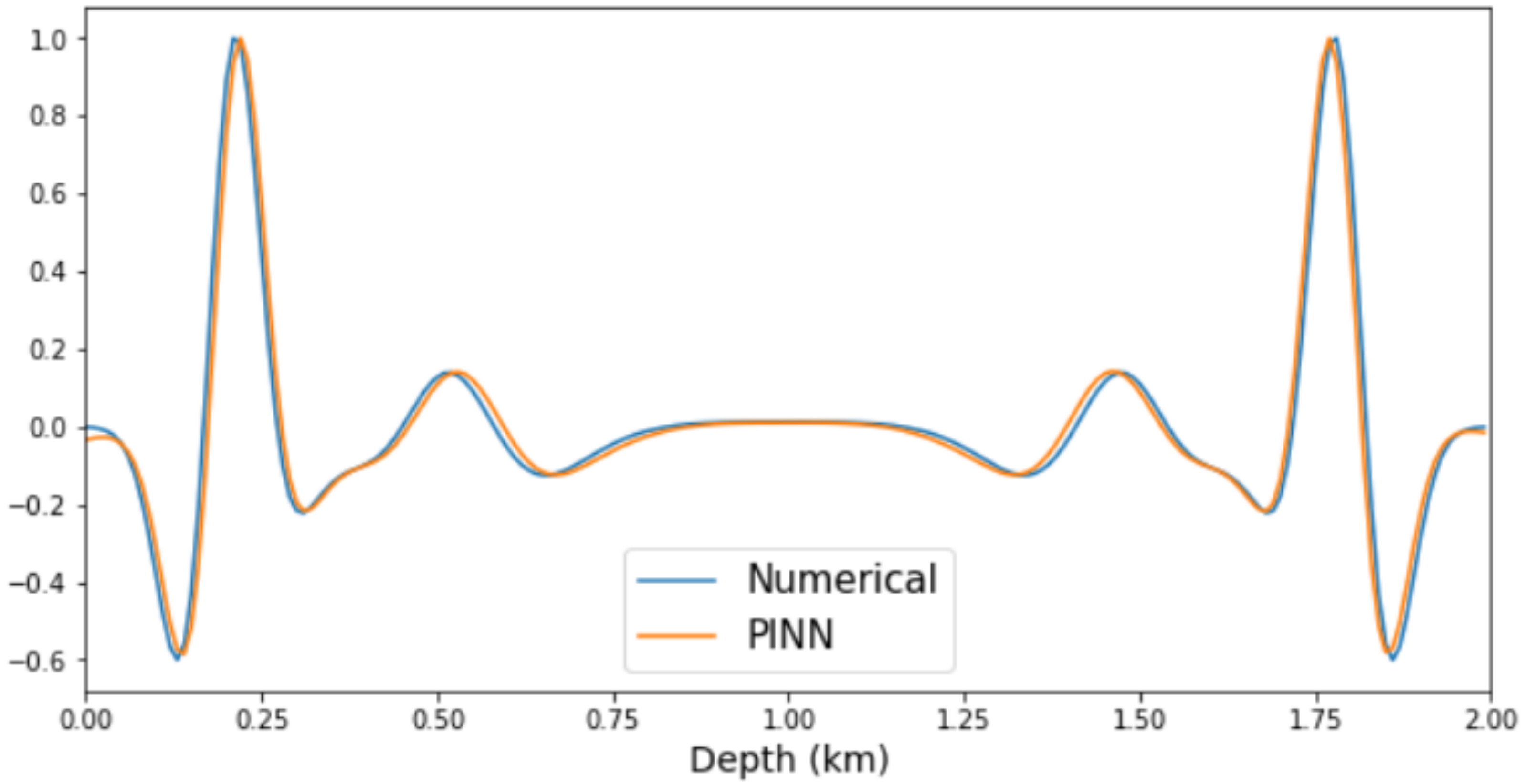
$t = 0.5\ s$

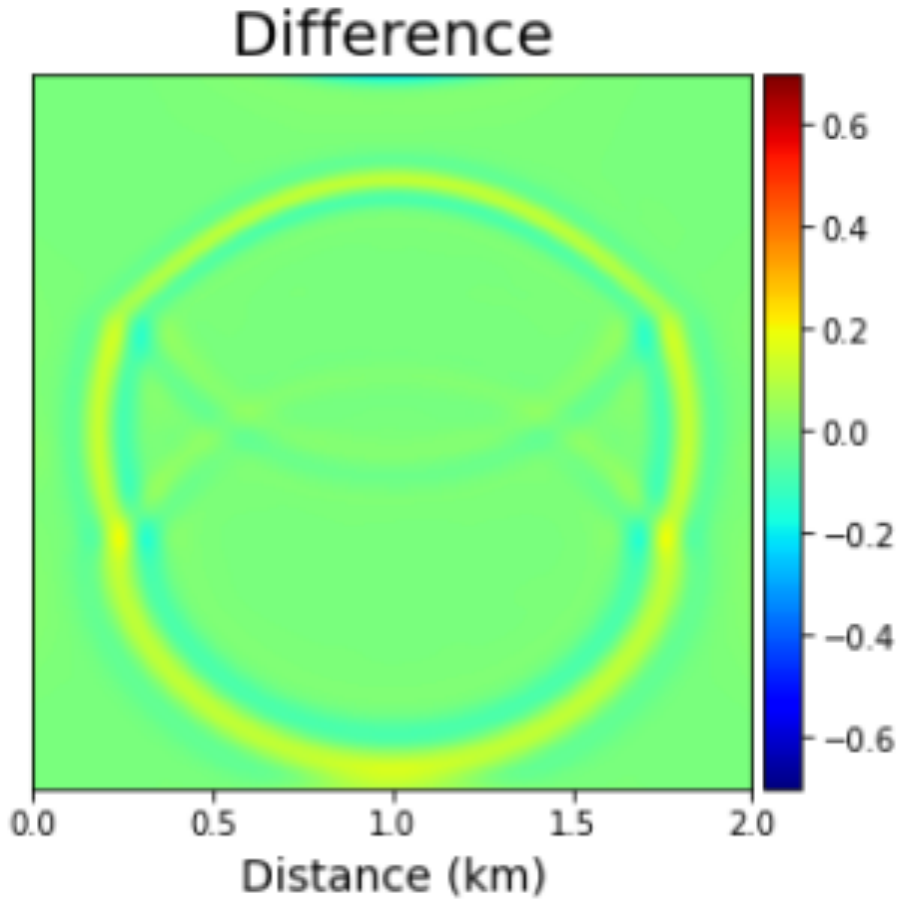
EXPERIMENT-2: LAYERED MODEL



