Gradient-enhanced Physics-Informed Neural Networks (gPINNs)

MIT PRIMES Conference, 10/16/21

Presented by: Jeremy Yu
Mentor: Dr. Lu Lu

Partial Differential Equations

$$f\left(\mathbf{x};\frac{\partial u}{\partial x_1},\ldots,\frac{\partial u}{\partial x_d};\frac{\partial^2 u}{\partial x_1 \partial x_1},\ldots,\frac{\partial^2 u}{\partial x_1 \partial x_d};\ldots;\boldsymbol{\lambda}\right)=0, \quad \mathbf{x}=(x_1,\cdots,x_d)\in\Omega$$

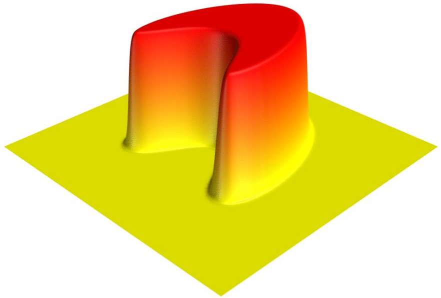
Unsolvable?

$$\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0$$

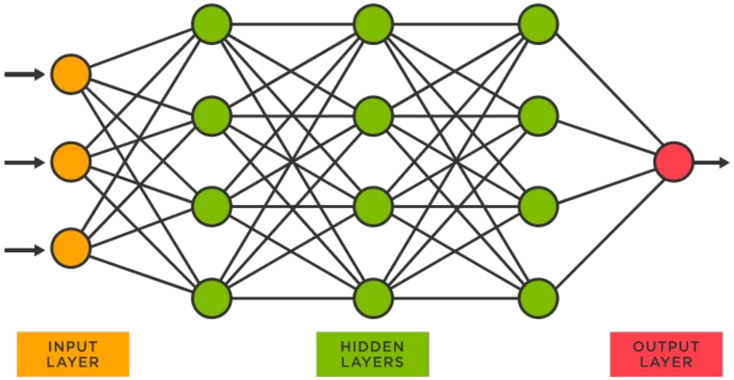
$$\rho(\frac{\partial v_x}{\partial t}+v_x\frac{\partial v_x}{\partial x}+v_y\frac{\partial v_x}{\partial y}+v_z\frac{\partial v_x}{\partial z})=-\frac{\partial P}{\partial x}+\mu(\frac{\partial^2 v_x}{\partial x^2}+\frac{\partial^2 v_x}{\partial y^2}+\frac{\partial^2 v_x}{\partial z^2})+\rho g_x$$

$$\rho(\frac{\partial v_y}{\partial t}+v_x\frac{\partial v_y}{\partial x}+v_y\frac{\partial v_y}{\partial y}+v_z\frac{\partial v_y}{\partial z})=-\frac{\partial P}{\partial y}+\mu(\frac{\partial^2 v_y}{\partial x^2}+\frac{\partial^2 v_y}{\partial y^2}+\frac{\partial^2 v_y}{\partial z^2})+\rho g_y$$

$$\rho(\frac{\partial v_z}{\partial t}+v_x\frac{\partial v_z}{\partial x}+v_y\frac{\partial v_z}{\partial y}+v_z\frac{\partial v_z}{\partial z})=-\frac{\partial P}{\partial z}+\mu(\frac{\partial^2 v_z}{\partial x^2}+\frac{\partial^2 v_z}{\partial y^2}+\frac{\partial^2 v_z}{\partial z^2})+\rho g_z$$



