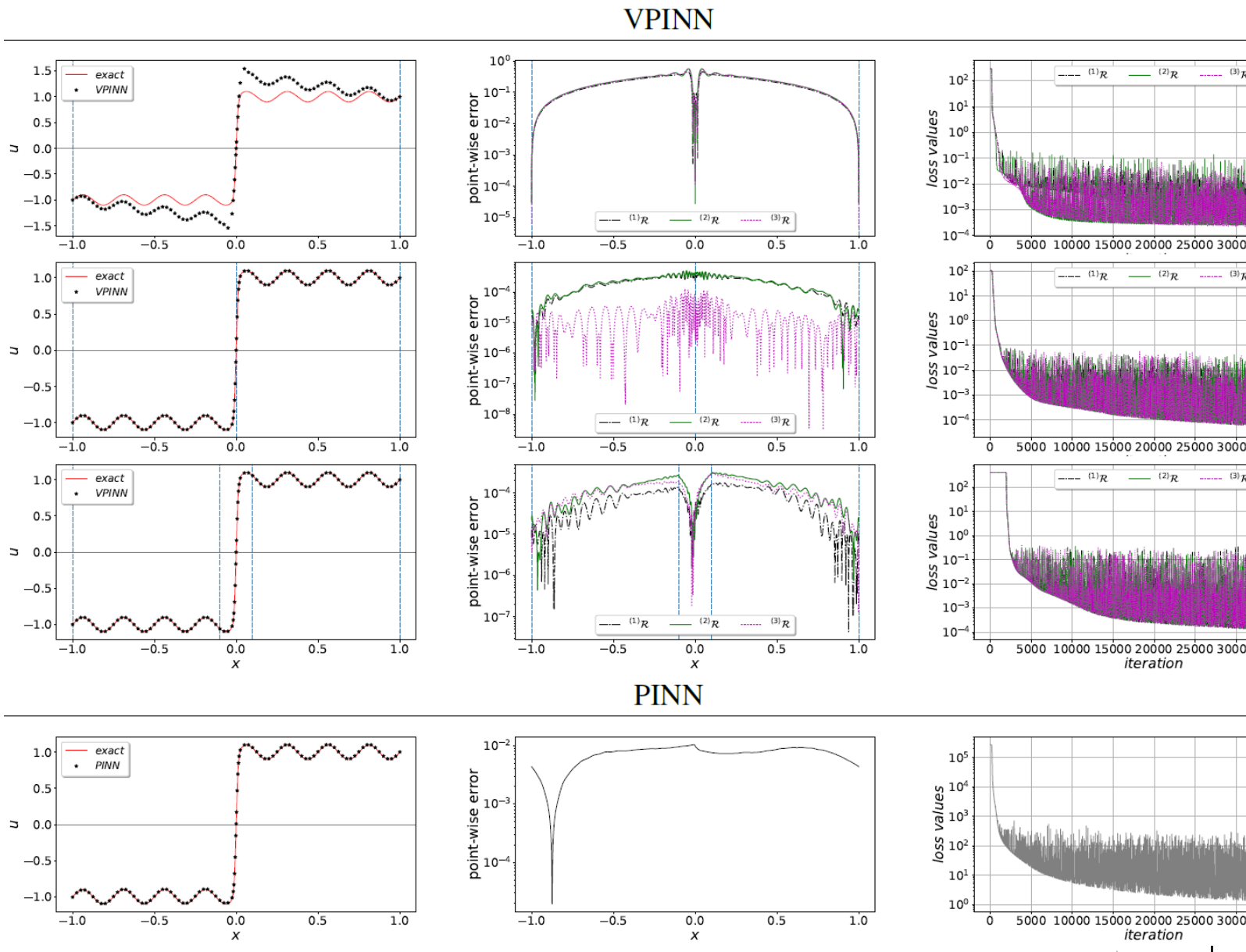
VPINNs: Poisson's Equation (1D)

le: **Steep Solution**

$$(x) = 0.1\sin(8\pi x) + \tanh(80x)$$

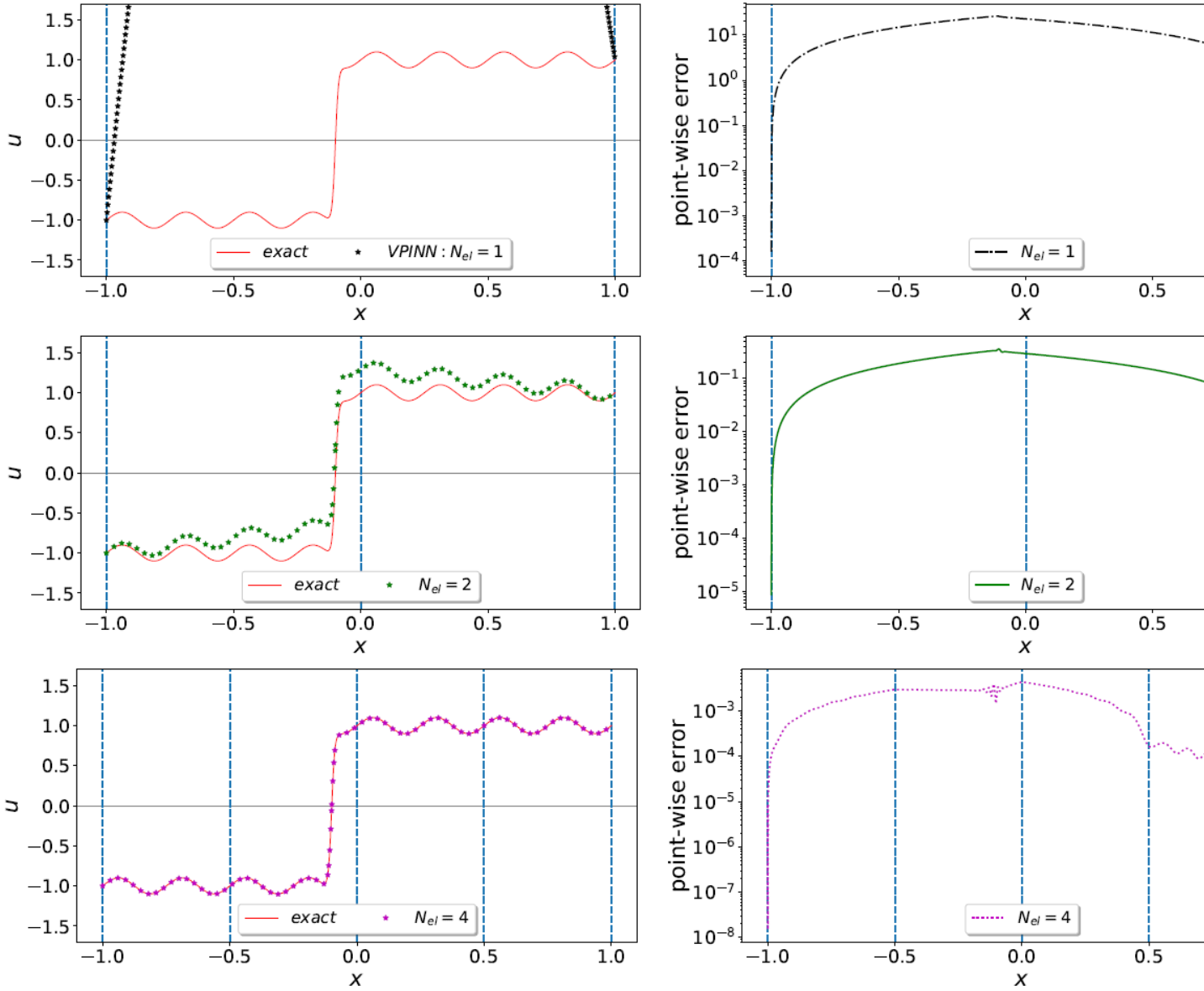


-VPINNs: Poisson's Equation (1-D)

ample: **Singularity Capturing**

$$u^{\text{exact}}(x) = 0.1\sin(8\pi x) + \tanh(80(x+0.1))$$

ymmetric steep part



Kharazmi E, Zhang Z, Karniadakis GE. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Computer Methods in Applied Mechanics and Engineering. 2021 Feb 1;374:113547.

hp-VPINNs: Poisson's Equation (2-D)

$$\nabla^2 u(x,y) = f(x,y), \qquad (x,y) \in \Omega = [-1,1] \times [-1,1]$$

Discrete finite dimensional space, given by

$$\tilde{V} = \text{span}\left\{v_{k_1 k_2}(x,y) = \phi_{k_1}(x)\phi_{k_2}(y), k_m = 1,2,\cdots,K_m, m = 1,2\right\}$$

$$\phi_{k_1}^{(e_x)}\left(x_{e_x-1}\right) = \phi_{k_1}^{(e_x)}\left(x_{e_x}\right) = 0, \quad k_1 = 1,2,\cdots,K_1,$$

$$\phi_{k_2}^{(e_y)}\left(y_{e_y-1}\right) = \phi_{k_2}^{(e_y)}\left(y_{e_y}\right) = 0, \quad k_2 = 1,2,\cdots,K_2.$$

The variational residual becomes

$$\mathcal{R}_{k_1 k_2}^{(e_e e_y)} = \int_{x_{e_x-1}}^{x_{e_x}} \int_{y_{e_y-1}}^{y_{e_y}} \left(\nabla^2 u_{NN}(x,y) - f(x,y)\right) \phi_{k_1}^{(e_x)}(x)\phi_{k_2}^{(e_y)}(y)dxdy$$

The variational loss

$$\mathcal{L}^{\mathfrak{v}(i)} = \sum_{e_r=1}^{N_{el_r}} \sum_{e_y=1}^{N_{el_y}} \frac{1}{K_1 K_2} \sum_{k=1}^{K_1 K_2} \left| {}^{(i)}\mathcal{R}_k^{(e_x e_y)} \right|^2 + \tau_b \frac{1}{N_b} \sum_{i=1}^{N_b} \left| r_b\left(x_b^i, y_b^i\right) \right|^2, \quad i = 1,2,3$$

Kharazmi E, Zhang Z, Karniadakis GE. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Computer Methods in Applied Mechanics and Engineering. 2021 Feb 1;374:113547.

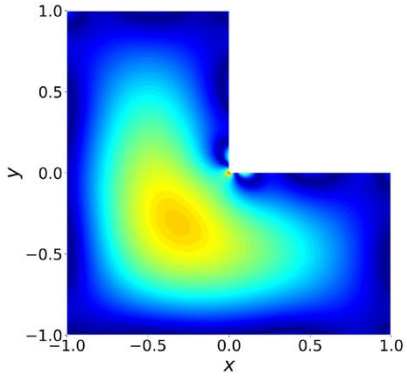
p-VPINNs: Poisson's Equation (2-D)

D PDE: $\begin{cases} \nabla^2 u(x,y) = f(x,y) \\ \text{Homogeneous Dirichlet B.C.} \end{cases}$

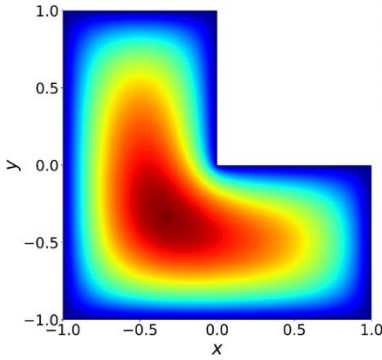
ne homogeneous case $f(x,y) = 0$ gives the Laplace equation

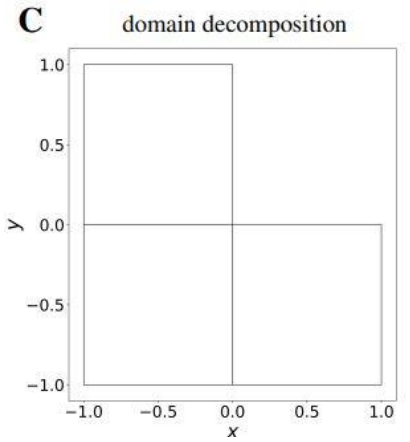
hp-VPINN:(1) $\mathcal{R}$ formulation

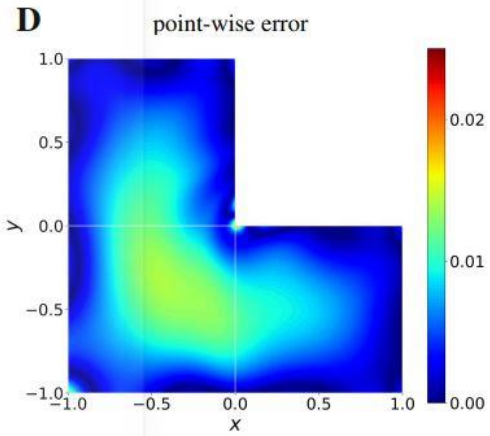
PINNs point-wise error

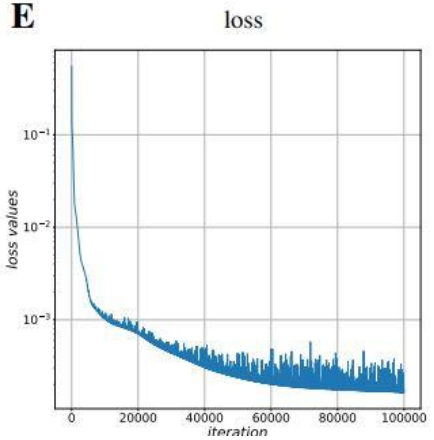


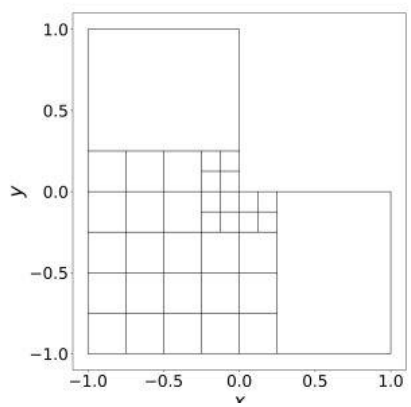
reference solution

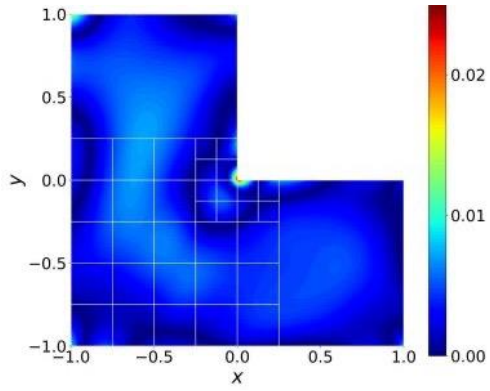


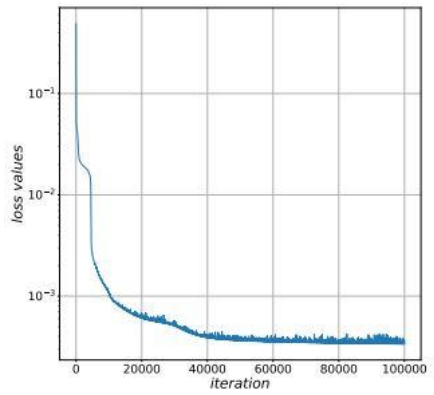










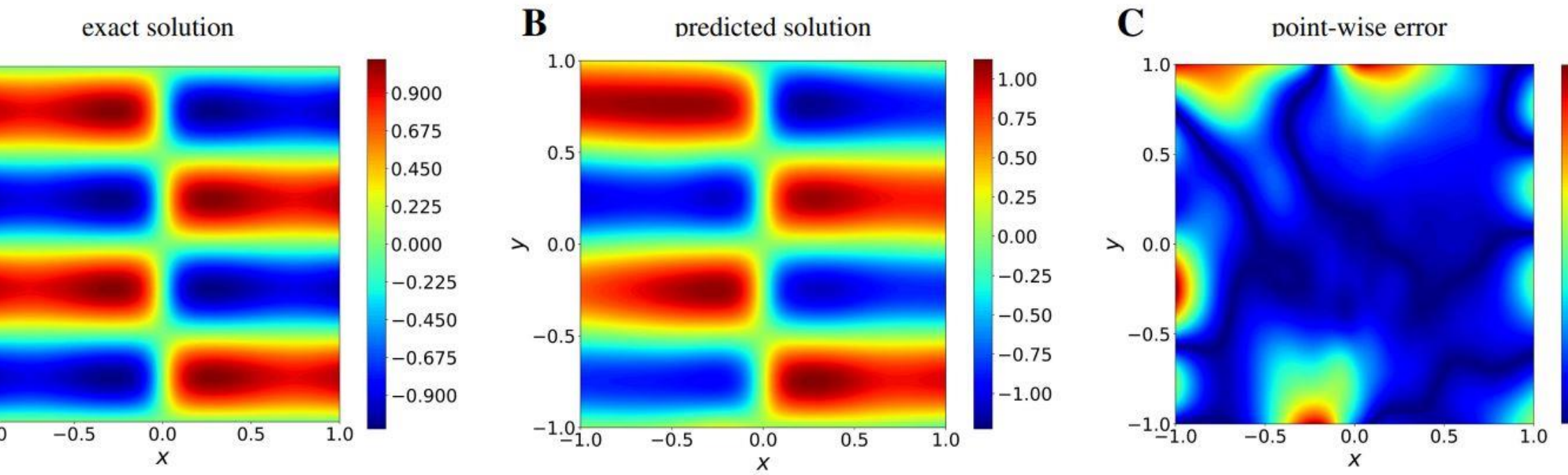


Kharazmi E, Zhang Z, Karniadakis GE. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Computer Methods in Applied Mechanics and Engineering. 2021 Feb 1;374:113547.