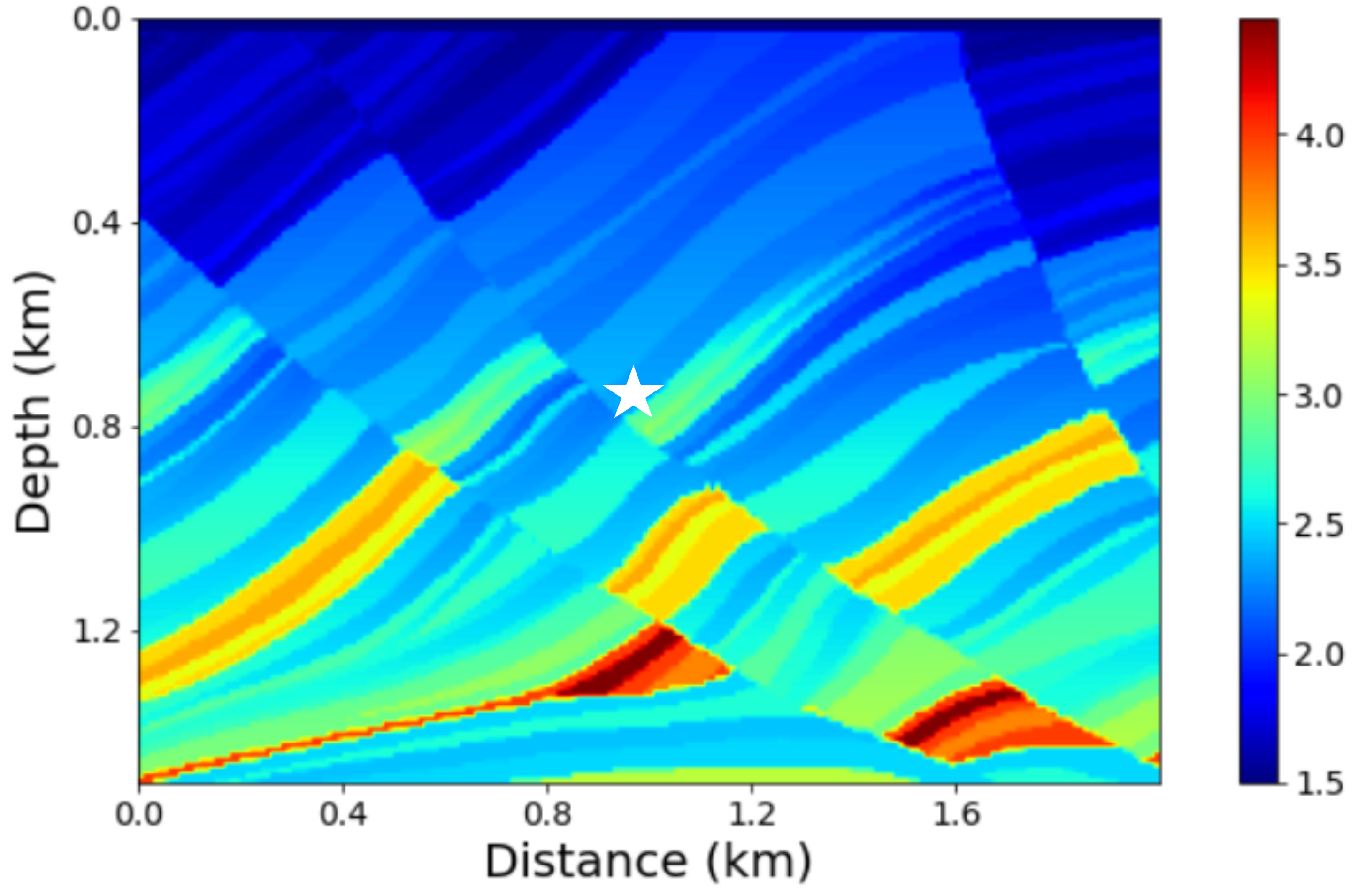
$t = 0.5\ s$

EXPERIMENT-2: LAYERED MODEL



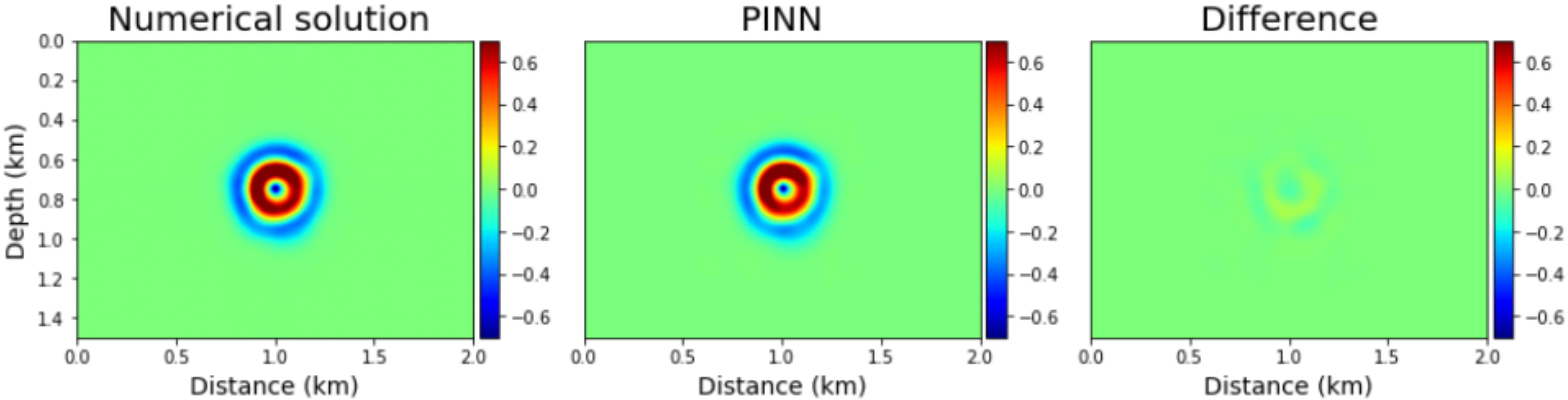