Deep-learning: FNNs



$$
\begin{aligned}
\text{input layer:} \quad & \mathcal{N}^0(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^{d_{\text{in}}}, \\
\text{hidden layers:} \quad & \mathcal{N}^\ell(\mathbf{x}) = \sigma(\boldsymbol{W}^\ell \mathcal{N}^{\ell-1}(\mathbf{x}) + \boldsymbol{b}^\ell) \in \mathbb{R}^{N_\ell}, \quad \text{for} \quad 1 \leq \ell \leq L-1, \\
\text{output layer:} \quad & \mathcal{N}^L(\mathbf{x}) = \boldsymbol{W}^L \mathcal{N}^{L-1}(\mathbf{x}) + \boldsymbol{b}^L \in \mathbb{R}^{d_{\text{out}}};
\end{aligned}
$$

Physics-Informed Neural Networks (PINNs)

$$f\left(\mathbf{x}; \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_d}; \frac{\partial^2 u}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_d}; \ldots; \boldsymbol{\lambda}\right) = 0, \quad \mathbf{x} = (x_1, \cdots, x_d) \in \Omega \quad \mathcal{B}(u, \mathbf{x}) = 0 \quad \text{on} \quad \partial\Omega$$

1. Create neural network

$$\hat{u}(\mathbf{x}; \boldsymbol{\theta})$$

2. Specify training set

3. Train the network to fit the constraints

Loss function: $\mathcal{L}(\boldsymbol{\theta}; \mathcal{T}) = w_f \mathcal{L}_f(\boldsymbol{\theta}; \mathcal{T}_f) + w_b \mathcal{L}_b(\boldsymbol{\theta}; \mathcal{T}_b)$

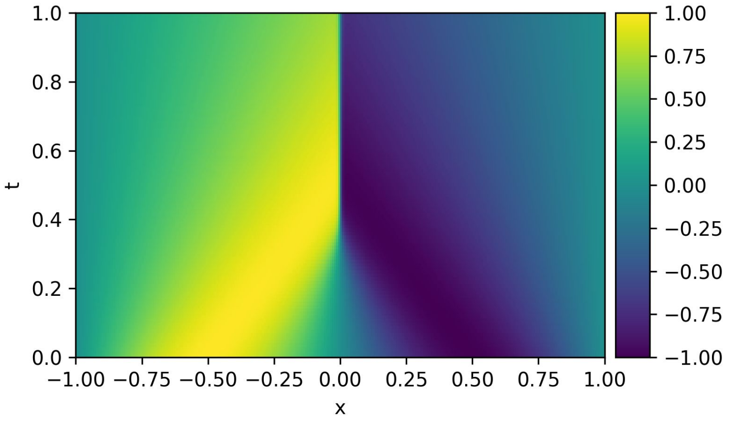
$$\mathcal{L}_f(\boldsymbol{\theta}; \mathcal{T}_f) = \frac{1}{|\mathcal{T}_f|} \sum_{\mathbf{x} \in \mathcal{T}_f} \left| f\left(\mathbf{x}; \frac{\partial \hat{u}}{\partial x_1}, \ldots, \frac{\partial \hat{u}}{\partial x_d}; \frac{\partial^2 \hat{u}}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 \hat{u}}{\partial x_1 \partial x_d}; \ldots; \boldsymbol{\lambda}\right) \right|^2$$

$$\mathcal{L}_b(\boldsymbol{\theta}; \mathcal{T}_b) = \frac{1}{|\mathcal{T}_b|} \sum_{\mathbf{x} \in \mathcal{T}_b} |\mathcal{B}(\hat{u}, \mathbf{x})|^2$$

Setback of PINNs

- Typically have a limited accuracy even with many training points

Consider Burgers' Equation:



gPINN

Idea: Provide additional information to the network through the gradient

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_d}\right) = \mathbf{0}, \quad \mathbf{x} \in \Omega.$$

New terms in Loss Function:

$$\mathcal{L} = w_f \mathcal{L}_f + w_b \mathcal{L}_b + w_i \mathcal{L}_i + \sum_{i=1}^{d} w_{g_i} \mathcal{L}_{g_i}\left(\boldsymbol{\theta}; \mathcal{T}_{g_i}\right)$$

$$\mathcal{L}_{g_i}\left(\boldsymbol{\theta}; \mathcal{T}_{g_i}\right) = \frac{1}{\left|\mathcal{T}_{g_i}\right|} \sum_{\mathbf{x} \in \mathcal{T}_{g_i}} \left|\frac{\partial f}{\partial x_i}\right|^2$$