p-VPINNs: Poisson's Equation

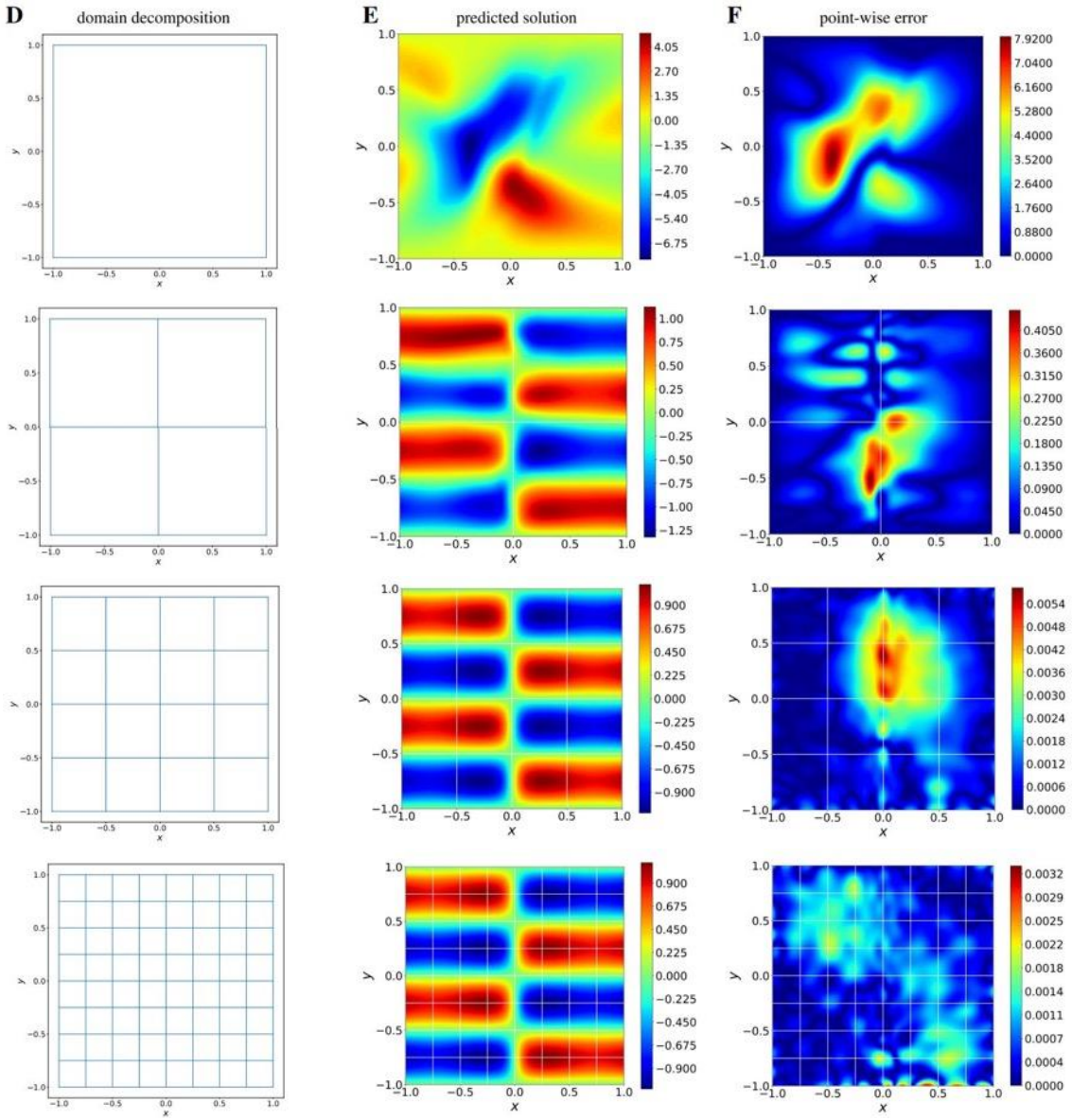
D PDE: $\begin{cases} \nabla^2 u(x,y) = f(x,y) \\ \text{Homogeneous Dirichlet B.C.} \end{cases}$, $u^{\text{exact}}(x,y) = \sin(2\pi y) \times (0.1\sin(2\pi x) + \tanh(10$

PINN



Kharazmi E, Zhang Z, Karniadakis GE. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Computer Methods in Applied Mechanics and Engineering. 2021 Feb 1;374:113547.

-VPINNs: Poisson's Equation



Kharazmi E, Zhang Z, Karniadakis GE. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Computer Methods in Applied Mechanics and Engineering. 2021 Feb 1;374:113547.

hp-VPINNs (Summary)

Nonlinear approximation via neural networks

h-refinement via domain decomposition

p-refinement via projection onto high-order polynomial space

Integration by parts to reduce the regularity and simplify the network

Analytical expression of variational loss for shallow networks

Adaptive regularity in each sub-domain via different networks and test functions

Numerical integration in deep networks

Mapping of test functions in domains with complex geometry

Integration in high dimensional problems VPINNs theory is still missing!

Kharazmi E, Zhang Z, Karniadakis GE. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Computer Methods in Applied Mechanics and Engineering. 2021 Feb 1;374:113547.

Convergence Theory for PINNs

INNs generate a sequence of minimizers, which correspond to a sequence of neural networks.

oes the sequence of minimizers converge to the solution of the PDE?

he sequence of minimizers strongly converges to the PDE solution in C^0 + if each minimizer sat
fies the initial/boundary conditions, the convergence mode becomes H^1.

ey idea: **Schauder Approach + Weak Maximum Principle**

Shin Y, Darbon J, Karniadakis GE. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. arXiv preprint arXiv:2004.01806. 2020 Apr 3.

Setup for PINNs

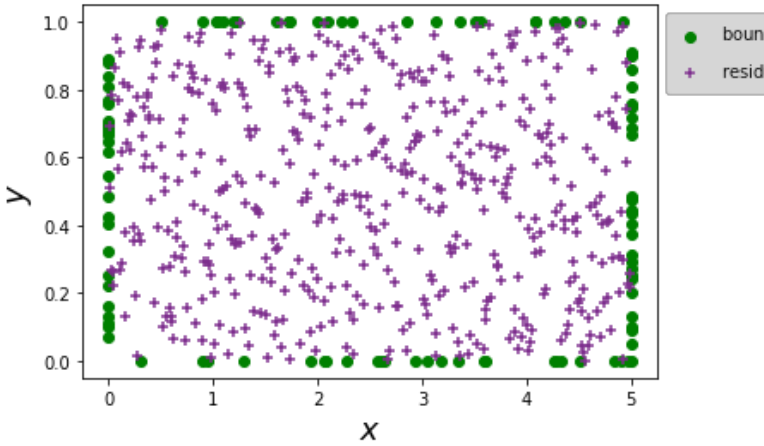
Consider the partial differential equations (PDEs):

$$\mathcal{L}[u](\mathbf{x}) = f(\mathbf{x}) \text{ in } U \subset \mathbb{R}^d$$
$$\mathcal{B}[u](\mathbf{x}) = g(\mathbf{x}) \text{ in } \partial U$$

Goal: Find a neural net $h_\theta(\mathbf{x})$ that approximates the PDE solution.

What are known to us?

- Pointwise PDE data: i.i.d. samples
- Residual data: $\left(\mathbf{x}_r^i, f\left(\mathbf{x}_r^i\right)\right)$ where $\mathbf{x}_r^i \in U$
- Boundary data: $\left(\mathbf{x}_b^i, g\left(\mathbf{x}_b^i\right)\right)$ where $\mathbf{x}_b^i \in \partial U$
- Physics Equations: $\mathcal{L}[u] - f = 0$ and $\mathcal{B}[u] - g = 0$



Physics-Informed Neural Networks

Idea: Encode "physics equations" and "data" into the neural net $h_\theta(\mathrm{x})$

Physics

Data

DEF: Let $\boldsymbol{m} = (m_r, m_b)$. The prototype PINN loss is

$$\mathrm{Loss}_m^{\mathrm{PINN}}(\theta) = \frac{1}{m_r}\sum_{i=1}^{m_r}\left(\mathcal{L}\left[h_\theta\right]\left(\mathbf{x}_r^i\right) - f\left(\mathbf{x}_r^i\right)\right)^2 + \frac{1}{m_b}\sum_{i=1}^{m_b}\left(\mathcal{B}\left[h_\theta\right]\left(\mathbf{x}_b^i\right) - g\left(\mathbf{x}_b^i\right)\right)^2$$

DEF: The Hölder regularized PINN loss is

$$\mathrm{Loss}_m^{\mathrm{Höld}}(\theta) = \mathrm{Loss}_m^{\mathrm{PINN}}(\theta) + \lambda_r^R\left(\left[\mathcal{L}\left[h_\theta\right]\right]_{\alpha;U}\right)^2 + \lambda_b^R\left(\left[\mathcal{B}\left[h_\theta\right]\right]_{\alpha;\partial U}\right)^2$$

Hölder constant → $[u]_{\alpha;U} = \sup_{x\neq y\in U}\frac{\|u(x)-u(y)\|}{\|x-y\|^\alpha}$

Goal: $\underset{\theta}{\text{minimize}}\,\mathrm{Loss}_m^{\mathrm{Höld}}(\theta) \Longrightarrow \theta_m^*$

$h_m := h_{\theta_m^*}$ is a physics informed neural network $\Longrightarrow \{h_m\}$

Q: Does $\{h_m\}$ converge to the PDE solution? In what norm?

Hölder Regularization: Motivation

deally, we wish to minimize the expected PINN loss :

$$\text{Loss}_{\infty}^{\text{PINN}}(h) = \mathbb{E}\left[\text{Loss}_{m}^{\text{PINN}}(h)\right]$$

Theorem: (Shin, Darbon, Karniadakis, 20) Suppose iid samples. Then, with high probability ove
d samples,

$$\text{Loss}_{\infty}^{\text{PINN}}(\theta) \leq C_m \, \text{Loss}_{m}^{\text{Höld}}(\theta) + C'\left(m_r^{-\frac{\alpha}{d}} + m_b^{-\frac{\alpha}{d-1}}\right)$$

Expected PINN loss

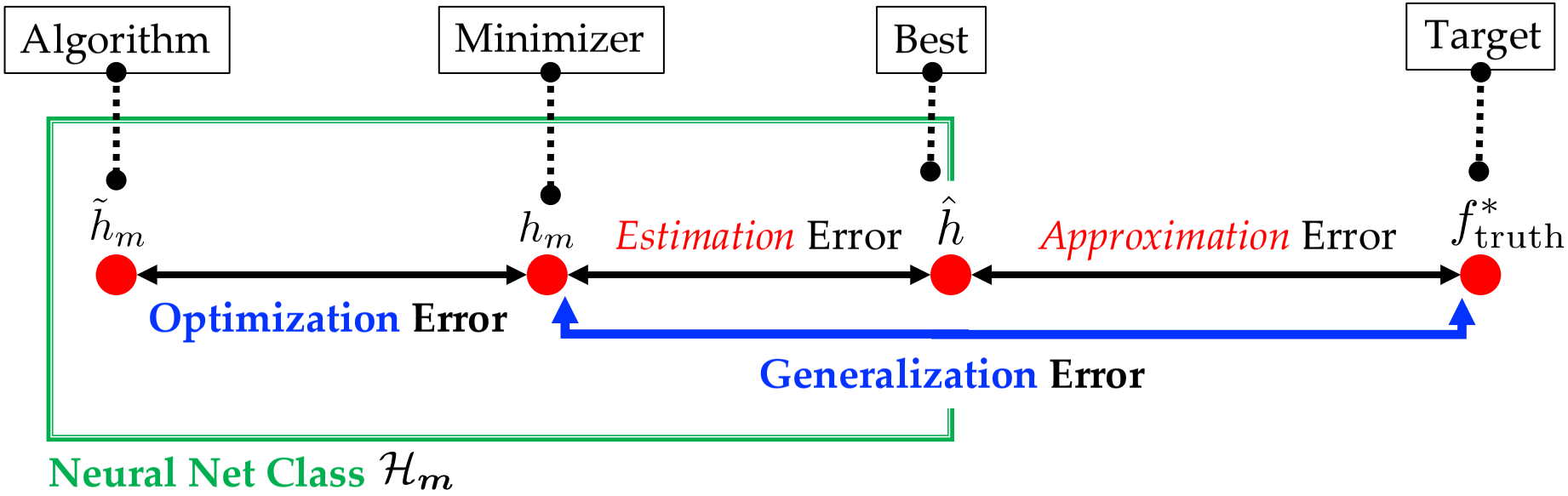
Hölder Regularized empirical loss

- $C_m = \Theta\left(\max\left\{\sqrt{m_r}, \sqrt{m_b}\right\}\right)$
- C' is a universal constant
- $\lambda^R = \left(\lambda_r^R, \lambda_b^R\right), \quad \lambda_r^R = \Theta\left(C_m^{-1} m_r^{-\alpha/d}\right), \quad \lambda_b^R = \Theta\left(C_m^{-1} m_b^{-\alpha/(d-1)}\right).$

Remark: Derived by applying probabilistic space filling arguments. Optimization is not involve
et.

Error Decomposition

$$\underset{\theta}{\text{minimize}}\, \text{Loss}_m^{\text{Höld}}(\theta) \Longrightarrow \theta_m^* \Longrightarrow h_m := h_{\theta_m^*}$$



- Neural net classes $\mathcal{H}_m$ have to be properly chosen for the convergence!
- $h_\theta \in \mathcal{H}_m$ gives $\min_\theta \text{Loss}_m^{\text{Höld}}(\theta)$
- Equivalently, $\min_{h \in \mathcal{H}_m} \text{Loss}_m^{\text{Höld}}(h)$

Assumptions on Neural Network Classes

To solve $\min_{h \in \mathcal{H}_m} \text{Loss}_m^{\text{Höld}}(h)$

choose a proper neural net class $\mathcal{H}_m$

Recall:

$$\text{Loss}_m^{\text{PINN}}(h) = \frac{1}{m_r} \sum_{i=1}^{m_r} \left(\mathcal{L}[h]\left(\mathbf{x}_r^i\right) - f\left(\mathbf{x}_r^i\right) \right)^2 + \frac{1}{m_b} \sum_{i=1}^{m_b} \left(\mathcal{B}[h]\left(\mathbf{x}_b^i\right) - g\left(\mathbf{x}_b^i\right) \right)^2$$

Assumption 1: Interpolation of data. There exists $u_m^* \in \mathcal{H}_m$ such that

Approximation error; $\quad \text{Loss}_m^{\text{PINN}}\left(u_m^*\right) = 0$

Assumption 2: Uniform boundedness of the Hölder constants.

Estimation error; $\quad \sup_m \left[\mathcal{L}\left[u_m^*\right]\right]_{\alpha;U} < \infty, \quad \sup_m \left[\mathcal{B}\left[u_m^*\right]\right]_{\alpha;\partial U} < \infty.$

- These assumptions are essential for the proof.
- Assumption 1 can be relaxed.
- Both are satisfied if $u^* \in \mathcal{H}_m$.

Convergence of the Expected PINN Loss

Recall: The expected PINN loss $\text{Loss}_{\infty}^{\text{PINN}}(h) = \mathbb{E}\left[\text{Loss}_{m}^{\text{PINN}}(h)\right]$

Theorem: (Shin, Darbon, Karniadakis 20) Let Assumptions 1 & 2 hold. Let $\min_{\mathcal{H}_m} \text{Loss}_{m}^{\text{Höld}}(h) \Longrightarrow h_m$.

- With high probability over i.i.d. samples,

$$\text{Loss}_{\infty}^{\text{PINN}}(h_{\boldsymbol{m}}) = \mathcal{O}\left(m^{-\frac{\alpha}{d}}\right).$$

Expected PINN loss

Worst case rate of convergence

- With probability 1 over i.i.d. samples,

$$\lim_{m\to\infty} \mathcal{L}[h_{\boldsymbol{m}}] = f \text{ in } C^0(U), \qquad \lim_{\boldsymbol{m}\to\infty} \mathcal{B}[h_{\boldsymbol{m}}] = g \text{ in } C^0(\partial U)$$

Remark: The uniform convergences do not necessarily imply the convergence of $\{h_m\}$ to the PDE solu
Ex) $f_n(x) = n\cos(x/n)$ on $U = [-1, 1]$

Schauder and Maximum Principle

- PDE classes having
  - existence, regularity and uniqueness of classical solution
- Schauder approach for linear 2^{nd} order elliptic and parabolic PDEs.
- PINNs DO NOT impose hard constraints on boundary conditions
  - Weak Maximum Principle
- Available information

$$\lim_{m\to\infty} \|\mathcal{L}[h_m] - f\|_{C^0(U)} = 0, \quad \lim_{m\to\infty} \|\mathcal{B}[h_m] - g\|_{C^0(\partial U)} = 0$$

- The sensitivity analysis of the PDE is required:
  - Suppose u satisfies $\mathcal{L}[u] = \epsilon, \mathcal{B}[u] = \epsilon$
  - Q: Is u necessarily small if $\|\epsilon\|$ is small?

Elliptic PDEs with Dirichlet BCs

- $\begin{cases} \mathcal{L}[u] = f & \text{in } U, \\ u = g & \text{in } \partial U, \end{cases}$ $\mathcal{L}[u] = \sum_{i,j=1}^{d} a^{ij}(\mathbf{x}) D_{ij} u + \sum_{i=1}^{d} \tilde{b}^i(\mathbf{x}) D_i u + \tilde{c}(\mathbf{x}) u$

- Schauder Assumptions:
  Let $\lambda(\boldsymbol{x})$ be the minimum eigenvalues of $\left[a^{ij}(\boldsymbol{x})\right]$ and $\alpha \in (0,1)$.
  - 1. (Strictly elliptic) For some constant $\lambda_0, \lambda(\boldsymbol{x}) \geq \lambda_0 > 0$ in U.
  - 2. The coefficients of the operator $\mathcal{L}$ are in $C^{\alpha}(U)$.
  - 3. There are constants $\Lambda, v \geq 0$ such that for all $x \in U$

$$\sum_{i,j} \left|a^{ij}(x)\right|^2 \leq \Lambda, \quad \lambda_0^{-2} \left(\|b(x)\|^2 + \|c(x)\|^2\right) + \lambda_0^{-1} |d(x)| \leq v^2,$$

    where $\boldsymbol{b}(\boldsymbol{x}) = \left[b^1(\boldsymbol{x}), \cdots, b^d(\boldsymbol{x})\right]^T$ and $\boldsymbol{c}(\boldsymbol{x}) = \left[c^1(\boldsymbol{x}), \cdots, c^d(\boldsymbol{x})\right]^T$.
  - 4. a^{ij} and b^i are in $C^{1,\alpha}(U)$. Also g is continuous on ∂U.
  - 5. U satisfies the exterior sphere condition at every boundary point.

- Weak Maximum Principle. $\tilde{c}(\mathbf{x}) \leq 0$ is assumed.