EXPERIMENT-3: MARMOUSI MODEL (MODIFIED)



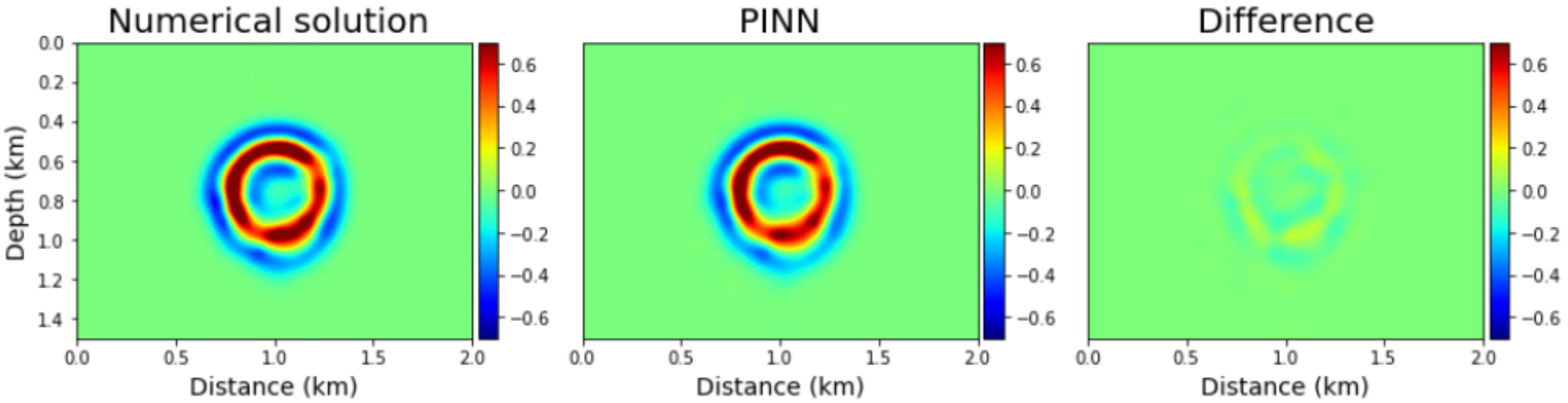
- Ricker wavelet with peak frequency 20 Hz
- Fourier neural network with 50 frequencies

EXPERIMENT-3: MARMOUSI MODEL (MODIFIED)