Example additional loss terms

$$\Delta u = f$$

In 1D, $$\mathcal{L}_g = w_g \frac{1}{|\mathcal{T}_g|} \sum_{\mathbf{x} \in \mathcal{T}_g} \left| \frac{d^3 u}{dx^3} - \frac{df}{dx} \right|^2$$

In 2D, $$\mathcal{L}_{g_1} = w_{g_1} \frac{1}{|\mathcal{T}_{g_1}|} \sum_{\mathbf{x} \in \mathcal{T}_{g_1}} \left| \frac{\partial^3 u}{\partial x^3} + \frac{\partial^3 u}{\partial x \partial y^2} - \frac{\partial f}{\partial x} \right|^2,$$

$$\mathcal{L}_{g_2} = w_{g_2} \frac{1}{|\mathcal{T}_{g_2}|} \sum_{\mathbf{x} \in \mathcal{T}_{g_2}} \left| \frac{\partial^3 u}{\partial x^2 \partial y} + \frac{\partial^3 u}{\partial y^3} - \frac{\partial f}{\partial y} \right|^2.$$

Results

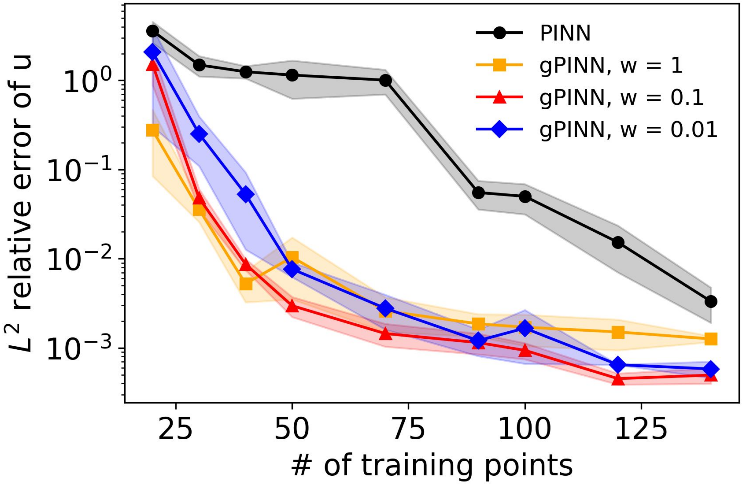
Diffusion-reaction equation

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + R(x,t), \quad x \in [-\pi, \pi],\ t \in [0,1],$$

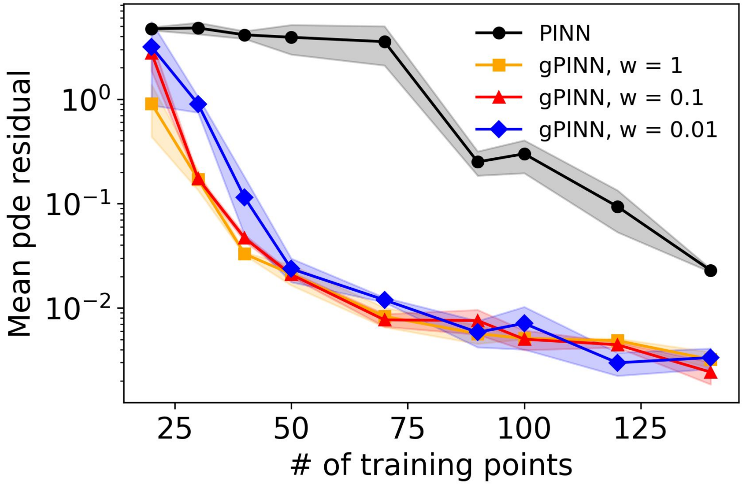
$$u(x,0) = \sum_{i=1}^{4} \frac{\sin(ix)}{i} + \frac{\sin(8x)}{8},$$
$$u(-\pi,t) = u(\pi,t) = 0,$$