Elliptic PDEs with Dirichlet BCs

Key idea: Schauder Approach + Weak Maximum Principle

- **Theorem:** (Shin, Darbon, Karniadakis 20)
  Let Assumptions hold. Let $\min_{\mathcal{H}_m} \text{Loss}_m^{\text{Höld}}(h) \Longrightarrow h_m$.
  - With probability 1 over i.i.d. samples,

$$\lim_{m\to\infty} \|h_m - u^*\|_{C^0} = 0$$

  Furthermore, if the neural nets satisfy the boundary conditions,
  - With probability 1 over i.i.d. samples,

$$\lim_{m\to\infty} \|h_m - u^*\|_{H^1} = 0$$

  Proof idea: Linearity + Weak Maximum Principle + Apriori bound

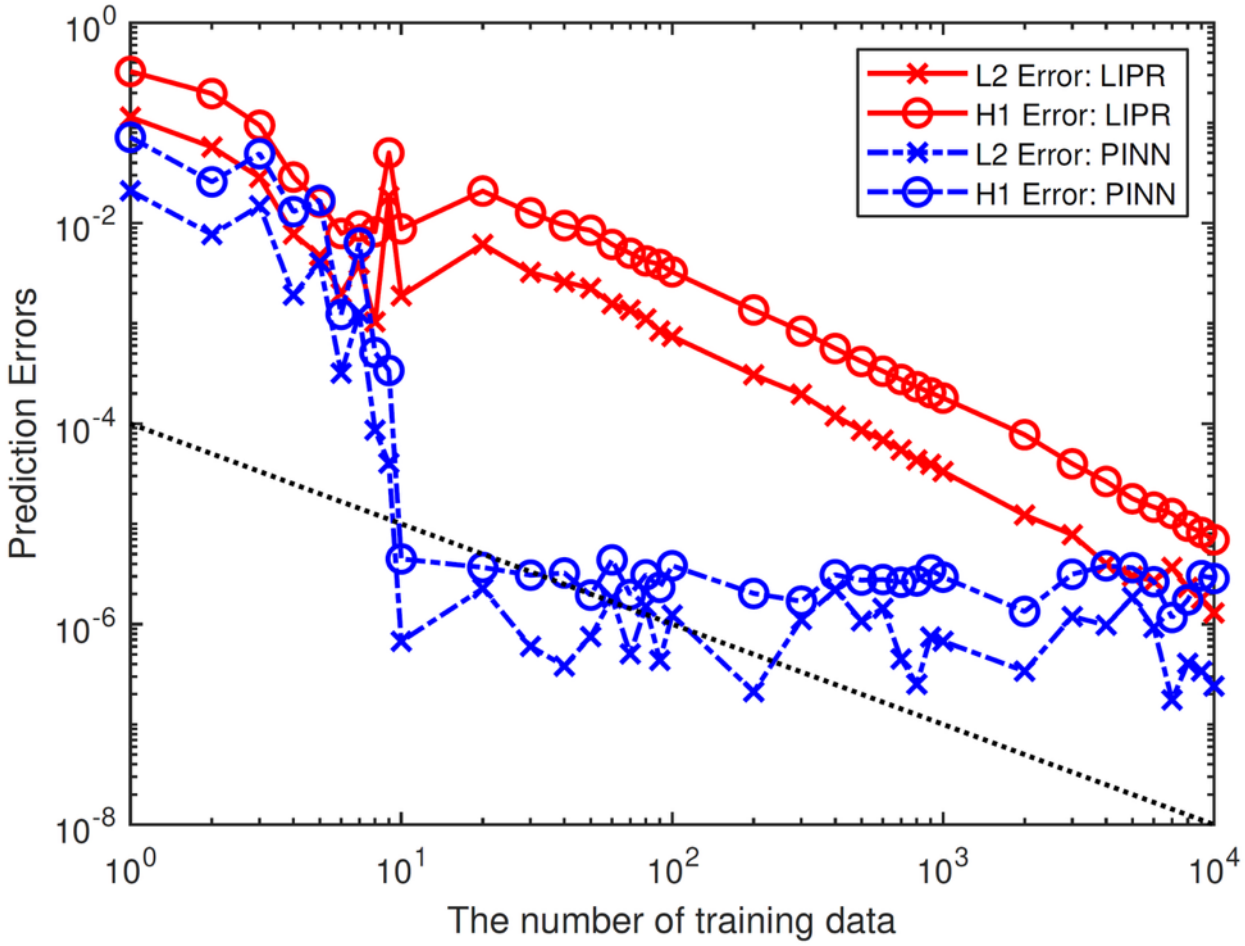
$$\|u^* - h_{\boldsymbol{m}}\|_{C^0(U)} \leq C\left(\|f - \mathcal{L}\left[h_{\boldsymbol{m}}\right]\|_{C^0(U)} + \|g - h_{\boldsymbol{m}}\|_{C^0(\partial U)}\right)$$

- Similar results are available for parabolic PDEs (omitted)

Computational Examples

1D Poisson Equation: $-u_{xx} = f$

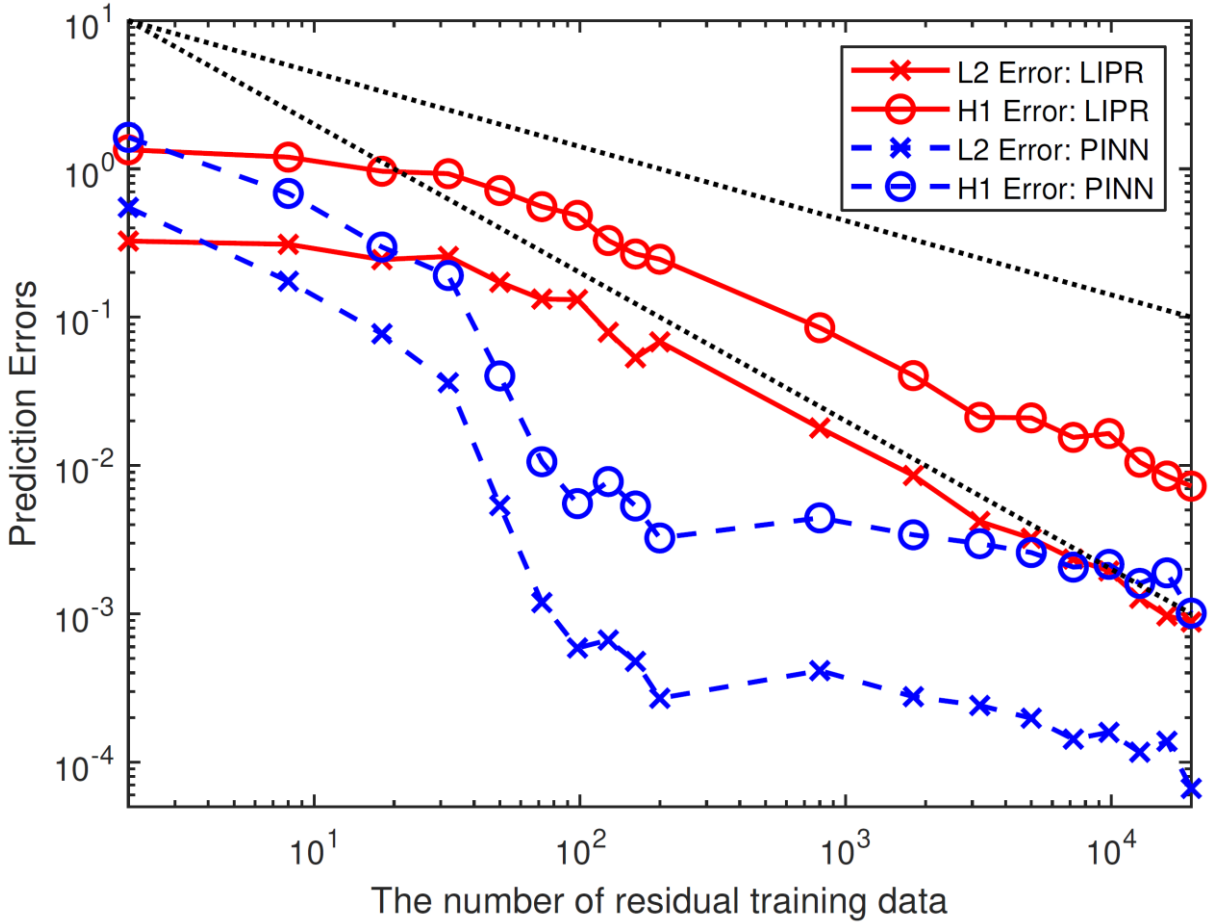
Exact: $u^*(x) = \tanh(x)$



Computational Examples

1D Heat Equation: $-u_t + \nu u_{xx} = f$

Exact: $u^*(x,t) = \sin(\pi x)e^{-t}$



Error Estimates of Residual Minimization Using Neural Networks for Linear PDEs

Shin Y, Zhang Z, Karniadakis GE. Error estimates of residual minimization using neural networks for lin PDEs. arXiv preprint arXiv:2010.08019. 2020 Oct 15.

bstract framework for analyzing the convergence of both PINNs and (hp-)variational PINNs fo near PDEs.

priori and a posteriori error estimates → Convergence

he loss formulations: continuous vs discrete. For example,

Discrete formulation

Continuous formulation

$$\mathcal{J}_m(u_{\mathrm{NN}}) = \frac{1}{m}\sum_{i=1}^{m}\left(\nabla u_{\mathrm{NN}}(x_i) - f(x_i)\right)^2, \qquad \mathcal{J}_\infty(u_{\mathrm{NN}}) = \int \left(\nabla u_{\mathrm{NN}}(x) - f(x)\right)^2 dx$$

stability

existence

capable neural networks

compatible neural networks (stability in discrete norm or Rademacher complexity)

convergence for continuous formulation

convergence for discrete formulation

Linear Problem

Consider the following linear problem

$$Au = f, \quad x \in \Omega \subset \mathbb{R}^d$$
$$Bu = g, \quad x \in \Gamma \subset \mathbb{R}^d.$$

$A : X \to Y$ is linear; $B : X \to Z$ is linear, X, Y and Z are Banach spaces.

The physics-informed neural networks (PINN)

$$\inf_{v_{\mathbb{N},\theta} \in \mathfrak{N}_\theta \cap X} \mathcal{J}_\tau \left(v_{\mathbb{N},\theta} \right),$$

where $v_{\mathbb{N},\theta} \in \mathfrak{N}_\theta$ is a feedforward neural network and

- (continuous formulation)
$$\mathcal{J}_\tau(v) = \|Av - f\|_Y^2 + \tau \|Bv - g\|_Z^2, \tau > 0$$
where $Y = L^2(\Omega)$ and $Z = L^2(\partial\Omega)$.
- (discrete formulation) $\inf_{v_{\mathbb{N},\theta} \in \mathfrak{N}_\theta \cap x} \mathcal{J}_\tau^N \left(v_{\mathbb{N},\theta} \right)$,
$$\mathcal{J}_\tau^N(v) = \|Av - f\|_{Y_N}^2 + \tau \|Bv - g\|_{Z_N}^2, \tau > 0.$$

Stability of Solution

Assumption
The operators A and B satisfy the following condition:

$$C_1\|u\|_V \leq \|Au\|_Y + \|Bu\|_Z \leq C_2\|u\|_X, \quad X \subset V$$

for all $u \in X$ and the constants C_1 and C_2 do not depend on u but on the domain and the coefficients of the operator A and B.

Example 1 (Elliptic)

- Let $A = -\Delta, \quad B = Id. \quad Y = L^2(\Omega), Z = L^2(\partial\Omega)$ $X = H^2(\Omega)$. Ω is a C^2 domain. $V = H^{1/2}(\Omega), X = H^2(\Omega)$

$$C_1\|u\|_{H^{1/2}} \leq \| - \Delta u\| + \|u\|_{L^2(\partial\Omega)} \leq C_2\|u\|_{H^2}.$$

Example 2 (Advection)

- Let $\Omega = [0, L], A = \partial_x, \quad Bu = u(0). \quad Y = L^2(\Omega). \quad X = H^1(\Omega).$

$$C_1\|u\|_{H^1} \leq \|\partial_x u\| + |u(0)| \leq C_2\|u\|_{H^1}.$$

- Let $\Omega = [0, L], A = \partial_t + c\partial_x, \quad Bu$: periodic boundary condition and square wave as initial condition. $Y = L^1(\Omega). \quad X = \mathrm{BV}(\Omega).$

$$C_1\|u\|_{\mathrm{BV}} \leq \|Au\|_{L^1} + |Bu|_{L^1(\partial\Omega)} \leq C_2\|u\|_{\mathrm{BV}}.$$

Error Estimates

Theorem (A priori and a posteriori errors)

Assume

** (norm relation)
$C_1\|u\|_V \leq \|Au\|_Y + \|Bu\|_Z \leq C_2\|u\|_X, X \subset V.$

** NN approximation
Let $\tau \geq 1$, and $u^{\tau}_{\mathbb{N},\theta}$ is a minimizer of $\mathcal{J}_\tau$ and $u^* \in V$ is a solution to the considered problem. Let $Y = L^p(\Omega), Z = L^p(\Gamma)$.

- a posteriori estimate

$$\left\|u^* - u^{\tau}_{\mathbb{N},\theta}\right\|_V \leq C_1^{-1} 2^{\frac{p-1}{p}} \mathcal{J}_\tau^{1/p}\left(u^{\tau}_{\mathbb{N},\theta}\right),$$

- a priori estimate

$$\left\|u^{\tau}_{\mathbb{N},\theta} - u^*\right\|_V \leq 2^{\frac{p-1}{p}} \left(1 + \tau^{1/p}\right) C_1^{-1} \left(C_2 \inf_{w \in \mathfrak{N}_{\theta,n} \cap x} \|w - Au^*_\varepsilon\|_X + \mathcal{J}_1\left(u^*_\varepsilon\right)\right)$$

Stability in Discrete Norm

$$\mathcal{J}_\tau^N(v) = \|Av - f\|_{Y_N}^2 + \tau\|Bv - g\|_{Z_N}^2, \tau > 0$$

Y_N as a subspace of $Y\left(L^2(\Omega)\right)$; $\int_\Omega I_N v^2 = \sum_{i=1}^N \omega_i v^2(x_i) := \|v\|_{Y_N}^2$

- Assume that stability in discrete norm (Key assumption II)

$$\|v\|_{Y_N} \geq \frac{1}{2}\|v\|_Y.$$

  for all $v \in Y_c$ where Y_c is a compact subspace of Y and Y_N.

- Similar requirement for Z_N.

  Assuming $Au \in Y_c$ leads to

$$\frac{1}{2}C_1\|u\|_V \leq \|Au\|_{Y_N} + \|Bu\|_Z \leq C_3\|Au\|_Y + \|Bu\|_Z \leq (C_2 + C_3)\|u\|_X.$$

Stability in discrete norm can be achieved via inverse inequality.

High-Dimensional Problems

Uniform law of large numbers, instead of numerical integration (curvature)

$$\sup_{f\in\mathfrak{F}}\left|\frac{1}{M}\sum_{i=1}^{M}f\left(X_i\right)-\mathbb{E}[f(X)]\right|$$

Definition (Rademacher complexity)

Given a collection $\{X_i\}_{i=1}^{M}$ of i.i.d. random samples, the Rademacher complexity of the function class $\mathfrak{F}$ defined by

$$R_M(\mathfrak{F})=\mathbb{E}_{\{X_i,\epsilon_i\}_{i=1}^{M}}\left[\sup_{f\in\mathfrak{F}}\left|\frac{1}{M}\sum_{i=1}^{M}\epsilon_i f\left(X_i\right)\right|\right]$$

where ϵ_i 's are i.i.d. Rademacher random variables i.e., $\mathbb{P}\left(\epsilon_i=\pm 1\right)=0.5$.

Then it holds that with probability $Q_b=1-\exp\left(-\frac{M\delta^2}{2b^2}\right)$

$$\sup_{f\in\mathfrak{F}}\left|\frac{1}{M}\sum_{i=1}^{M}f\left(X_i\right)-\mathbb{E}[f(X)]\right|\leq 2R_M(\mathfrak{F})+\delta$$

Here b is the uniform bound for the function class $\mathfrak{F}$.