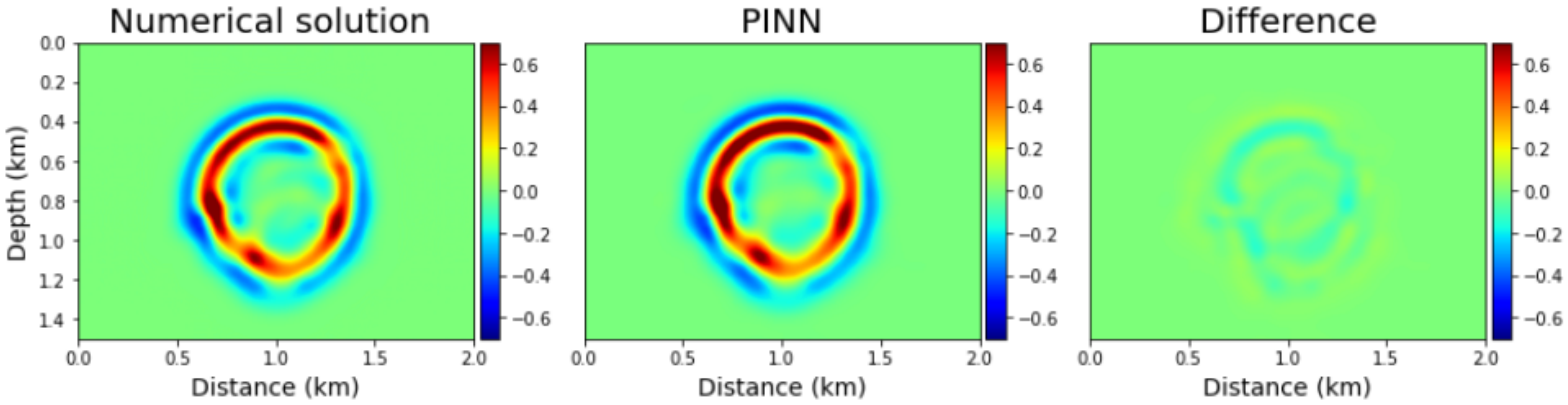
$t = 0.1\ s$

EXPERIMENT-3: MARMOUSI MODEL (MODIFIED)



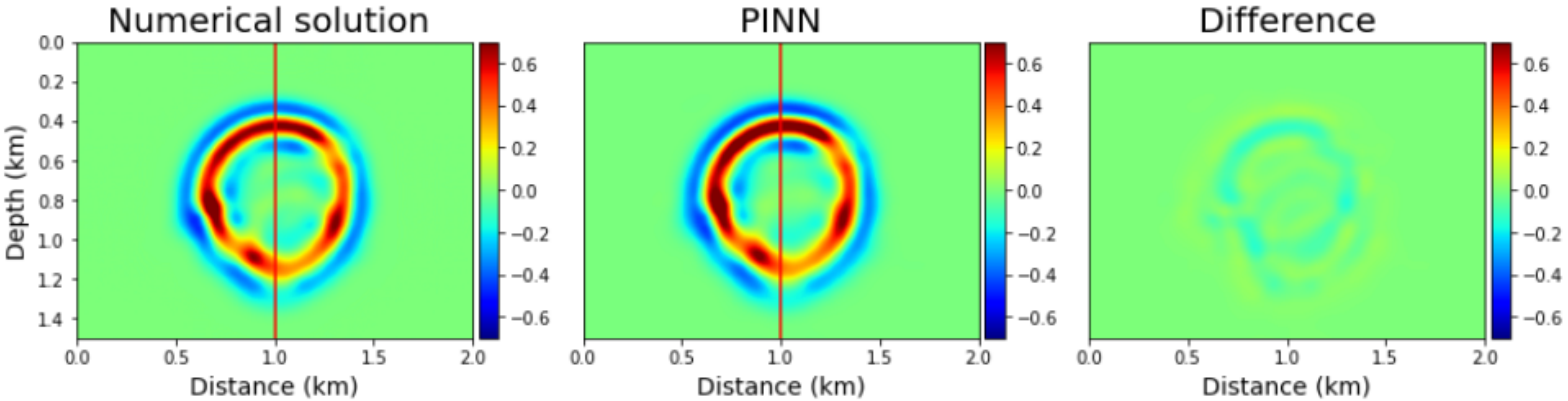
$t = 0.2\ s$

EXPERIMENT-3: MARMOUSI MODEL (MODIFIED)