$$R(x,t) = e^{-t}\left[\frac{3}{2}\sin(2x) + \frac{8}{3}\sin(3x) + \frac{15}{4}\sin(4x) + \frac{63}{8}\sin(8x)\right]$$

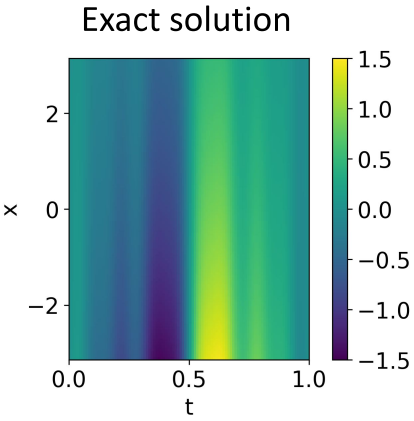
Analytic solution $$u(x,t) = e^{-t}\left[\sum_{i=1}^{4} \frac{\sin(ix)}{i} + \frac{\sin(8x)}{8}\right]$$

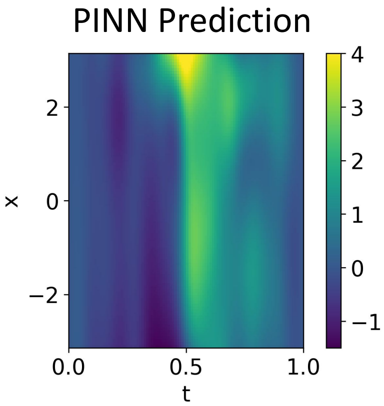


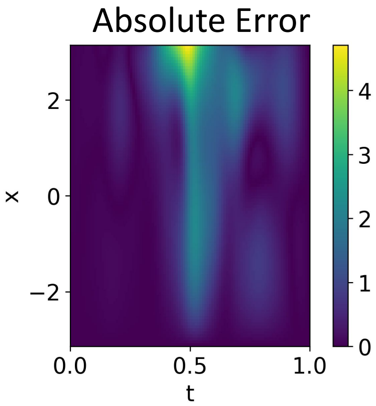
PINN
gPINN, w = 1
gPINN, w = 0.1
gPINN, w = 0.01
L^2 relative error of u
10^0
10^{-1}
10^{-2}
10^{-3}
25
50
75
100
125
# of training points

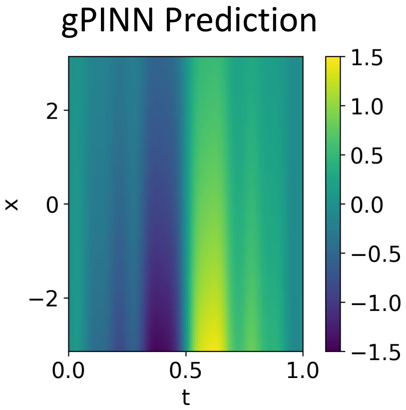


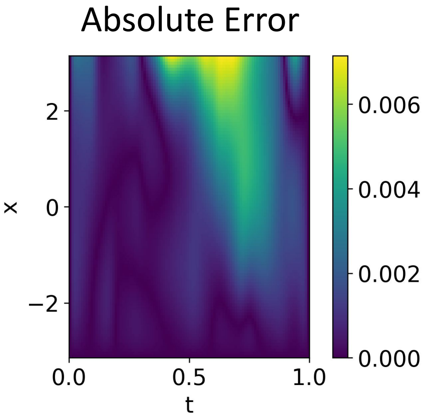
PINN
gPINN, w = 1
gPINN, w = 0.1
gPINN, w = 0.01
Mean pde residual
10^0
10^{-1}
10^{-2}
25
50
75
100
125
# of training points