Convergence of VPINNs

Hilbert Spaces: $(V, \|\cdot\|_V) \quad (X, \|\cdot\|_X) \quad (Y, \|\cdot\|_Y) \quad (Z, \|\cdot\|_Z)$

Problem definition:

$$\mathcal{L}^q u(\mathrm{x}) = f(\mathrm{x}), \quad \mathrm{x} \in \Omega \subset \mathbb{R}^d \qquad \mathcal{L}^q : X \to Y$$

$$u(\mathrm{x}) = h(\mathrm{x}), \quad \mathrm{x} \in \partial\Omega. \qquad X \subsetneq V \quad Y = L^2(\Omega)$$

Assumptions:

Norm relations: $C_1\|u\|_V \leq \|\mathcal{L}^q u\|_Y + \|u\|_Z \leq C_2\|u\|_X, \quad \forall u \in X,$

Unique solution to PDE: $u_{pde} \in V$

Loss function: $\mathcal{J}_{\tau,K,N_b}(v) = \|\pi_K\left(\mathcal{L}^q v - f\right)\|_Y^2 + \tau\,\|i_{N_b}(v-h)\|_Z^2$

i_N : interpolation operat
π_N : projection operator
u_{NN} : minimizier of $\mathcal{J}_{\tau,}$
$\mathcal{N}_{L,N}$: DNNs with depth
and width N

Theorem: Assume that $2\,\|\pi_N\left(\mathcal{L}^q v - f\right)\|_Y \geq \|\mathcal{L}^q v - f\|_Y$ for all v in a compact subset $Y_c \cap \mathcal{N}_{L,N}$. Assume also that $C_3\|v\|_Z \geq 2\,\|i_{N_b} v\|_Z \geq \|v\|_Z$ for some compact subset Z_c of $Z \cap \mathcal{N}_{L,N}$. Then, the following estimates hold

$$C_1\,\|u_{NN} - u_{\mathrm{pde}}\,\|_V \leq \left(2 + 2\tau^{-1}\right)^{1/2} \mathcal{J}_\tau^{1/2}\left(u_{NN}\right) \leq \left(4 + 4\tau^{-1}\right)^{1/2} \mathcal{J}_{\tau,K,N_b}^{1/2}\left(u_{NN}\right),$$

$$C_1\,\|u_{NN} - u_{\mathrm{pde}}\,\|_V \leq (2+2\tau)^{1/2}\,(1+C_3)\,C_2\left(8 + 8\tau^{-1}\right)^{1/2}\left(\inf_{v \in \mathcal{N}_{L,N}} \|v - u_{\epsilon,\,\mathrm{pde}}\,\|_X + C\epsilon\right).$$

the mollifier approximation $u_{\epsilon,\,\mathrm{pde}}$ of u_{pde} has an accuracy $C\epsilon$.

ror Estimate for Elliptic Equations

Consider the following elliptic problem

$$-\Delta u = f, \quad x \in \Omega, \quad u = g, \quad x \text{ on } \partial\Omega.$$

The least-squares formulation

$$J_\tau(u) = \int_\Omega (f + \Delta u)^2 + \tau \int_{\partial\Omega} (g - u)^2, \quad \tau > 0.$$

The loss function is

$$J_\tau^M(u_{NN}) = \frac{1}{M_r} \sum_{k=1}^{M_r} |f + \Delta u_{NN}|^2 + \tau \frac{1}{M_b} \sum_{j=1}^{M_b} |g - u_{NN}|^2 .$$

or Estimate for Elliptic Equations

Let Ω be a smooth domain (at least C^2). For the elliptic equation/operator, we have the the following norm relation (stability)

$$C_1\|v\|_{H^{1/2}} \leq \|-\Delta v\|_{L^2(\Omega)} + \|v\|_{L^2(\partial\Omega)} \leq C_2\|v\|_{H^2}, \forall v \in H^2(\Omega)$$

In the place of v, we use $u - u_{NN}$, where u is the solution to the elliptic equation. Then we have

$$C_1 \|u - u_{NN}\|_{H^{1/2}} \leq \|-\Delta (u - u_{NN})\|_{L^2(\Omega)} + \|(u - u_{NN})\|_{L^2(\partial\Omega)} = \|f + \Delta u_{NN}\|_{L^2(\Omega)} + \|g - u_{NN}\|_{L^2(\partial\Omega)} .$$

Let $\tau = 1$. The generalization error is then bounded by

$$\begin{aligned}
\|u - u_{NN}\|_{H^{1/2}} &\leq C_1^{-1} \left(\|f + \Delta u_{NN}\|_{L^2(\Omega)} + \|g - u_{NN}\|_{L^2(\partial\Omega)} \right) \\
&\leq C_1^{-1}\sqrt{2} \left(\|f + \Delta u_{NN}\|^2_{L^2(\Omega)} + \|g - u_{NN}\|^2_{L^2(\partial\Omega)} \right)^{1/2}, \quad \left(\text{Applying } a + b \leq \sqrt{2\left(a^2 + b^2\right)} \right) \\
&= C_1^{-1}\sqrt{2}\sqrt{J_1\left(u_{NN}\right)} \\
&= C_1^{-1}\sqrt{2}\sqrt{\left(J_1\left(u_{NN}\right) - J_1^M\left(u_{NN}\right)\right) + J_1^M\left(u_{NN}\right)} \\
&\leq C_1^{-1}\sqrt{2} \left(\sqrt{\underbrace{\left|J_1\left(u_{NN}\right) - J_1^M\left(u_{NN}\right)\right|}_{\text{numerical integration error}}} + \sqrt{\underbrace{J_1^M\left(u_{NN}\right)}_{\text{training error}}} \right). \quad (\text{ Applying } \sqrt{a+b} \leq \sqrt{a} + \sqrt{b}, a, b \geq 0)
\end{aligned}$$

stimates on the Generalization Error of PINNs for Approximating PDEs

Let X, Y be Banach spaces with $Y = L^p(\mathbb{D}; \mathbb{R}^m)$, with $1 \leq p < \infty$

$\boldsymbol{X}^* \subset \boldsymbol{X}, \boldsymbol{Y}^* \subset \boldsymbol{Y}$

$\mathbb{D} = D$ or $\mathbb{D} = D \times (0, T)$, with $D \subset \mathbb{R}^d$

Abstract PDE is

$$\mathcal{D}(\mathrm{u}) = \mathrm{f},$$

$\mathcal{D} : X^* \mapsto Y^*$ is the Differential operator.

$f \in Y^*$ is the input.

Boundary (Initial) conditions are implicit.

AIM is to ensure small Total Error:

$$\mathcal{E}(\theta) := \|\mathrm{u} - \mathrm{u}_\theta\|_p$$

Mishra S, Molinaro R. Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs. arXiv preprint arXiv:2006.16144. 2020 Jun 29.

stimates on the Generalization Error of PINNs for Approximating PDEs

PINNs are minimizers of $\|\mathcal{R}_\theta\|_Y^p \sim \int_{\mathbb{D}} |\mathcal{R}_\theta(y)|^p \, dy$

Replace Integral by Quadrature !

Let $\mathcal{S} = \{y_i\}_{1 \le i \le N}$ be quadrature points in $\mathbb{D}$, with weights w_i

PINN for approximating PDE is defined as $u^* = u_{\theta^*}$ such that

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{i=1}^{N} w_i \left|\mathcal{R}_\theta\left(y_i\right)\right|^p$$

On the other hand, PINNs minimize PDE Residual:

$$\varepsilon_G(\theta) = \|\mathcal{R}_\theta\|_p = \|\mathcal{D}\left(u_\theta\right) - f\|_p$$

In practice, we only have access to Training Error:

$$\mathcal{E}_T(\theta) = \left(\sum_{i=1}^{N} w_i \left|\mathcal{R}_\theta\left(y_i\right)\right|^p\right)^{\frac{1}{p}}$$

Mishra S, Molinaro R. Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs. arXiv preprint arXiv:2006.16144. 2020 Jun 29.

stimates on the Generalization Error of PINNs for Approximating PDEs

For sufficiently smooth u solving $\mathcal{D}(\mathrm{u}) = \mathrm{f}$ observe that

$$\mathcal{E}_G(\theta) = \|\mathcal{D}(\mathrm{u}_\theta) - \mathrm{f}\|_p = \|\mathcal{D}(\mathrm{u}_\theta) - \mathcal{D}(\mathrm{u})\|_p \leq C(\mathrm{u}, \mathrm{u}_\theta) \|\mathrm{u} - \mathrm{u}_\theta\|_{W^{s,p}}$$

Here s depends on the number of derivatives in the Differential Operator $\mathcal{D}$.

Novel DNN approximation results in high-order Sobolev spaces:

For example in (DeRyck, Lanthaler, Mishra, 2021) can be used to conclude:

$$\exists \hat{\theta} \in \Theta, \quad \left\|\mathrm{u} - \mathrm{u}_{\hat{\theta}}\right\|_{W^{s,p}} < \epsilon$$

smoothness of u $\Rightarrow$ small PINN Residuals.

Mishra S, Molinaro R. Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs. arXiv preprint arXiv:2006.16144. 2020 Jun 29.

stimates on the Generalization Error of PINNs for Approximating PDEs

Sufficient Conditions of Mishra, Molinaro, 2021:

Coercivity of the PDE $\mathcal{D}u = f$: for any $u, \bar{u} \in X^*$:

$$\|\mathrm{u} - \bar{\mathrm{u}}\|_p \leq C_{pde}(\bar{\mathrm{u}}, \mathrm{u}) \|\mathcal{D}(\bar{\mathrm{u}}) - \mathcal{D}(\mathrm{u})\|_p$$

Coercivity $\Rightarrow$ Bounds in terms of Residuals as,

$$\begin{aligned}
\mathcal{E}(\theta) &= \|\mathrm{u}_\theta - \mathrm{u}\|_p , \\
&\leq C_{pde}\left(\mathrm{u}, \mathrm{u}_\theta\right) \|\mathcal{D}\left(\mathrm{u}_\theta\right) - \mathcal{D}(\mathrm{u})\|_p \qquad \text{(Coercivity)}, \\
&\leq C_{pde}\left(\mathrm{u}, \mathrm{u}_\theta\right) \|\mathcal{D}\left(\mathrm{u}_\theta\right) - f\|_p \qquad \text{as } \mathcal{D}(\mathrm{u}) = f, \\
&\leq C_{pde}\left(\mathrm{u}, \mathrm{u}_\theta\right) \mathcal{E}_G(\theta) \qquad \text{(Definition of } \mathcal{E}_G\text{)}
\end{aligned}$$

Training Error $\mathcal{E}_T$ is Quadrature Approximation of $\mathcal{E}_G$:

$$\mathcal{E}_G \leq \mathcal{E}_T + C_{\text{quad}} \left(u_{\theta^*}\right)^{\frac{1}{p}} N^{-\frac{\alpha}{p}} \qquad \text{quadrature error},$$

Mishra S, Molinaro R. Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs. arXiv preprint arXiv:2006.16144. 2020 Jun 29.

Strategy for PINN Error Bounds

- Use smoothness of exact solution to u of PDE $\mathcal{D}(\mathrm{u}) = f$
- And DNN approximation results in high-order Sobolev spaces to show that:

$$\exists \theta \in \Theta : \quad \mathcal{E}_G(\theta), \mathcal{E}_T(\theta) \leq \underline{C}\,(\mathrm{u}, \mathrm{u}_\theta)\, \|\mathrm{u} - \mathrm{u}_\theta\|_{W^{s,p}}.$$

- Use Coercivity of a given PDE to show that

$$\|\mathrm{u} - \mathrm{u}_\theta\|_p \leq \bar{C}\,(\mathrm{u}, \mathrm{u}_\theta)\, \mathcal{E}_G(\theta), \quad \forall \theta \in \Theta.$$

- Use Quadrature bounds to show that,

$$\mathcal{E}_G \leq \mathcal{E}_T + C_{\mathrm{quad}}\ (u_{\theta^*})^{\frac{1}{\rho}}\ N^{-\frac{\alpha}{p}}$$

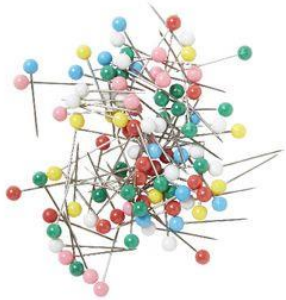
- Prove explicit growth bounds on the constants $\underline{C}, \bar{C}, C_{quad}$ in terms of Neural Network architecture and number of collocation points.

Mishra S, Molinaro R. Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs. arXiv preprint arXiv:2006.16144. 2020 Jun 29.

Summary

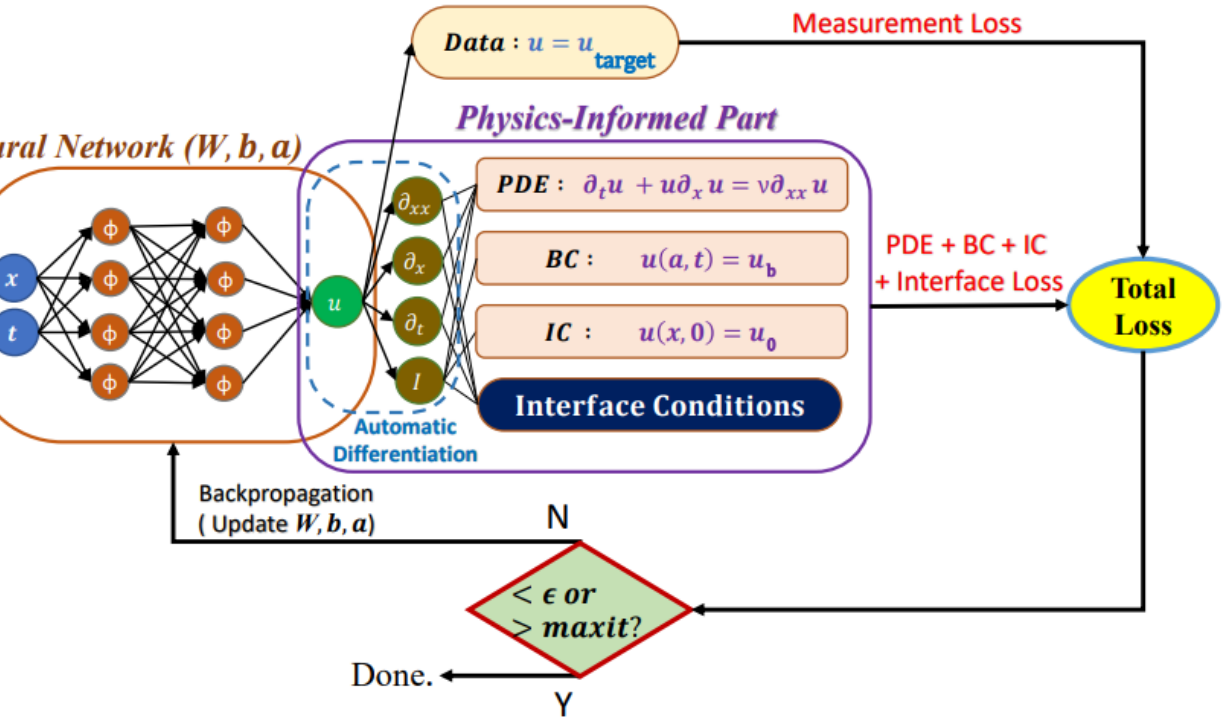
- PINN is simple and can be implemented in a few lines of code for any PDE
- Boundary conditions need special treatment and can be implemented via self-adaptive or dynamic weights
- Weighted residual methods can be implemented following a Petrov-Galerkin projection
- Domain decomposition can enhance the accuracy and flexibility of PINNs
- PINN converges for boundary values problems even in the absence of BCs
- Several works have obtained a priori and a posteriori errors for PINNs
- There are three main errors: Approximation, Generalization and Optimization

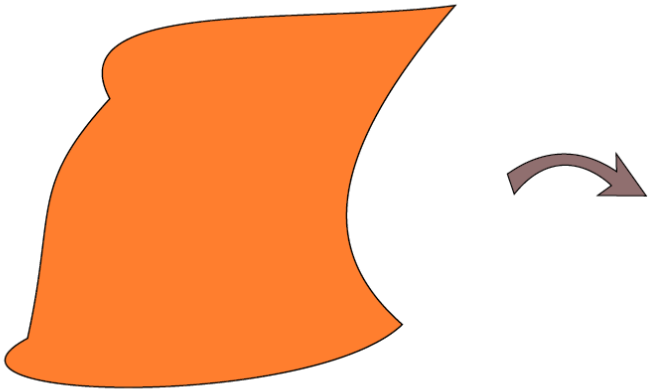
An Alphabet of PINNs:

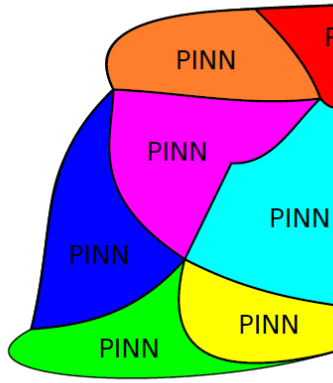


- cPINNs: conservative PINNs
- vPINNs: variational PINNs
- pPINNs: parareal PINNs
- sPINNs: stochastic PINNs
- fPINNs: fractional PINNs
- BPINNs: Bayesian PINNs
- xPINNs: eXtended PINNs

NNs with Domain Decomposition







$$\mathcal{L}(\tilde{\Theta}) = \frac{1}{N_u}\sum_{i=1}^{N_u}\left|u^i_{\text{target}} - u_{\tilde{\Theta}}\left(x^u_i\right)\right|^2 + \frac{1}{N_f}\sum_{i=1}^{N_f}\left|\mathcal{F}_{\tilde{\Theta}}\left(x^f_i\right)\right|^2 + \text{Interface Loss}$$

Conservative PINNs (cPINNs): Application to Conservation Laws

PINNs: Interface conditions in the " q^{th} " subdomain.

$$\text{MSE}_{\text{flux}} = \sum_{\forall q^+} \left(\frac{1}{N_{l_q}} \sum_{i=1}^{N_{l_q}} \left| f_q\left(u\left(\mathbf{x}_{l_q}^{(i)}\right)\right) \cdot \mathbf{n} - f_{q^+}\left(u\left(\mathbf{x}_{l_q}^{(i)}\right) \cdot \mathbf{n}\right) \right|^2 \right)$$

where f is the flux and $q = 1, 2, \cdots, N_{sd}$.

$$\text{MSE}_{u_{\text{avg}}} = \sum_{\forall q^+} \left(\frac{1}{N_{l_q}} \sum_{i=1}^{N_{l_q}} \left| u_{\tilde{\Theta}_q}\left(\mathbf{x}_{l_q}^{(i)}\right) - \left\{\left\{ u_{\tilde{\Theta}_q}\left(\mathbf{x}_{l_q}^{(i)}\right) \right\}\right\} \right|^2 \right)$$

$\{\} \rightarrow$ average value

The parameterized nonlinear conservation law is given by

$$u_t + f[u; \lambda]_{\mathbf{x}} = 0, x \in \Omega \subset \mathbb{R}^d, t > 0,$$

with initial $u(x, 0) = u_0$ and appropriate boundary conditions. $u(t, x)$ denotes the solution of conservation laws, $f[\lambda]$ is a nonlinear flux parametrized by λ. As an example, the one-dimensional viscous Burgers equation flux is given as $f = u^2/2 - \lambda u_x$, where the parameter λ plays the role of viscosity coefficient.