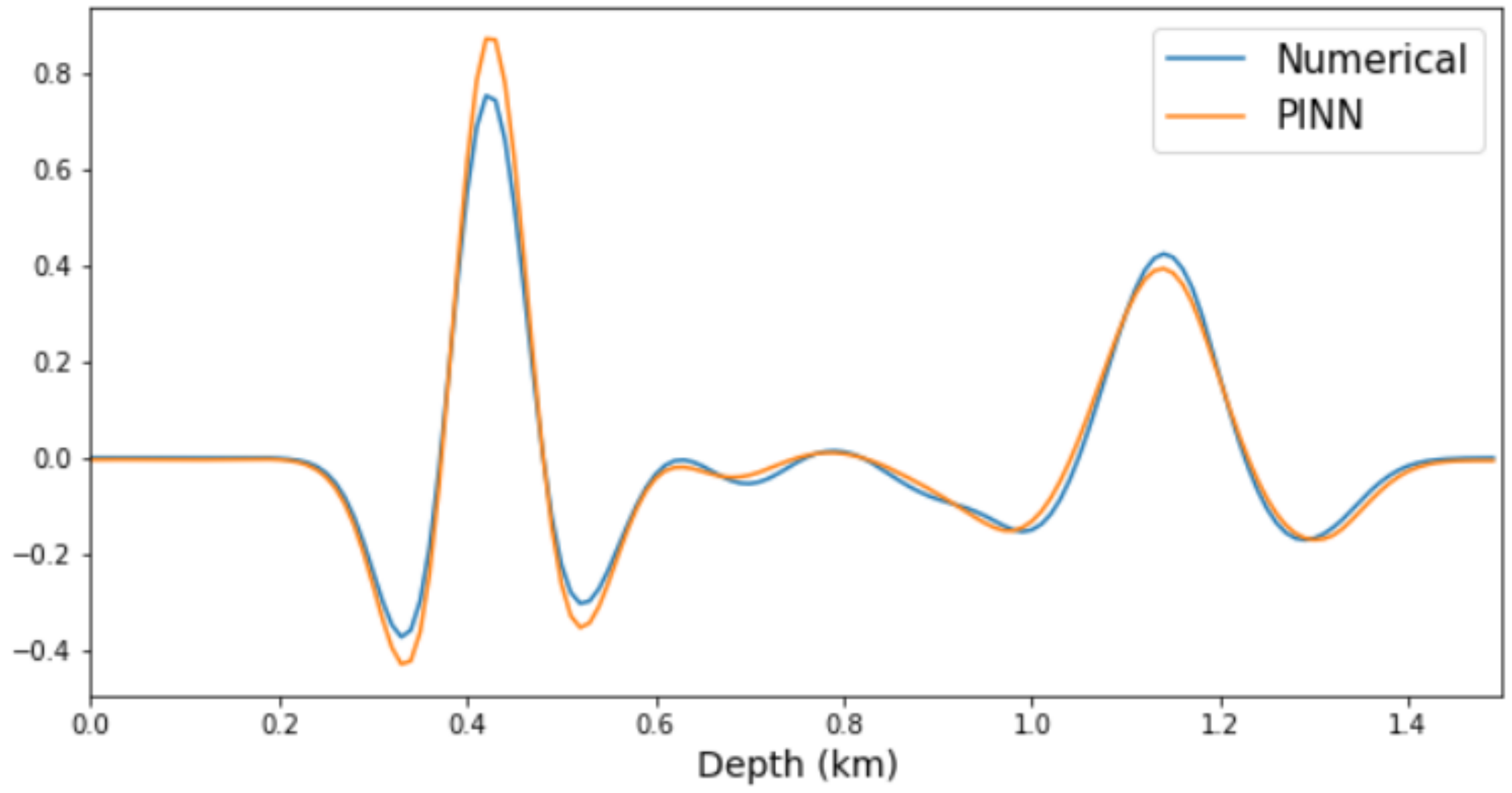
$t = 0.35\ s$

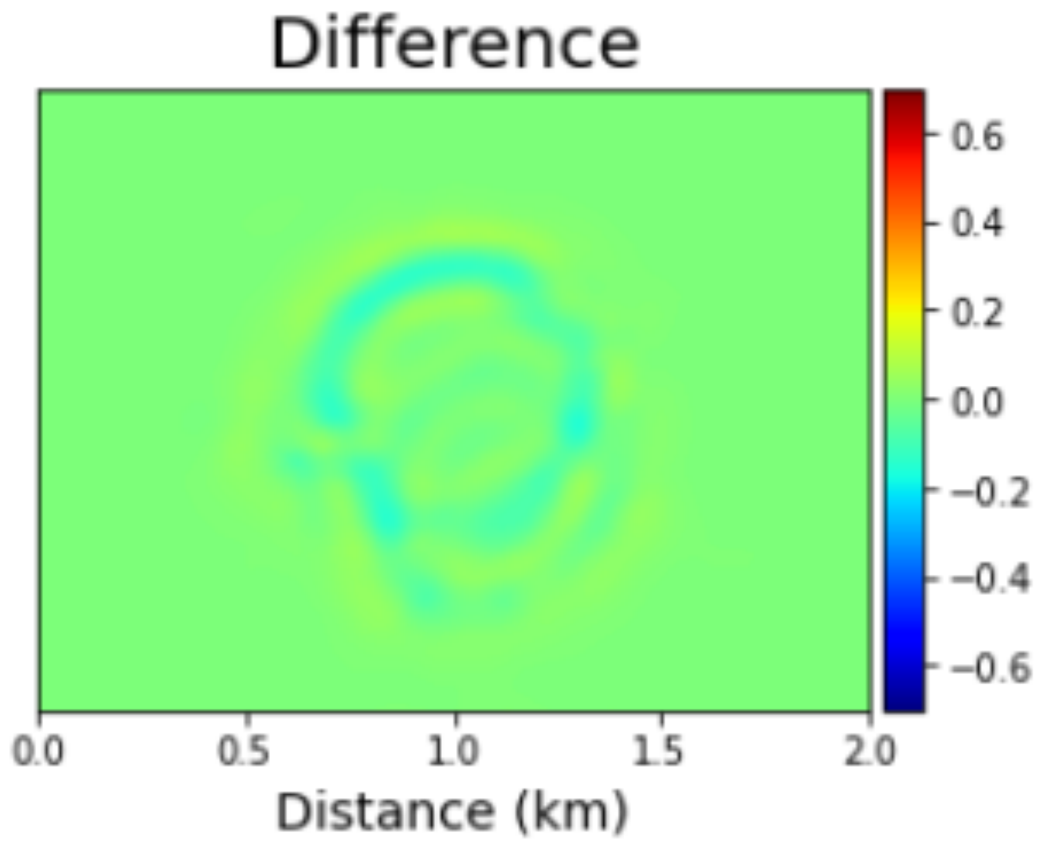
EXPERIMENT-3: MARMOUSI MODEL (MODIFIED)



$t = 0.35\ s$

EXPERIMENT-3: MARMOUSI MODEL (MODIFIED)



