A Universal PINNs Method for Solving Partial Differential Equations with a Point Source

Xiang Huang[1] Hongsheng Liu[2] Beiji Shi[2] Zidong Wang[2]
Kang Yang[2] Yang Li[2] Min Wang[2] Haotian Chu[2] Jing Zhou[2]
Fan Yu[2] Bei Hua[1] Bin Dong[3] Lei Chen[4]

1 2 3 4

Deep Learning for Solving PDEs

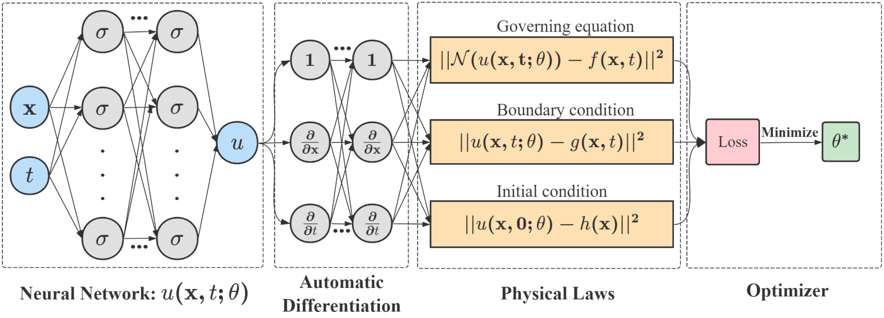
- **Data-Driven Methods** Based on data obtained from traditional numerical solvers or real-world observations, the solution mapping of parametric PDEs is approximated by neural network.
  - PDE-Net Z. Long et al., ICML 2018.
  - Deep Operator network (DeepONet) L. Lu, P. Jin, G. E. Karniadakis, arXiv:1910.03193.
  - Fourier Neural Operator (FNO) Z. Li et al., arXiv:2010.08895.

- **Physics-Informed Methods** PDEs or their variant forms are used as loss terms for training the neural network, so that the output of the neural network can satisfy PDEs.
  - Deep Ritz Method (DRM) W. E and B. Yu, CMS, 6(1), 1-12, 2018.
  - Deep Galerkin Method (DGM) Sirignano and Spiliopoulos, JCP, 375:1339-1364, 2018.
  - Physics-Informed Neural Networks (PINNs) M. Raissi et al., JCP, 378:686-707, 2019.

Physics-Informed Neural Networks (PINNs)

General form of a time-dependent PDE

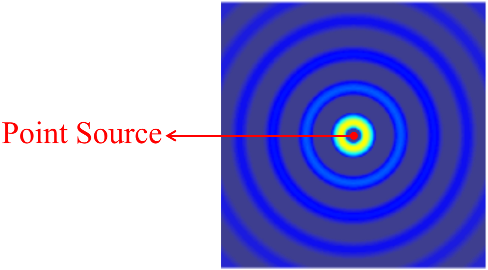
$$\begin{aligned} \mathcal{N}(u(\mathbf{x},t)) &= f(\mathbf{x},t), \quad & (\mathbf{x},t) \in \Omega \times [0,T] \\ u(\mathbf{x},t) &= g(\mathbf{x},t), \quad & (\mathbf{x},t) \in \partial\Omega \times [0,T] \\ u(\mathbf{x},0) &= h(x), \quad & \mathbf{x} \in \Omega \end{aligned}$$



PDEs with a Point Source

Acoustic wave equation with a point source

$$u_{tt} - c^2(u_{xx} + u_{yy}) = f(x, y, t)$$
$$f(x, y, t) = h(t)\delta(x - x_0)\delta(y - y_0)$$