- Jagtap AD, Kharazmi E, Karniadakis GE. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering. 2020 Jun 15;365:113028

Conservative PINNs (cPINNs): Application to Conservation Laws

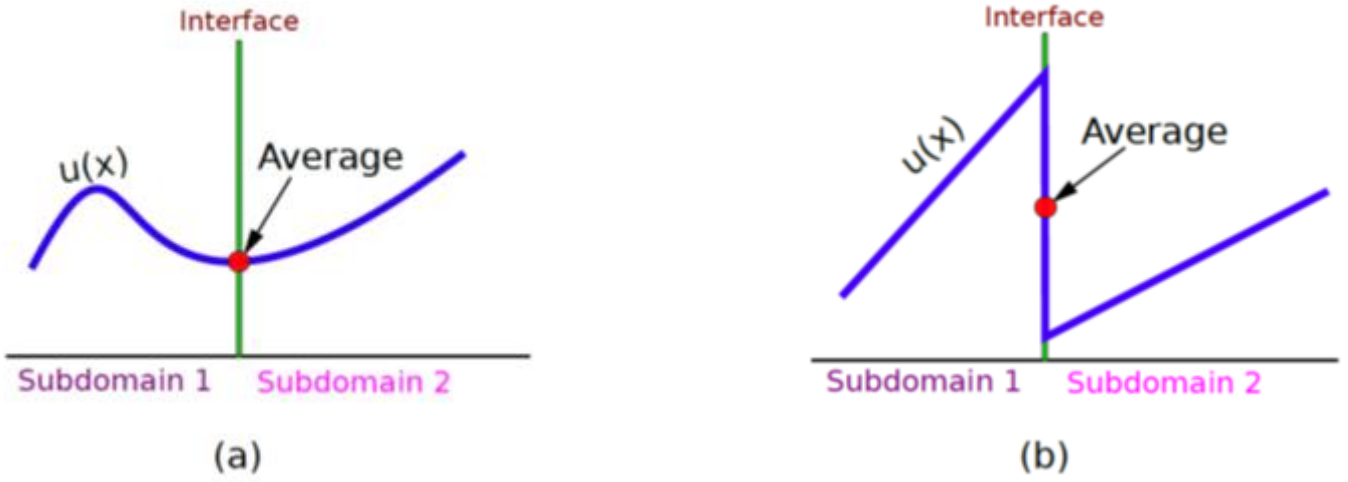
PINNs: Interface conditions in the " q^{th} " subdomain.

$$\text{SE}_{\text{flux}} = \sum_{\forall q^+} \left(\frac{1}{N_{l_q}} \sum_{i=1}^{N_{l_q}} \left| f_q \left(u \left(\mathbf{x}_{l_q}^{(i)} \right) \right) \cdot \mathbf{n} - f_{q^+} \left(u \left(\mathbf{x}_{l_q}^{(i)} \right) \cdot \mathbf{n} \right) \right|^2 \right)$$

where f is the flux and $q = 1, 2, \cdots, N_{sd}$.

$$\text{SE}_{u_{\text{avg}}} = \sum_{\forall q^+} \left(\frac{1}{N_{l_q}} \sum_{i=1}^{N_{l_q}} \left| u_{\tilde{\Theta}_q} \left(\mathbf{x}_{l_q}^{(i)} \right) - \left\{ \left\{ u_{\tilde{\Theta}_q} \left(\mathbf{x}_{l_q}^{(i)} \right) \right\} \right\} \right|^2 \right)$$

$\{\} \rightarrow$ average value



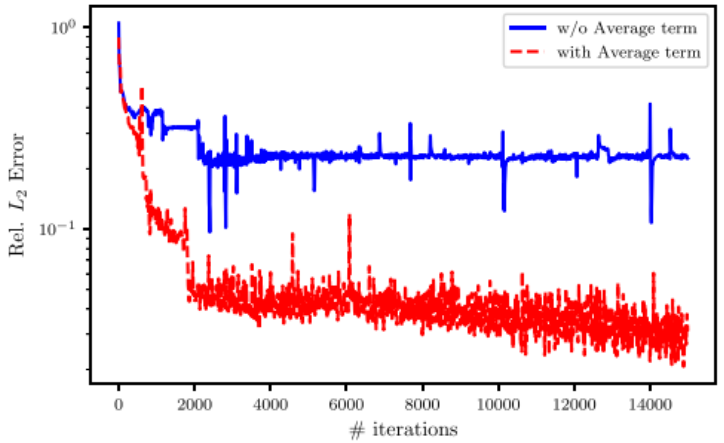
- Jagtap AD, Kharazmi E, Karniadakis GE. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering. 2020 Jun 15;365:113028

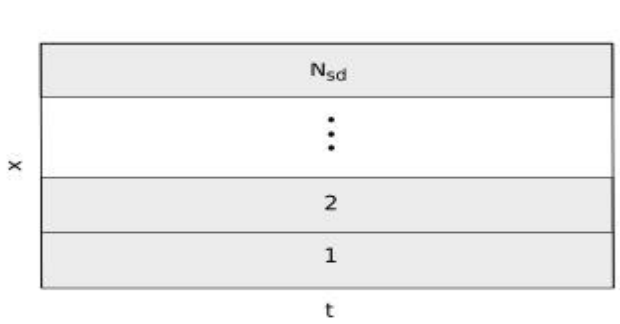
PINN Results: Burgers Equation

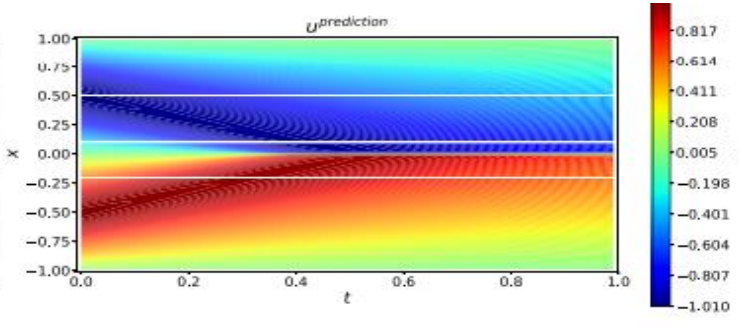
$+ uu_x = \nu u_x x, x \in \mathbf{R}, t > 0$ with IC : $u(x,0) = -\sin(\pi x)$ and **BC** :$u(t,1) = u(t,-1) = 0$.

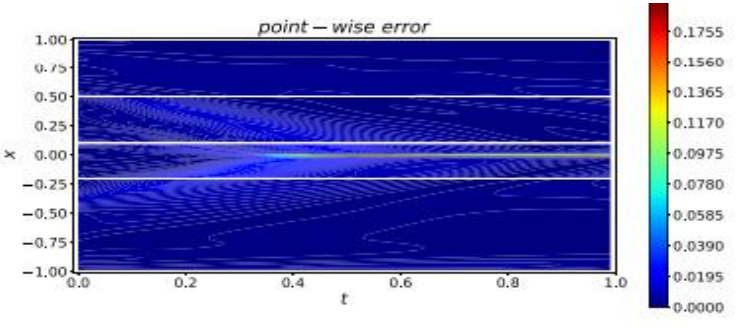
Subdomain number	1	2	3	4
# Layers	2	6	3	2
# Neurons	20	30	25	20
# Residual points	6000	8000	4000	4000
Adaptive activation function	cos	sin	tanh	sin

arning rate : 8e-4, Optimizer : Adam







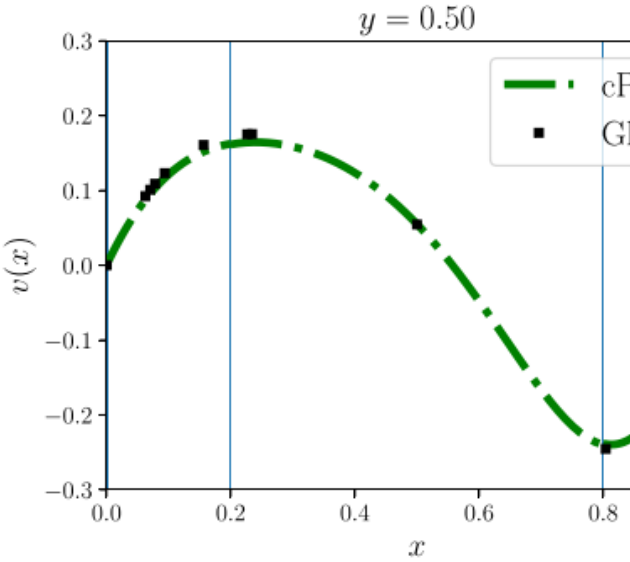


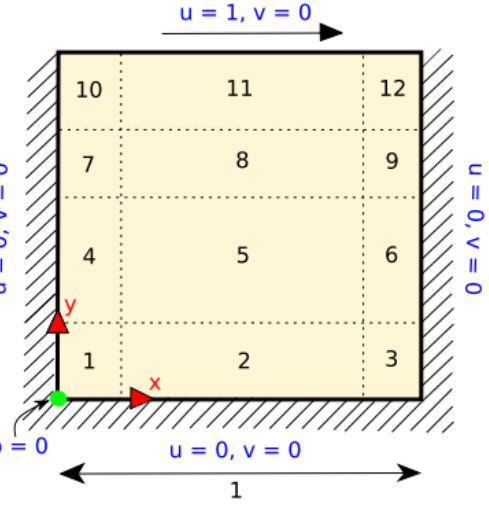
cPINN Results: 2D Incompressible Navier Stokes Equation

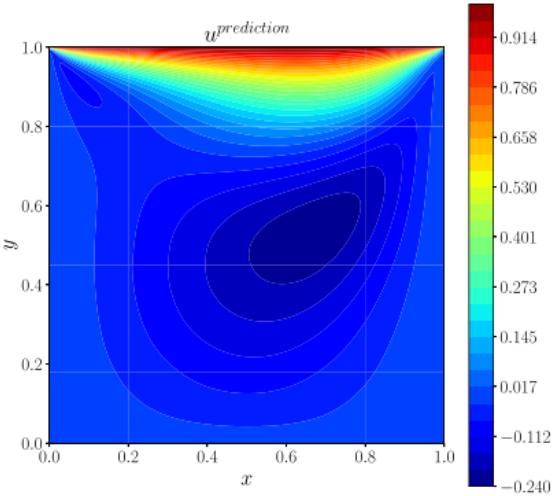
$$\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u}, \quad \text{in } \Omega$$
$$\nabla\cdot\mathbf{u} = 0, \quad \text{in } \Omega$$

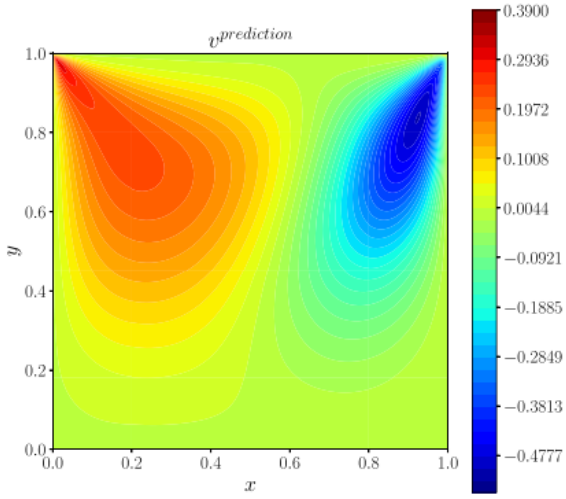
SD No.	1	2	3	4	5	6	7	8	9	10	11	12
# L	6	6	6	6	4	6	6	4	6	6	6	6
# N	20	20	20	20	20	20	20	20	20	20	20	20
#R.pts	2.5k	5k	2.5k	2.5k	5k	2.5k	2.5k	5k	2.5k	2.5k	5k	2.5k

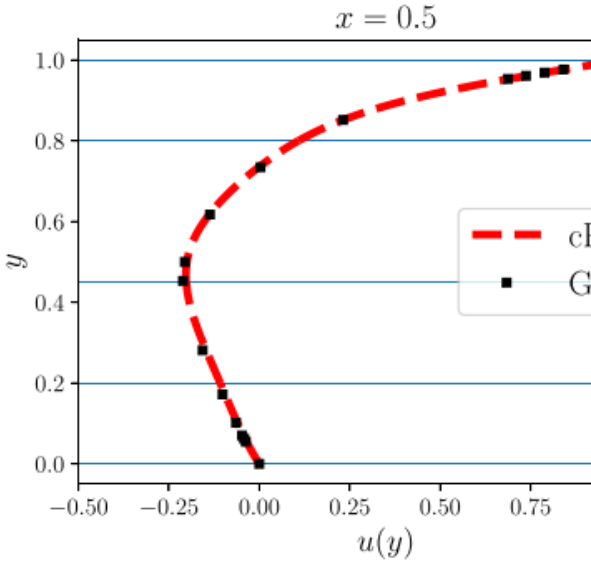
Learning rate : 6e-4, Optimizer : Adam, Adaptive Activation Fun. : tanh









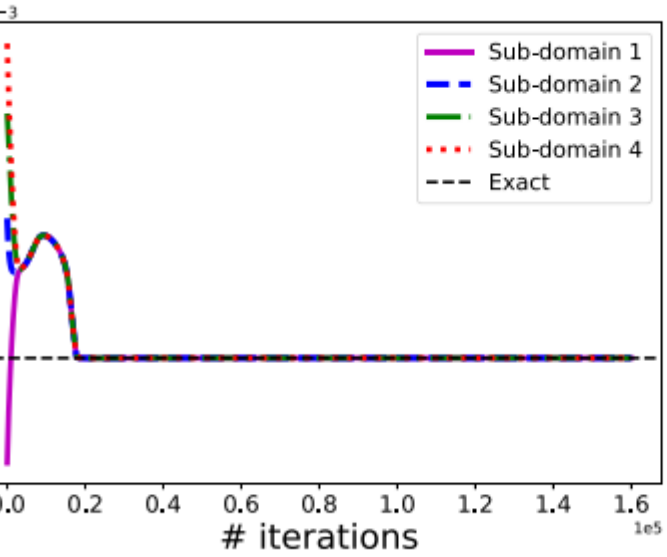


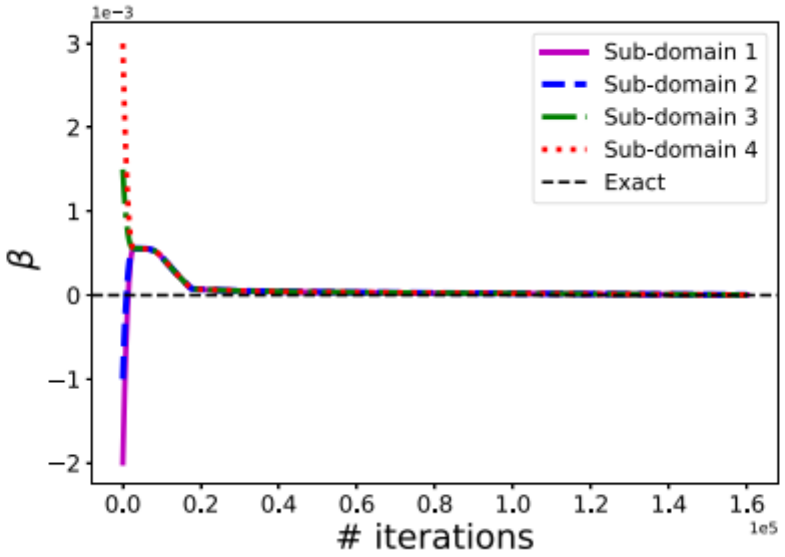
cPINN Results: Inverse 2D Burgers Equation

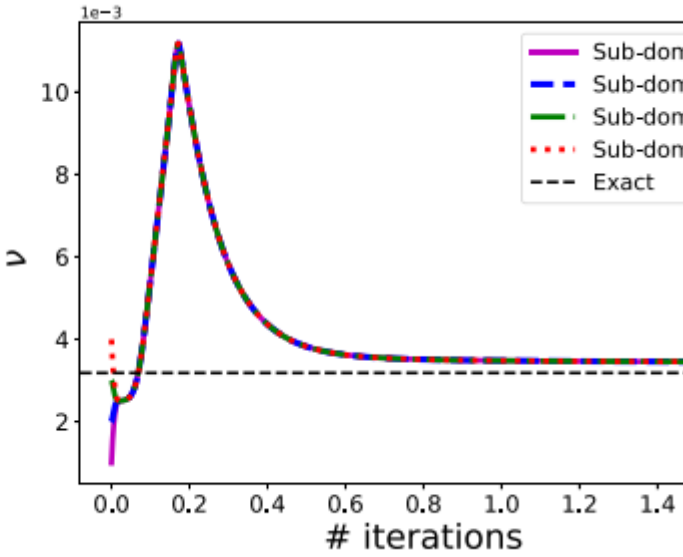
We have the following equation with some unknown parameters α, β, ν.

$$u_t + uu_x - \beta u_x - \nu u_{xx} + \alpha u_{xxx} = 0, \quad x \in \Omega \subset \mathbb{R}, t > 0,$$

Aim : Identify all the terms α, β, ν with data given by the viscous Burgers equation.







Extended PINNs (XPINNs)

XPINNs : Interface conditions in the " q^{th} " subdomain.

$$\text{MSE}_{u_{\text{avg}}} = \sum_{\forall q^+} \left(\frac{1}{N_{l_q}} \sum_{i=1}^{N_{l_q}} \left| u_{\tilde{\Theta}_q}\left(\mathbf{x}_{l_q}^{(i)}\right) - \left\{ \left\{ u_{\tilde{\Theta}_q}\left(\mathbf{x}_{l_q}^{(i)}\right) \right\} \right\} \right|^2 \right)$$

$\{\} \rightarrow$ average value

$$\text{MSE}_{\mathcal{R}} = \sum_{\forall q^+} \left(\frac{1}{N_{l_q}} \sum_{i=1}^{N_{l_q}} \left| \mathcal{F}_{\tilde{\Theta}_q}\left(\mathbf{x}_{l_q}^{(i)}\right) - \mathcal{F}_{\tilde{\Theta}_{q^+}}\left(\mathbf{x}_{l_q}^{(i)}\right) \right|^2 \right)$$

$+$

Additional continuity conditions

Additional advantages over cPINN

(1) Extension to any differential equation(s)

(2) Generalized space-time domain decomposition

(3) Simple interface conditions

omparison of PINN, cPINN and XPINN frameworks

	PINN	cPINN	XPINN
Spatial Domain Decomposition	×	✓	✓
Parallelization capacity	×	✓	✓
Localized representation capacity	×	✓	✓
Efficient hyperparameter adjustment	×	✓	✓
Applicability	Any DEs	Conservation laws	Any DEs
Interface conditions	–	Complex	Simple
Spatio-Temporal Domain Decomposition	×	×	✓

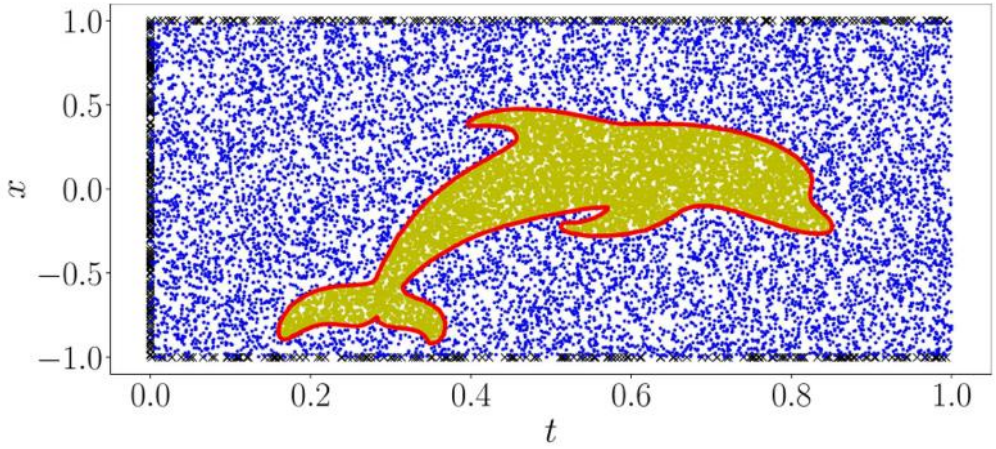
- Jagtap AD, Karniadakis GE. Extended physics-informed neural networks (xPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. Communications in Computational Physics. 2020 Nov 1;28(5):2002-41.