Exact solution
x
t
0.0 0.5 1.0
2 0 −2
1.5 1.0 0.5 0.0 −0.5 −1.0 −1.5


PINN Prediction
x
t
0.0 0.5 1.0
2 0 −2
4 3 2 1 0 −1


Absolute Error
x
t
0.0 0.5 1.0
2 0 −2
4 3 2 1 0


gPINN Prediction
x
t
0.0 0.5 1.0
2 0 −2
1.5 1.0 0.5 0.0 −0.5 −1.0 −1.5


Absolute Error
x
t
0.0 0.5 1.0
2 0 −2
0.006 0.004 0.002 0.000

Inverse problems

$$f\left(\mathbf{x}; \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_d}; \frac{\partial^2 u}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_d}; \ldots; \boldsymbol{\lambda}\right) = 0, \quad \mathbf{x} \in \Omega \quad \mathcal{B}(u, \mathbf{x}) = 0 \quad \text{on} \quad \partial\Omega$$

$$\mathcal{L}_i\left(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_i\right) = \frac{1}{|\mathcal{T}_i|} \sum_{\mathbf{x} \in \mathcal{T}_i} |\hat{u}(\mathbf{x}) - u(\mathbf{x})|^2$$

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}) = w_f \mathcal{L}_f\left(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_f\right) + w_b \mathcal{L}_b\left(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_b\right) + w_i \mathcal{L}_i\left(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_i\right)$$