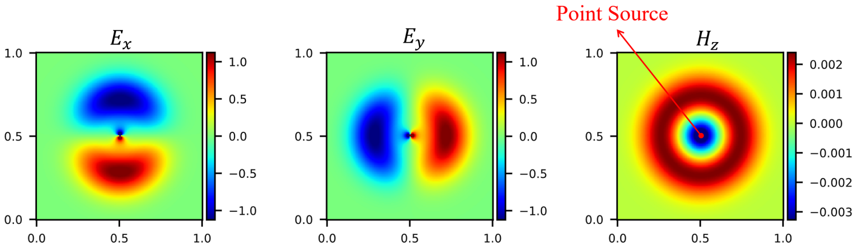


PDEs with a Point Source

Maxwell's equations with a point source

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon}\frac{\partial H_z}{\partial y}, \qquad \frac{\partial E_y}{\partial t} = -\frac{1}{\epsilon}\frac{\partial H_z}{\partial x}, \qquad \frac{\partial H_z}{\partial t} = -\frac{1}{\mu}\left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} + J\right).$$

$$J(x, y, t) = e^{-\left(\frac{t-d}{\tau}\right)^2}\delta(x - x_0)\delta(y - y_0).$$

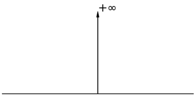


Dilemmas of PINNs in Solving PDEs with a Point Source

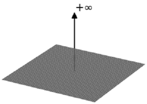
Definition of Dirac function

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

$$\int_{-\infty}^{+\infty} \delta(x)\, dx = 1.$$



1D Dirac function



2D Dirac function

Because of the singularity brought by Dirac function, the PINNs method is difficult to solve PDEs with a point source.

Approximating $\delta(\mathbf{x})$ with a Probability Density Function

$$\eta_\alpha(\mathbf{x}) = \alpha^{-1}\eta(\frac{\mathbf{x}}{\alpha}) \ \approx \ \delta(\mathbf{x}), \ as\ \alpha \to 0.$$

- Gaussian distribution

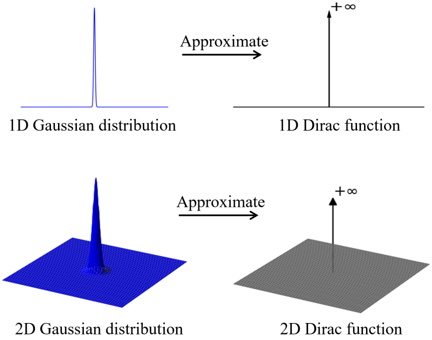
$$\eta_\alpha(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\alpha} e^{-\frac{x^2}{2\alpha^2}}$$

- Cauchy distribution

$$\eta_\alpha(\mathbf{x}) = \frac{1}{\pi}\frac{\alpha}{x^2+\alpha^2}$$

- Laplacian distribution

$$\eta_\alpha(\mathbf{x}) = \frac{1}{2\alpha} e^{-\frac{|\mathbf{x}|}{\alpha}}$$



Lower Bound Constrained Uncertainty Weighting

- **General Form of PDEs with a Point Source**

$$
\begin{aligned}
&\mathcal{N}(u(\mathbf{x}, t)) = f(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \Omega \times [0, T] \\
&u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \partial\Omega \times [0, T] \\
&u(\mathbf{x}, 0) = h(x), \qquad \mathbf{x} \in \Omega
\end{aligned}
$$

where $f(\mathbf{x}, t)$ contains $\delta(\mathbf{x} - \mathbf{x}_0)$ and $\mathbf{x}_0$ is the coordinate of the point source.

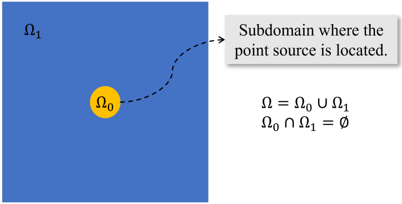
- **Physics-Informed Loss**

$$L_{total}(\theta) = L_r(\theta) + \lambda_{bc} L_{bc}(\theta) + \lambda_{ic} L_{ic}(\theta),$$

$$L_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} ||\mathcal{N}(u(\mathbf{x}_i, t_i; \theta)) - f(\mathbf{x}_i, t_i)||_2^2,$$

$$L_{bc}(\theta) = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} ||u(\mathbf{x}_i, t_i; \theta) - g(\mathbf{x}_i, t_i)||_2^2,$$

$$L_{ic}(\theta) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} ||u(\mathbf{x}_i, 0; \theta) - h(\mathbf{x}_i)||_2^2.$$



- **Decompose Ω into Ω_0 and Ω_1**

$$
\begin{aligned}
L_r(\theta) =& \lambda_{r,0} L_{r,0}(\theta) + \lambda_{r,1} L_{r,1}(\theta), \\
L_{total}(\theta) =& \lambda_{r,0} L_{r,0}(\theta) + \lambda_{r,1} L_{r,1}(\theta) + \\
& \lambda_{bc} L_{bc}(\theta) + \lambda_{ic} L_{ic}(\theta).
\end{aligned}
$$

$$L_{r,0} \gg L_{r,1},\ L_{bc},\ L_{ic}$$

How to choose different λ for different loss terms?