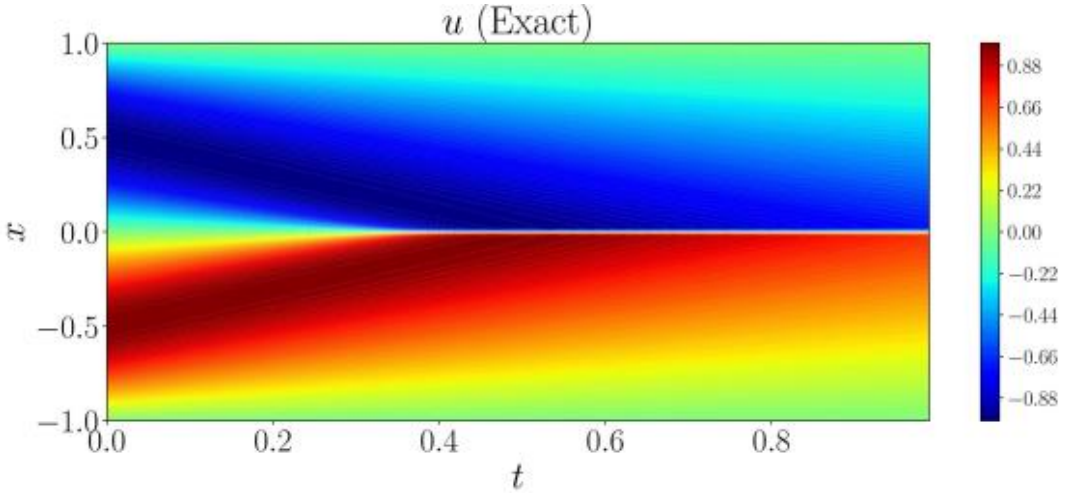
XPINN Results: Burgers Equation

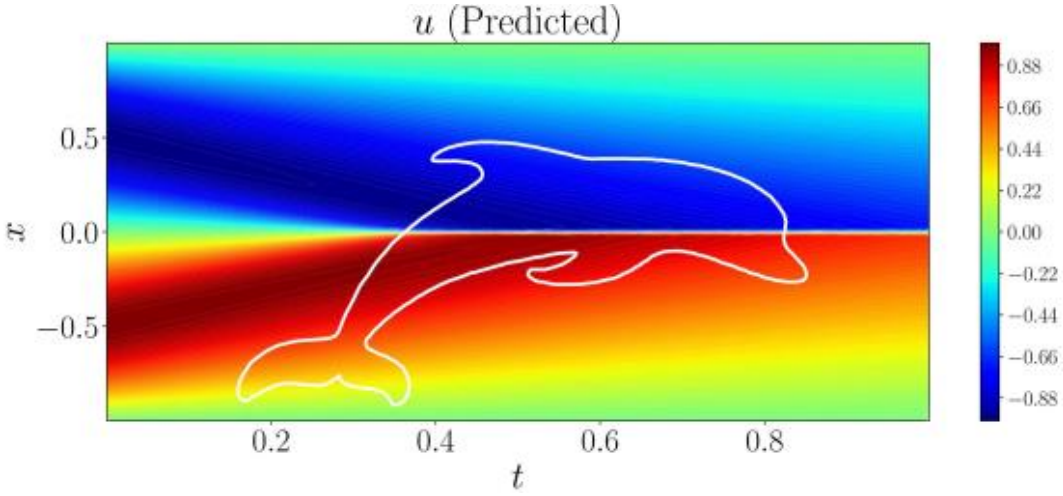
Subdomain number	1	2
# Layers	6	7
# Neurons	20	25
# Residual points	7000	3000
Adaptive Activation function	tanh	sin

Learning rate : $8e-4$, Optimizer : Adam

Relative L_2 error in the whole domain : $8.93265e-3$



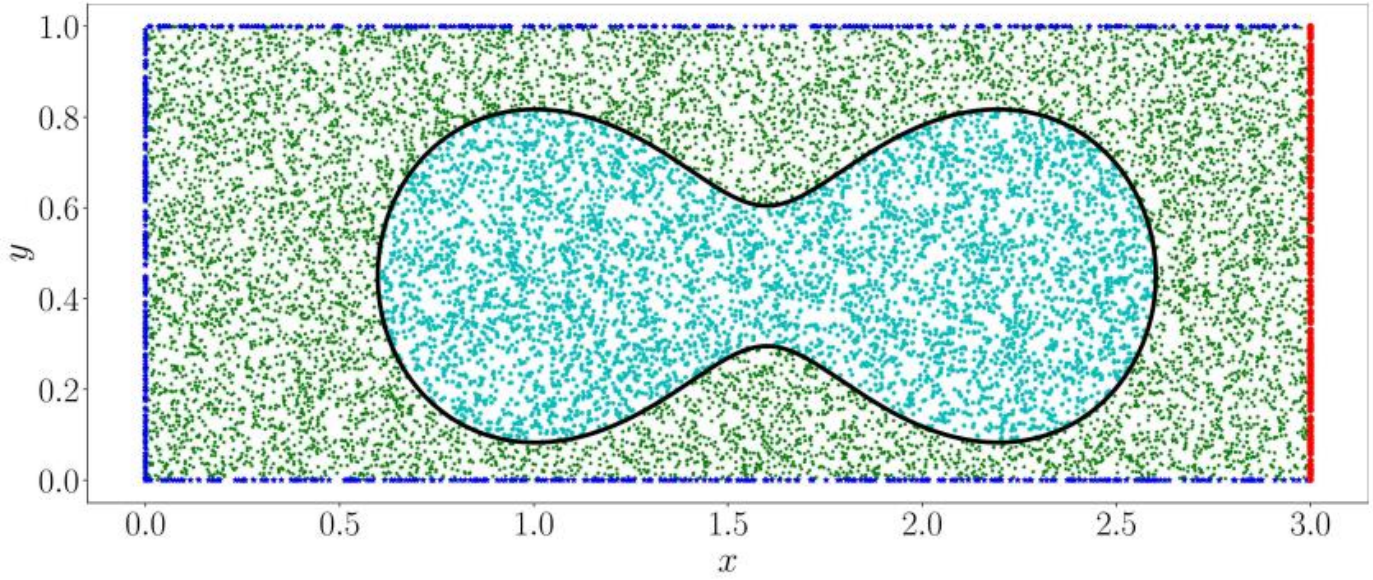




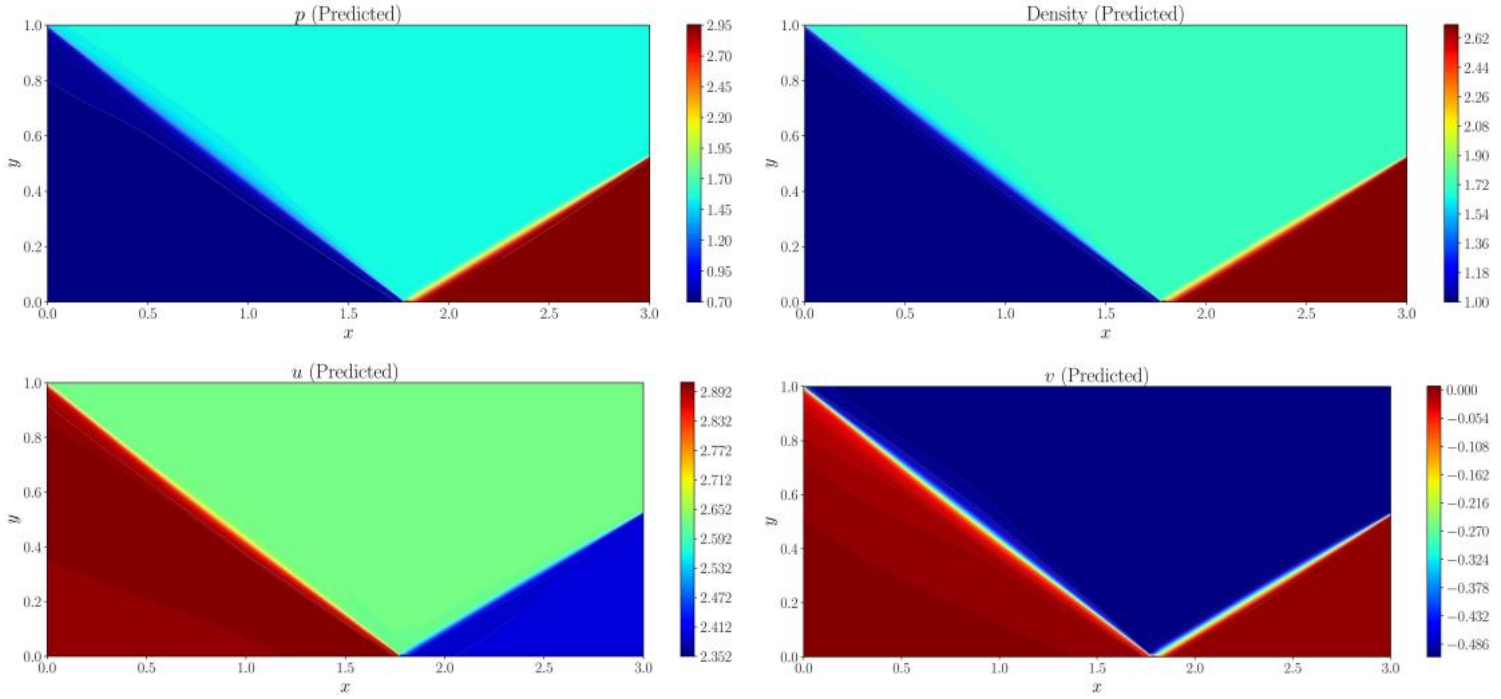
PINN Results: 2D Compressible Euler Equations

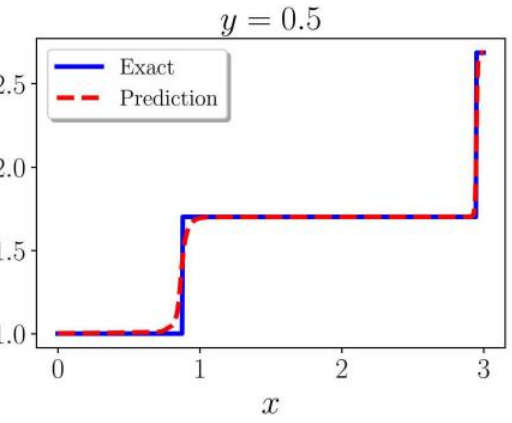
$+F_x+G_y=0, \quad (\mathbf{x},t)\in\Omega\times(0,T]\subset\mathbb{R}^2\times\mathbb{R}_+$, with appropriate initial and boundary conditions.

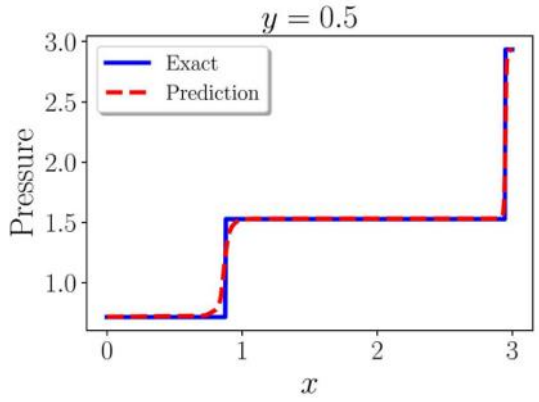
Subdomain number	1	2
# Layers	6	6
# Neurons	20	25
# Residual points	12000	8000
Adaptive activation function	sin	tanh

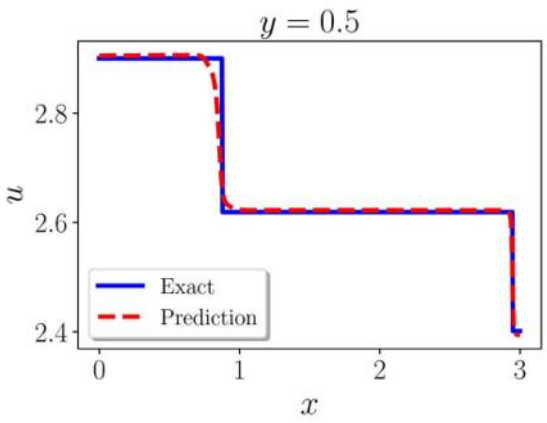


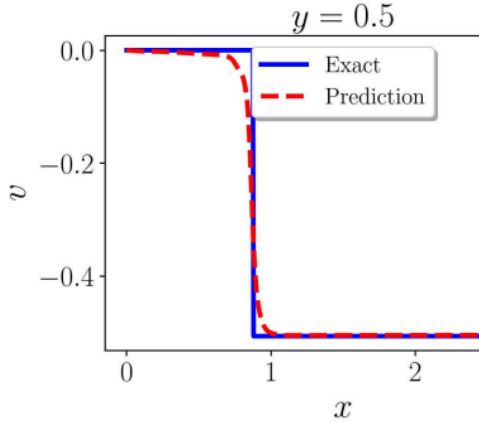
PINN Results: 2D Compressible Euler Equations











Generalization of PINNs vs XPINNs

For supervised learning in PINN and XPINN, generalization error is a measure of how accurately the model is able to predict outcome values for previously unseen data.

The train loss $R_S(\theta) = \frac{1}{n}\sum_{i=1}^{n} l\left(z_i, \theta\right)$, is the average loss over all training samples $\{z_i\}_{i=1}^{n}$, where θ can be model parameters of PINN and XPINN.

The test loss is $R_D(\theta) = E_z[l(z, \theta)]$, where z is drawn from an unknown test distribution.

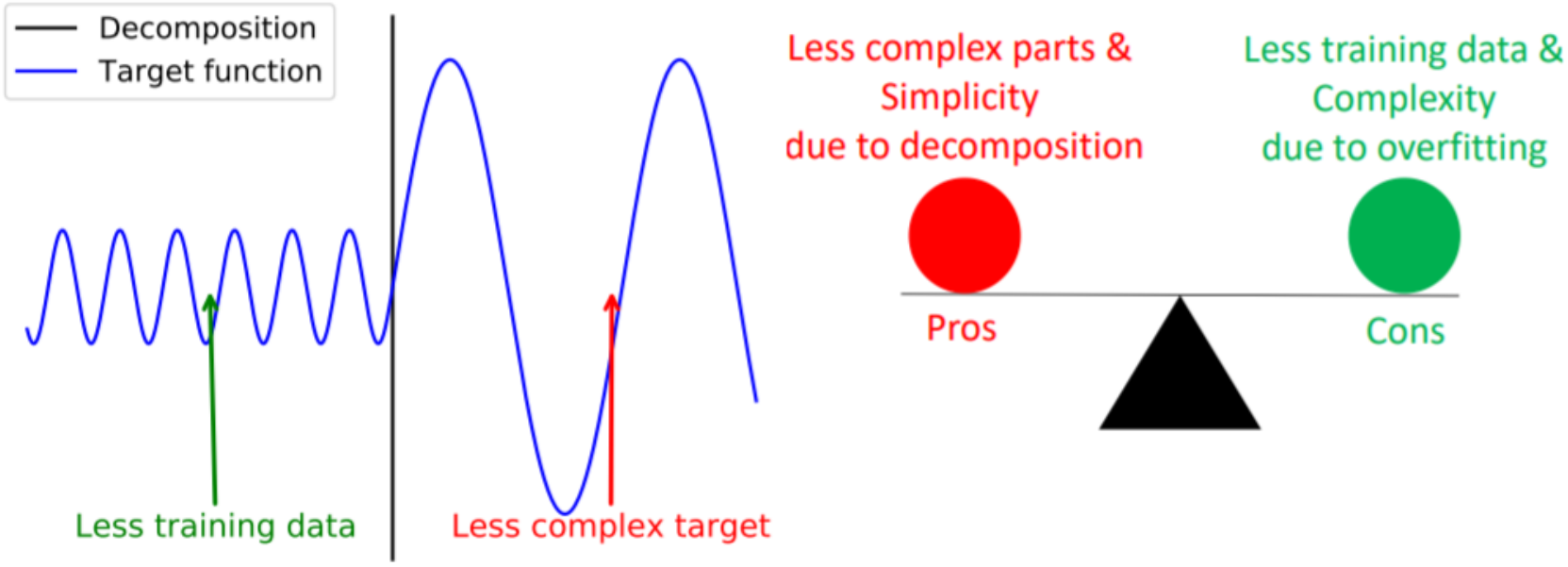
Convergence theory only guarantees small $R_S(\theta)$, but not small $R_D(\theta)$.

Thus, a small generalization gap $R_D(\theta) - R_S(\theta)$ is desired, to ensure that PINN and XPINN perform good over the entire domain of the PDE.

- Hu Z, Jagtap AD, Karniadakis GE, Kawaguchi K. When Do Extended Physics-Informed Neural Networks (XPINNs) Improve Generalization?. arXiv preprint arXiv:2109.09444. 2021 Sep 20.

hen Do XPINNs Improve Generalization?

- Domain decomposition in XPINNs introduces a tradeoff.
- It decomposes the complex PDE solution into several simple parts, which decreases the complexity needed to learn each part and boosts generalization.
- It leads to less training data being available in each subdomain, and hence such model is typically prone to overfitting and may become less generalizable.