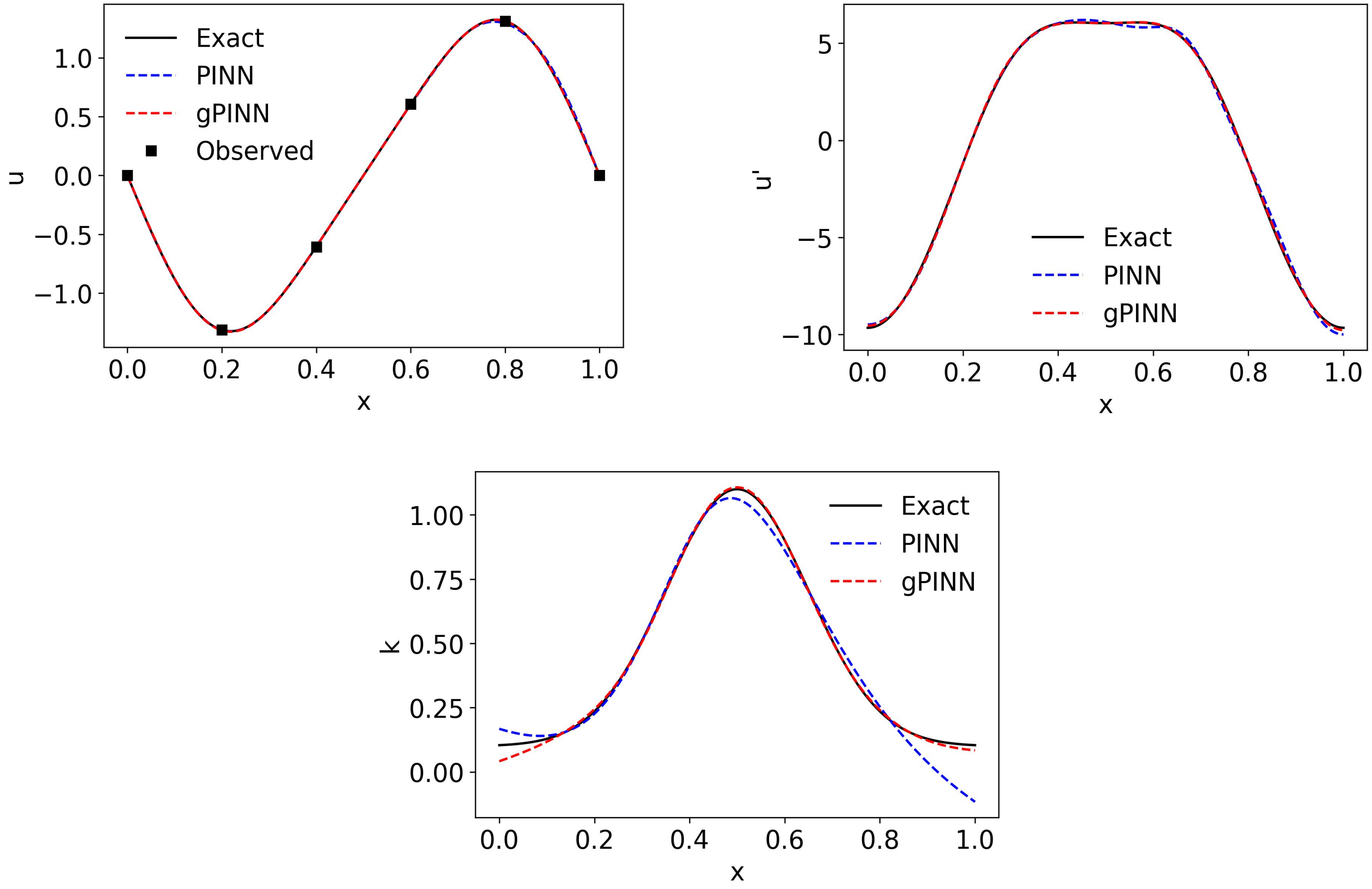
Inferring space-dependent reaction rate

$$\lambda \frac{\partial^2 u}{\partial x^2} - k(x)u = f, \quad x \in [0, 1],$$

$u(x) = 0$ is imposed at $x = 0$ and 1.

$$k(x) = 0.1 + \exp\left[-0.5\frac{(x-0.5)^2}{0.15^2}\right]$$

Inferring whole function instead of just a constant


Exact
PINN
gPINN
Observed
u
x
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
Exact
PINN
gPINN
u'
x
5
0
−5
−10
Exact
PINN
gPINN
k
x
1.00
0.75
0.50
0.25
0.00

gPINN + RAR

Algorithm 1: gPINN with RAR.

Step 1 Train the neural network using gPINN on the training set $\mathcal{T}$ for a certain number of iterations.

Step 2 Compute the PDE residual $\left| f\left(\mathbf{x}; \frac{\partial \hat{u}}{\partial x_1}, \ldots, \frac{\partial \hat{u}}{\partial x_d}; \frac{\partial^2 \hat{u}}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 \hat{u}}{\partial x_1 \partial x_d}; \ldots; \boldsymbol{\lambda}\right)\right|$ at random points in the domain.

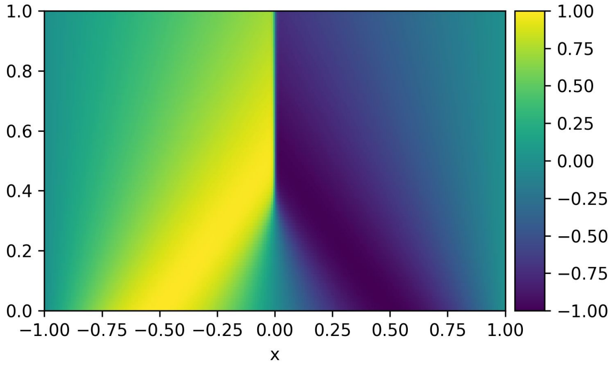
Step 3 Add m new points to the training set $\mathcal{T}$ where the residual is the largest.

Step 4 Repeat Steps 1, 2, and 3 for n times, or until the mean residual falls below a threshold $\mathcal{E}$.