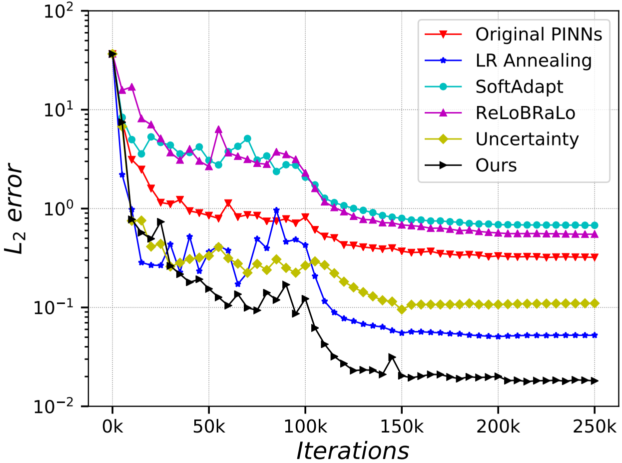
Lower Bound Constrained Uncertainty Weighting

w_i represents the uncertainty of the i-th loss term, which is a trainable parameter.

Weighting Methods	Formula
Original PINNs	$L_{total}(\theta) = \sum_{i=1}^{m} \lambda_i L_i(\theta)$
Uncertainty (Kendall et al., CVPR 2018)	$L_{total}(\theta; \mathbf{w}) = \sum_{i=1}^{m} \frac{1}{2w_i^2} L_i(\theta) + \log w_i$
Ours	$L_{total}(\theta; \mathbf{w}) = \sum_{i=1}^{m} \frac{1}{2(\epsilon^2 + w_i^2)} L_i(\theta) + \log(\epsilon^2 + w_i^2)$

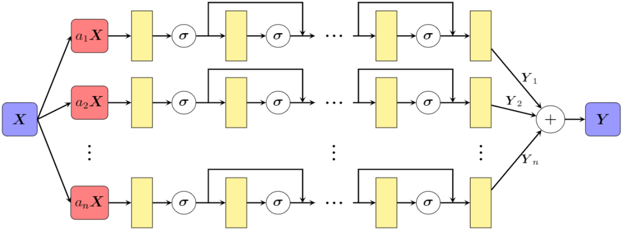
the lower bound of uncertainty

Lower Bound Constrained Uncertainty Weighting

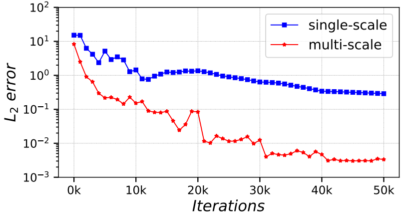


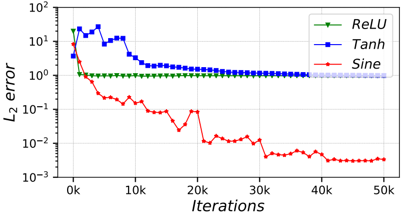
Barry and Mercer's source problem: Convergence speed of the mean L_2 $error$s with different loss weighting methods.

Multi-Scale DNNs with Periodic Activation Function



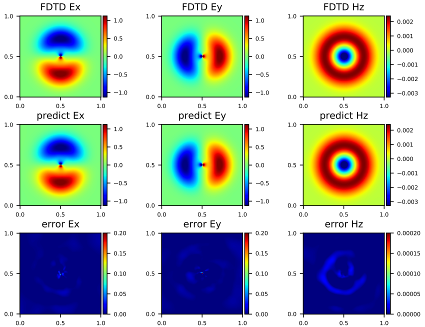
MS-SIREN: Consists of n subnets with different scaling parameter $\{a_1, \cdots, a_n\}$ and the activation function $\sigma(x) = sin(x)$.