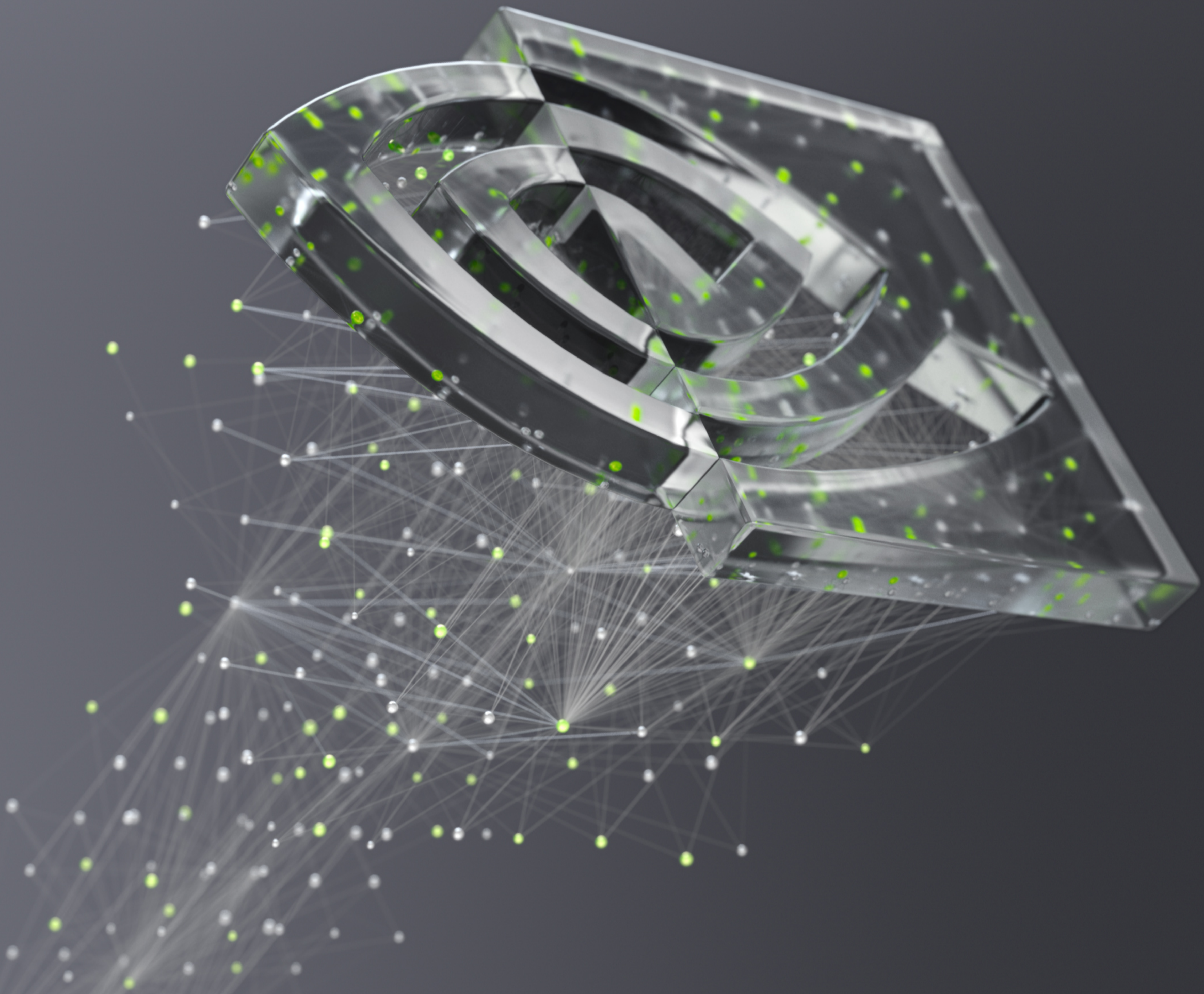


AGENDA

- Motivation
- Theory
- Experimental setup
- Numerical examples
- Conclusion and next steps

NEXT STEPS

- Numerical solutions with reasonable accuracy even for heterogeneous models.
- Hard enforcement of the initial conditions was the key factor to train the NN without using data.
- Extension to waveform inversion applications.



NVIDIA