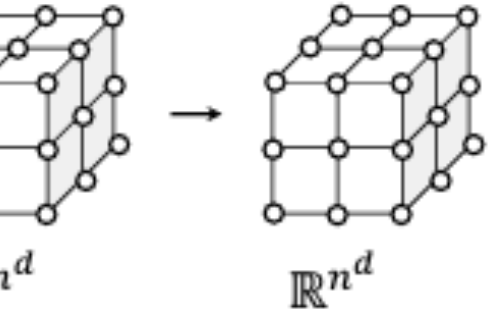
Separable PINNs

Mitigating the Curse of Dimensionality in Physics-Informed Neural Networks

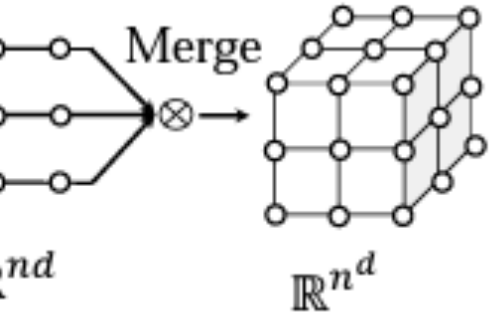
Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, Eunbyung Park

NeurIPS 2022 - DLDE Workshop

eparable PINNs

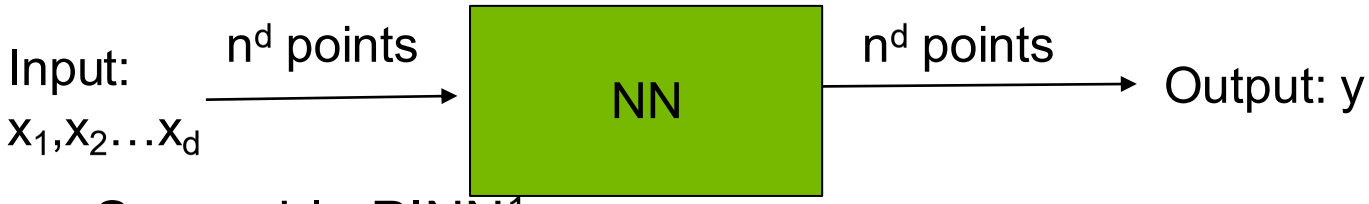


a) Non-separated

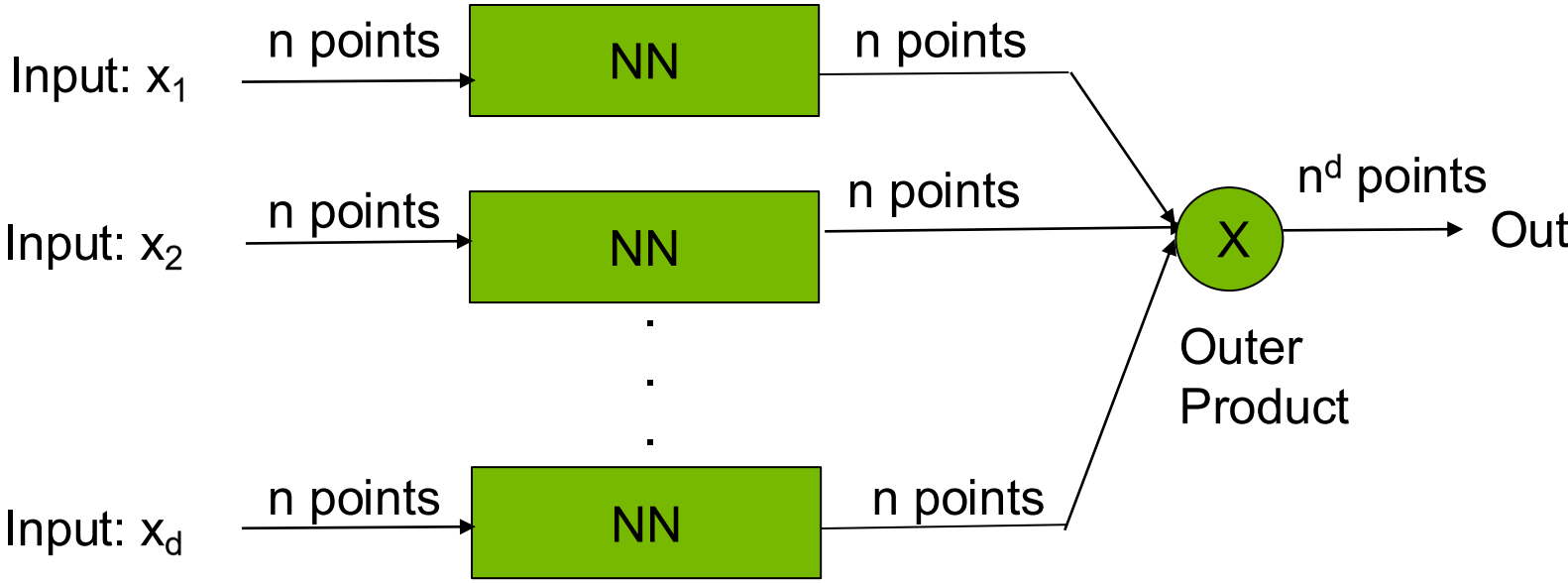


(b) Separated

- Vanilla PINN



- Separable PINN[1]

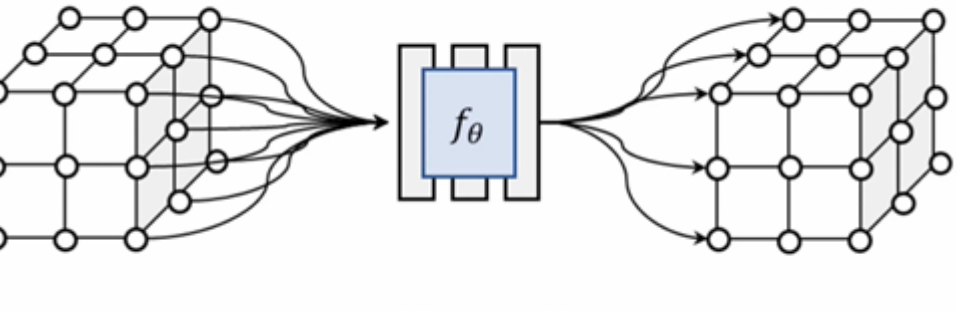


mber of collocation points in each dimension

[1]Cho, J., Nam, S., Yang, H., Yun, S. B., Hong, Y., & Park, E. (2022). Separable PINN: Mitiga
Curse of Dimensionality in Physics-Informed Neural Networks. arXiv preprint arXiv:2211.08

Forward/Reverse-Mode AD

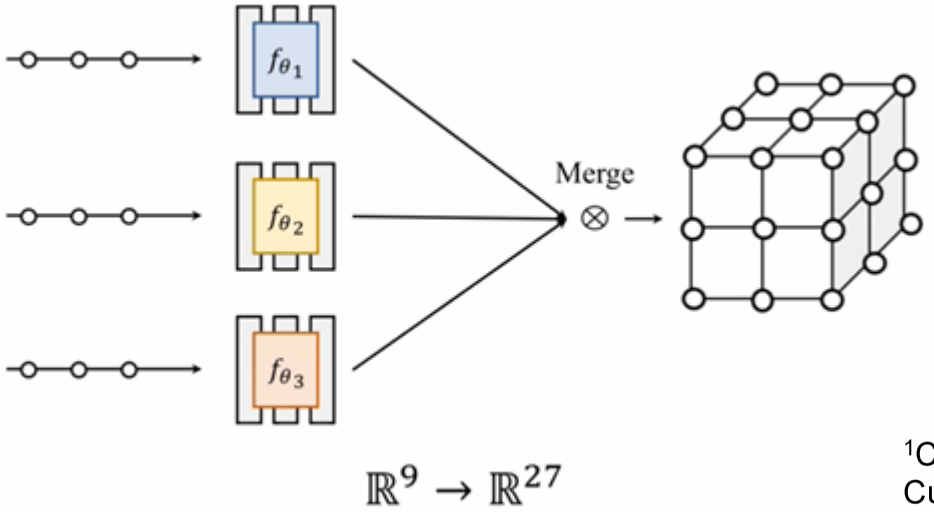
PINN



To compute $\frac{\partial u}{\partial x}$ using PINN:

1. Reverse mode: 27 backward passes
2. Forward mode: 27 forward passes

Separable PINN



To compute $\frac{\partial u}{\partial x}$ using SPINN[1]:

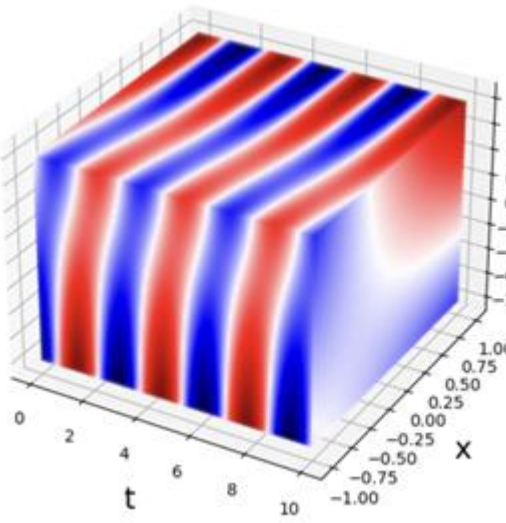
1. Reverse mode: 27 backward passes
2. Forward mode: 9 forward passes

[1]Cho, J., Nam, S., Yang, H., Yun, S. B., Hong, Y., & Park, E. (2022). Separable PINN: Mitiga
Curse of Dimensionality in Physics-Informed Neural Networks. arXiv preprint arXiv:2211.087

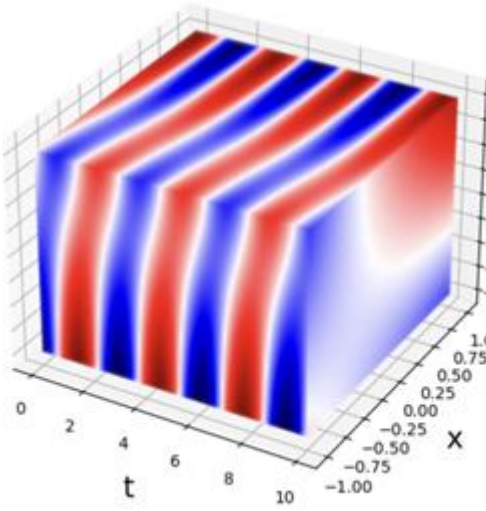
in-Gordon equation:

$$\partial_{tt}u - \Delta u + u^2 = f, \qquad x \in \Omega, t \in \Gamma$$
$$u(x,0) = x_1 + x_2, \qquad x \in \Omega$$
$$u(x,t) = u_{bc}(x), \qquad x \in \partial\Omega, t \in \Gamma,$$

INN (90^3)	0.0092±0.0022		
	0.0016±0.0017		
INN (180^3)	0.0009±0.0002		
INN (360^3)	0.0009±0.0003		



Ground Truth



SPINN[1]

With 40× less memory usage, SPINN finds more accurate solutions
56× faster than the baseline
180^3 collocation points showed the lowest error

Nam, S., Yang, H., Yun, S. B., Hong, Y., & Park, E. (2022). Separable PINN: Mitigating the Curse of Dimensionality
s-Informed Neural Networks. arXiv preprint arXiv:2211.08761.

From PINNs to PIKANs: A FAIR Comparison

- Shukla K, Toscano JD, Wang Z, Zou Z, Karniadakis GE. A comprehensive and FAIR comparison between MLP and KAN representations for differential equations and operator networks. arXiv preprint arXiv:2406.02917. 2024 Jun 5, CMAME, 2024.

Approximation Theory

https://developer.nvidia.com/teaching-kits

Multilayer Neural Networks

In the place of shallow neural networks, we can also use multilayer neural networks (feedforward), which can be proved by mathematical induction. The proof can also be based on the Kolmogorov Superposition Theorem.

Theorem (Kolmogorov Superposition Theorem) For every continuous function f of n variables on $[0,1]^n$, there exist n constants $\lambda_j > 0, j = 1,\ldots,n, \sum_{j=1}^{n} \lambda_j \leq 1$ and $2n+1$ strictly increasing continuous functions $\phi_i, i = 1,\ldots,2n+1$, which map $[0,1]$ to $[0,1]$, such that

$$f(x_1,\ldots,x_n) = \sum_{i=1}^{2n+1} g\left(\sum_{j=1}^{n} \lambda_j \phi_i(x_j)\right)$$

where $g \in C[0,1]$ depends on f.

Kolmogorov-Arnold Theory

1. Construction of refined versions to strengthen the connection to neural networks:
   - Reduce the number of one-dimensional functions
   - Smoothness conditions
2. Theory to practical application in deep learning and general machine learning:
   - Kolmogorov's mapping neural network
   - KANs

Version	Representation	Inner Function	Outer Func
KART (1957)	$\sum_{q=0}^{2d} g_q\left(\sum_{p=1}^{d} \psi_{p,q}(x_p)\right)$	$d(2d+1)$ $\psi_{p,q} \in C([0,1])$	$2d+1$ $g_q \in C(\mathbb{R}$
Lorentz (1962)	$\sum_{q=0}^{2d} g\left(\sum_{p=1}^{d} \lambda_p \psi_q(x_p)\right)$	$2d+1$ $\psi_q \in \mathrm{Lip}^{\alpha}; \in [0,1]; \nearrow$	1 $g \in C([0,$
Sprecher (1965)	$\sum_{q=0}^{m} g_q\left(\sum_{p=1}^{d} \lambda_p \psi(x_p + qa)\right)$	1 $\psi \in \mathrm{Lip}^{\log_\gamma 2}; \ [0,2] \to [0,2]; \nearrow$	$m+1$ $g_q \in C([0,2$
Braun (2009)	$\sum_{q=0}^{2d} g\left(\sum_{p=1}^{d} \lambda_p \psi(x_p + qa) + c_q\right)$	1 $\psi \in C(\mathbb{R}); \nearrow$	1 $g \in C(\mathbb{R}$
Schmidt-Hieber (2021)	$g\left(\sum_{p=1}^{d} 3^{1-p} \psi(x_p)\right)$ $f : \beta\text{-smooth}, \beta \in (0,1]$	1 $\psi \in C : [0,1] \to \mathcal{C}; \nearrow$	1 $g : \mathcal{C} \to \mathbb{R}; \alpha\text{-s}$
Ismayilova & Ismailov (2024)	Formula as in Braun (2009) with given $a(\gamma), \lambda_p(\gamma)$ $c_q = (2d+1)q$	1 $\psi \in \mathrm{Lip}^{\log_\gamma 2}; [0,2] \to [0,2]; \nearrow$	$f \in C \Rightarrow g$ f dis-$C \Rightarrow g$ $\|f\|_\infty, \|g\|_\infty$

cht-Nielsen, R. (1987, June). Kolmogorov's mapping neural network existence theorem. In Proceedings of the International nference on Neural Networks (Vol. 3, pp. 11-14). New York, NY, USA: IEEE press.

Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., ... & Tegmark, M. (2024). KAN: Kolmogorov-Arnold sworks. arXiv preprint arXiv:2404.19756.

Kolmogorov-Arnold Theory: Debate

- Pessimistic perspectives
  - "Kolmogorov's mapping neural network existence theorem for approximations of functions by networks is, at least in theory, sound, but the direct usefulness of this result is doubtful."
  - "Kolmogorov's theorem: an exact representation is hopeless in representation properties of networks."
- Optimistic perspectives
  - "One should sacrifice the exactness of representation by adopting an approximate version instead"

osi, F., & Poggio, T. (1989). Representation properties of networks: Kolmogorov's theorem is irrelevant. Neural Computation,), 465-469.
rková, V. (1991). Kolmogorov's theorem is relevant. Neural computation, 3(4), 617-622.
, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., ... & Tegmark, M. (2024). Kan: Kolmogorov-arnold tworks. arXiv preprint arXiv:2404.19756.
ps://www.youtube.com/watch?v=AUDHb-tnlB0&t=6s

Definition

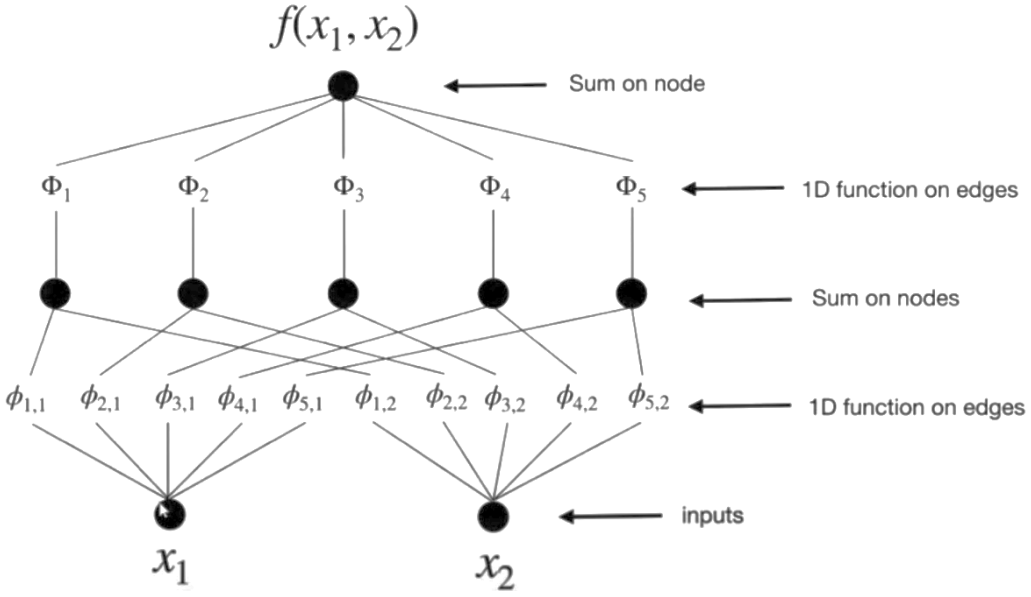
- KANs are function approximators
- What is unique about a KAN?

Learnable activation functions

✚ Activation functions on edges

Deeper than two layers.