Maxwell's Equations with a Point Source

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon}\frac{\partial H_z}{\partial y},$$
$$\frac{\partial E_y}{\partial t} = -\frac{1}{\epsilon}\frac{\partial H_z}{\partial x},$$
$$\frac{\partial H_z}{\partial t} = -\frac{1}{\mu}\left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} + J\right).$$
$$J(x,y,t) = e^{-\left(\frac{t-d}{\tau}\right)^2}\delta(x-x_0)\delta(y-y_0).$$

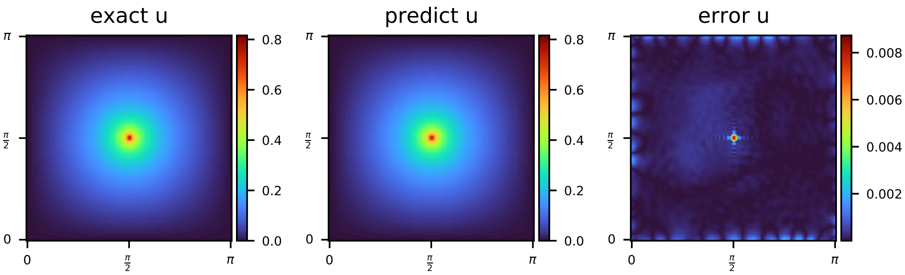


Network Architectures			L_2 *error*			
# subnets	# layers	# neurons	E_x	E_y	H_z	mean
1	7	256	0.192	0.183	0.433	0.269
1	9	256	0.147	0.146	0.070	0.121
2	7	128	0.079	0.074	0.058	0.072
2	9	128	0.054	0.053	0.019	0.027
4	7	64	0.021	0.022	0.001	0.018
4	9	64	0.025	0.022	0.017	0.021

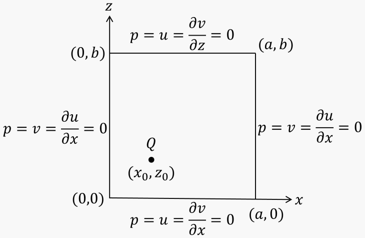
Poisson's Equation with a Point Source

$$
\begin{aligned}
-\Delta u &= \delta(\mathbf{x} - \mathbf{x}_0), & \mathbf{x} &\in \Omega, \\
u &= 0, & \mathbf{x} &\in \partial\Omega
\end{aligned}
$$

Network Architectures			L_2 *error*
# subnets	# layers	# neurons	
1	5	256	0.289
1	7	256	1.081
2	5	128	0.003
2	7	128	0.006
4	5	64	0.003
4	7	64	0.002



Barry and Mercer's Source Problem

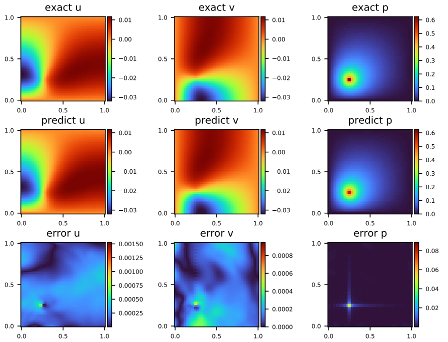


$$\frac{\partial^2 u}{\partial t \partial x} + \frac{\partial^2 v}{\partial t \partial z} - \frac{\partial^2 p}{\partial x^2} - \frac{\partial^2 p}{\partial z^2} - \beta Q = 0,$$

$$(\eta + 1)\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} + \eta \frac{\partial^2 v}{\partial x \partial z} - (\eta + 1)\frac{\partial p}{\partial x} = 0,$$

$$\frac{\partial^2 v}{\partial x^2} + (\eta + 1)\frac{\partial^2 v}{\partial z^2} + \eta \frac{\partial^2 u}{\partial x \partial z} - (\eta + 1)\frac{\partial p}{\partial z} = 0.$$

$$Q(x, z, t) = \delta(x - x_0)\delta(z - z_0) sin(\omega t)$$



Thank you! Q&A.

- Source Code: https://gitee.com/mindspore/mindscience/tree/master/MindElec/
- Our code is implemented by MindSpore.
- For more questions, please send email to sahx@mail.ustec.edu.cn.