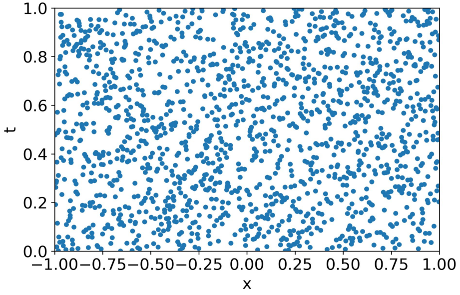
Burgers' Equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in [-1, 1], t \in [0, 1]$$

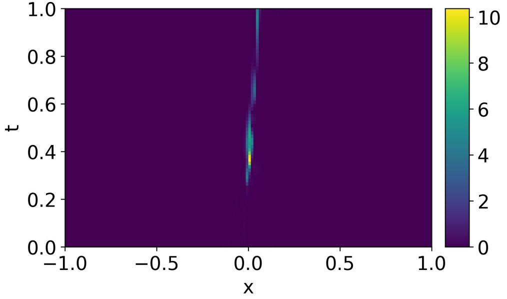
$$u(x, 0) = -\sin(\pi x), \quad u(-1, t) = u(1, t) = 0$$



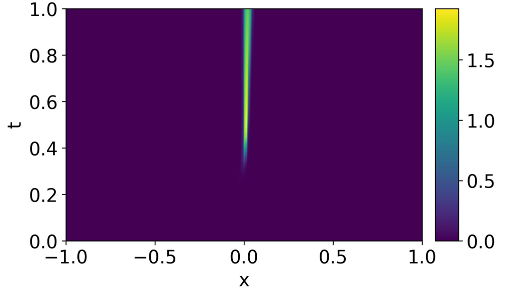
Initial training points



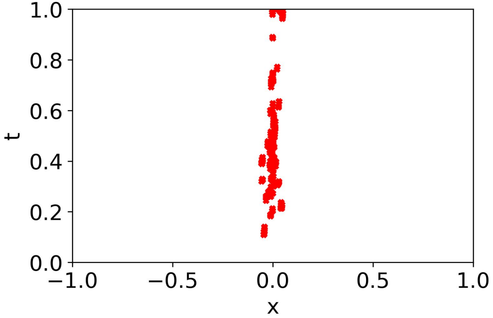
Error of PDE residual



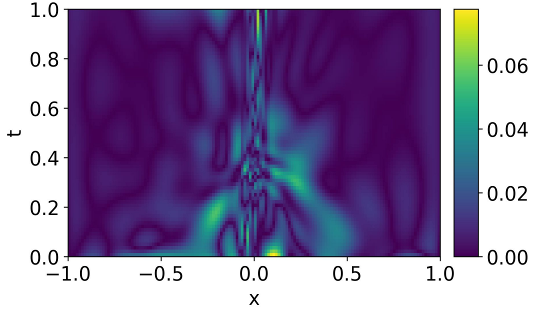
Absolute Error



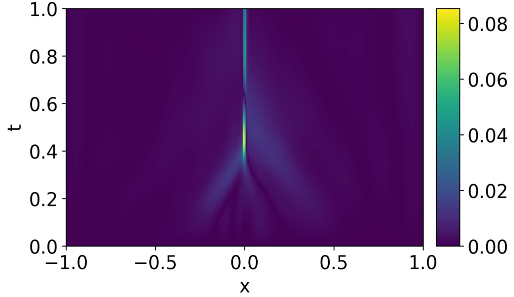
200 added points



Error of PDE residual



Absolute Error



Acknowledgements

- Mentor, Dr. Lu Lu
- MIT PRIMES-USA
- Parents

Thank you!

References

-Heat equation GIF: https://en.wikipedia.org/wiki/Partial_differential_equation#/media/File:Heat.gif

-Picture of FNN: https://www.tibco.com/reference-center/what-is-a-neural-network