KAN



https://www.youtube.com/watch?v=AUDHb-tnlB0&t=6s

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., ... & Tegmark, M. (2024). Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

Summary

Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)	(a) *fixed* activation functions on *nodes*; *learnable* weights on *edges*	(b) *learnable* activation functions on *edges*; sum operation on *nodes*
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)	(c) MLP(x); $\mathbf{W}_3$, σ_2, $\mathbf{W}_2$, σ_1, $\mathbf{W}_1$; *nonlinear, fixed*; *linear, learnable*; x	(d) KAN(x); $\mathbf{\Phi}_3$, $\mathbf{\Phi}_2$, $\mathbf{\Phi}_1$; *nonlinear, learnable*; x

Figure 0.1: Multi-Layer Perceptrons (MLPs) vs. Kolmogorov-Arnold Networks (KANs)

Kolmogorov-Arnold Representation theorem

- If f is a multivariate continuous function on a bounded domain, then f can be written as a finite composition of continuous functions of a single variable and the binary operation of addition. More specifically, for a smooth $f: [0,1]^n \to \mathbb{R}$.

$$f(\mathbf{x}) = f(x_1, \cdots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$

Univariate functions

- Problem? The univariate functions ($\phi_q, \phi_{q,p}$) can be non-smooth or fractal
  - $\phi_q, \phi_{q,p}$ may not be learnable!

https://www.youtube.com/watch?v=AUDHb-tnlB0&t=6s

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., ... & Tegmark, M. (2024). Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

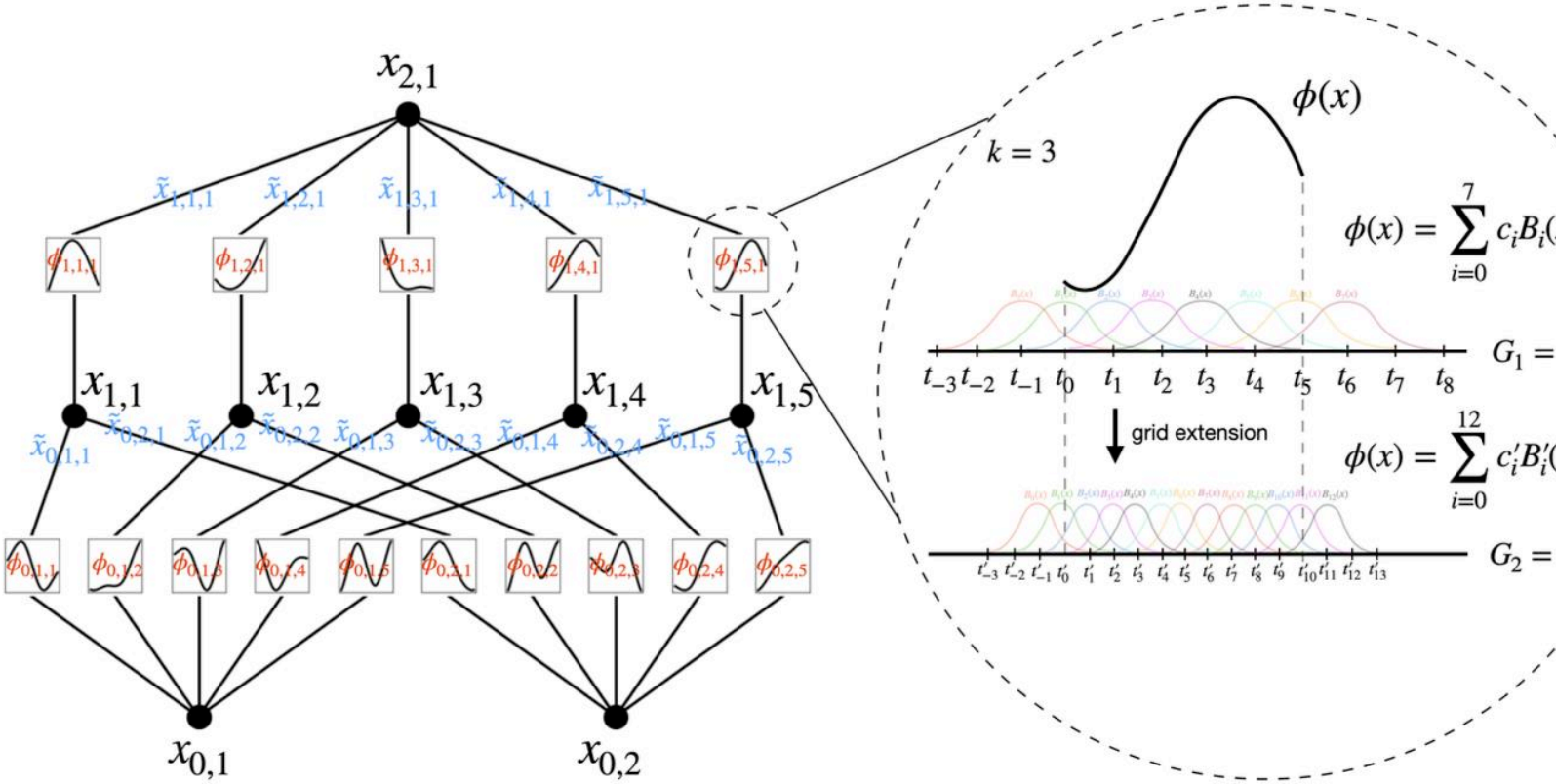
How do we learn Univariate functions?

esidual activation
unctions

$$x) = w(b(x) + spline(x))$$

$$spline(x) = \sum_i c_i B_i(x)$$

$$b(x) = \frac{x}{1 + e^{-x}}$$



https://www.youtube.com/watch?v=AUDHb-tnlB0&t=6s

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., ... & Tegmark, M. (2024). Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

Allen-Cahn equation

DE:

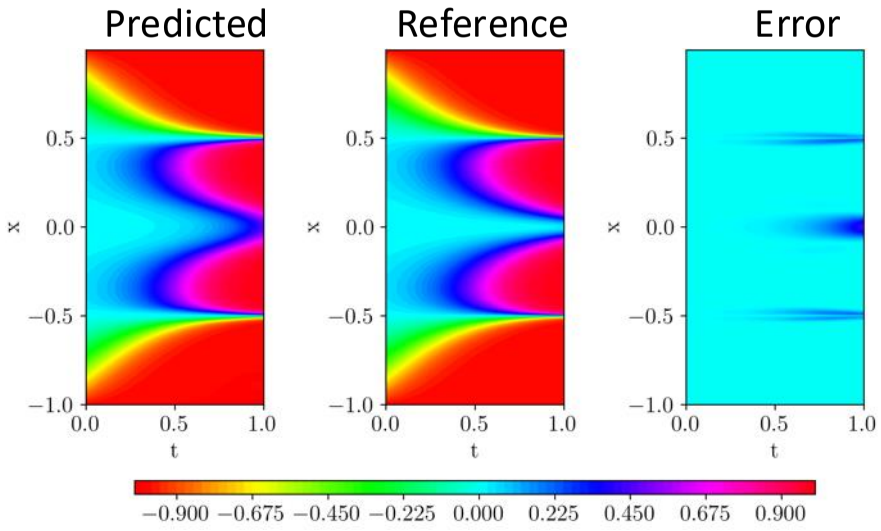
$$\frac{\partial u}{\partial t} = 10^{-4}\frac{\partial^2 u}{\partial x^2} - 5u^3 + 5u,$$

Cs/ICs:

$$u(0,x) = x^2\cos(\pi x), \quad \forall x \in [-1,1],$$

$$x+1) = u(t,x-1), \quad \forall t \geq 0 \quad \text{and} \quad x \in [-1,1].$$

cPIKAN



Method	$[N_i, N_b, N_f]$	No. of parameters	Relative L^2 error	Time: (ms/it)
INN with RBA	[200, 100, 50000]	50,049	1.51 %	22.93 ms
cPIKAN	[200, 100, 50000]	6,720	5.15 %	39.31 ms
IKAN with RBA	[200, 100, 50000]	56,720*	5.04 %	39.28 ms
PIKAN	[200, 100, 50000]	6721	58.39 %	2633.12 ms

vier-Stokes Equations

E:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0,$$

$$\frac{u}{x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right),$$

$$\frac{v}{x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right),$$

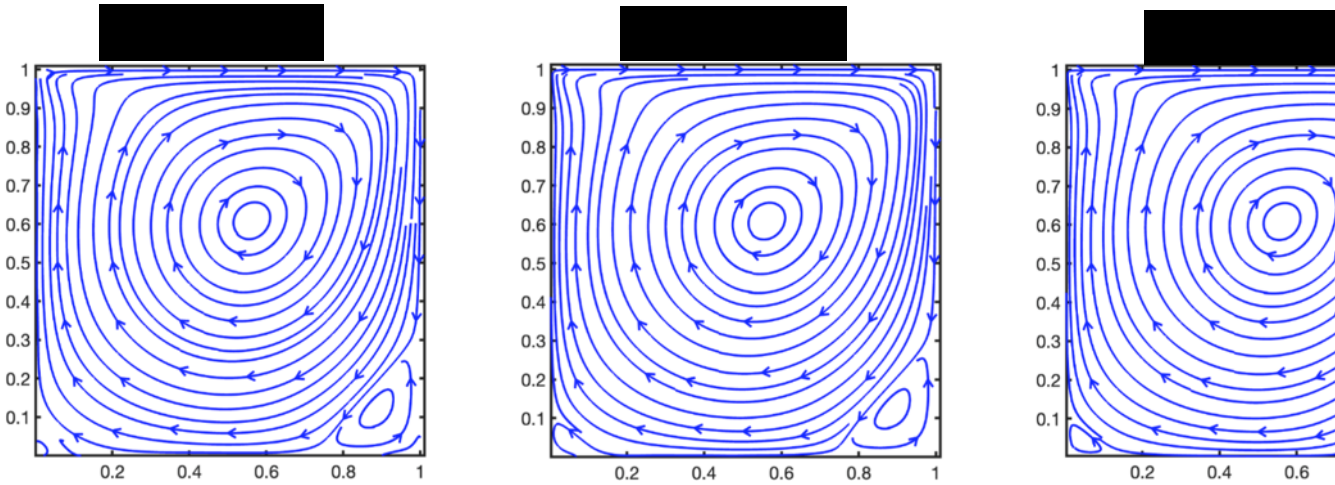
E reformulation:

$$\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \left(\frac{1}{Re} + \nu_E\right)\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right),$$

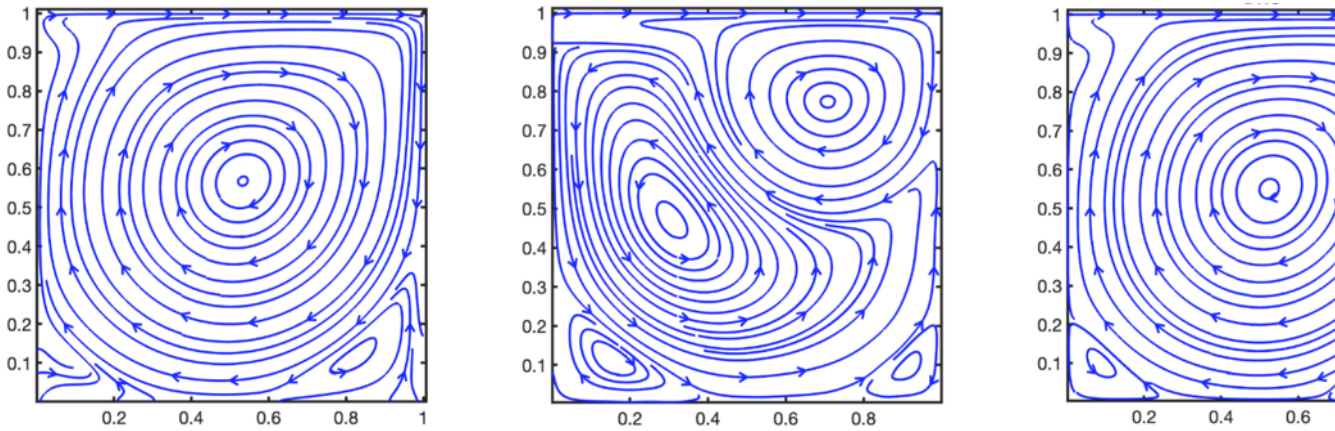
$$\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \left(\frac{1}{Re} + \nu_E\right)\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right),$$

$$e_3 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y},$$

Re=400



Re=2000



2D Helmholtz equation

DE:

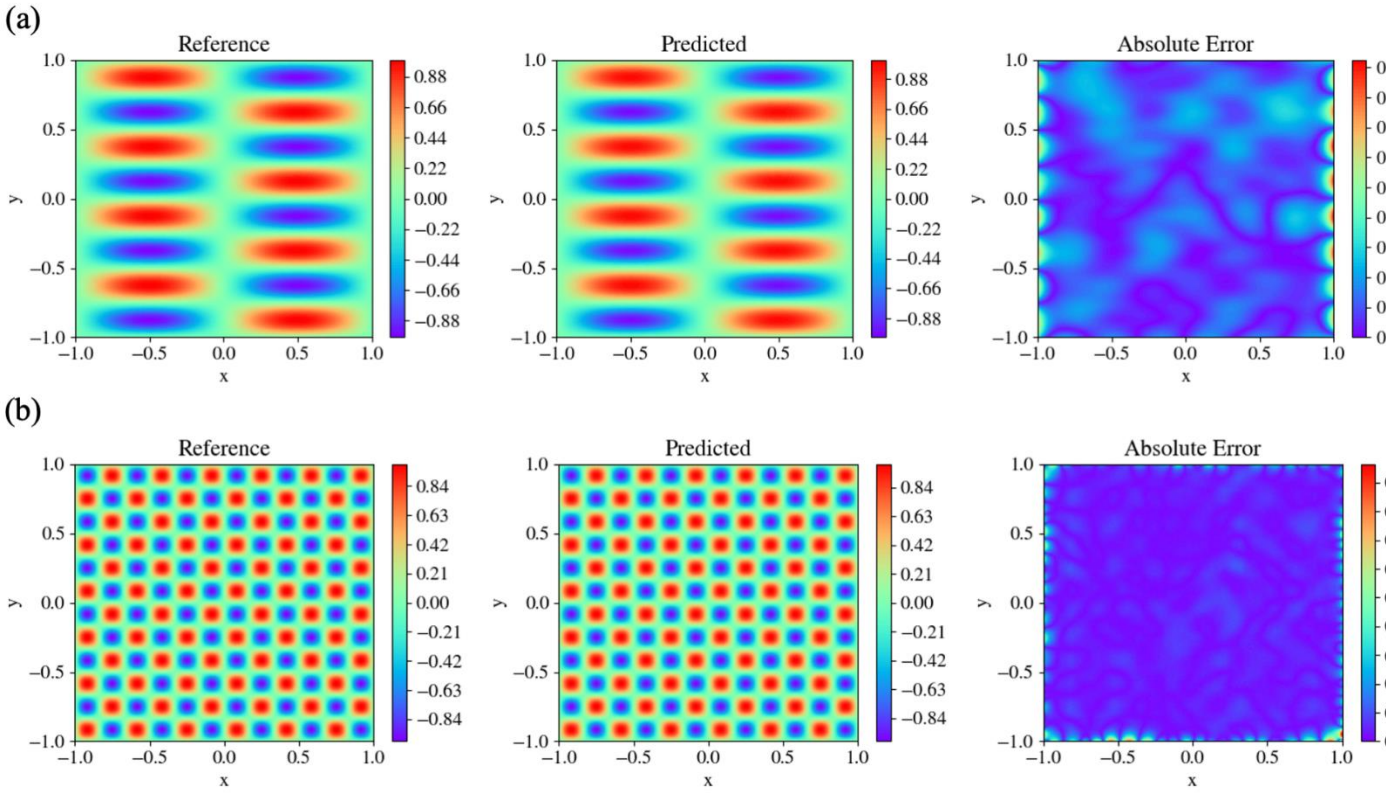
$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + k^2 u - q(x, y) = 0$$

Cs:

$$-1, y) = u(1, y) = u(x, -1) = u(x, -1) = 0$$

ses:

(a) LBFGs
(b) ADAM



Analytical solution for the 2D Helmholtz and corresponding network prediction with the absolute error difference

2D Helmholtz equation (Comparison)

Parameter-based Comparison $(a_1 = 1, a_2 = 4)$

Computational time-based Comparison $(a_1 = 1, a_2 = 4)$

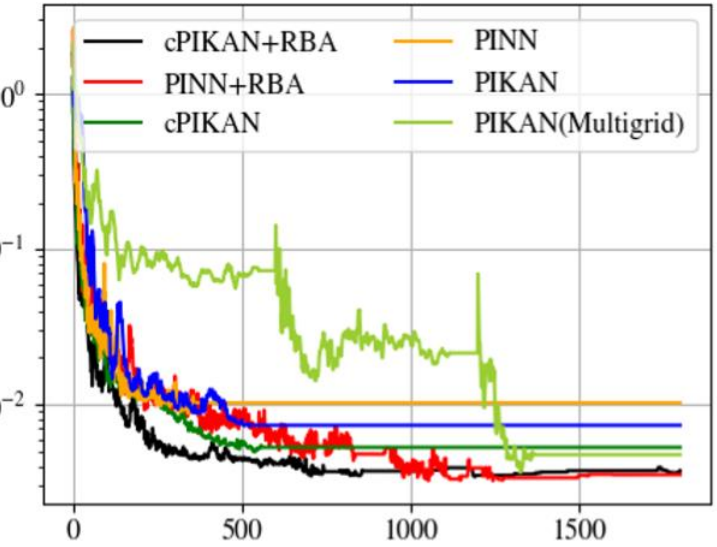
Complexity time-based Comparison $(a_1 = a_2 = 6)$

	Method	N. Params	Optimizer	Iterations	Relative L^2	Time($\frac{ms}{it}$)
(a)	PINN	304	LBFGS	1.8e3	1.03e-2	64
	PIKAN	300	LBFGS	1.8e3	7.35e-3	4550
	PIKAN(Multigrid)	240-690	LBFGS	1.8e3	4.76e-3	3243
	cPIKAN	350	LBFGS	1.8e3	5.301e-3	183
	PINN+RBA	304	LBFGS	1.8e3	3.54e-3	108
	cPIKAN+RBA	350	LBFGS	1.8e3	3.76e-3	243
(b)	PINN	30300	Adam	2.0e5	5.30e-3	5.1
	cPIKAN	15840	Adam	2.0e5	5.00e-3	6.7
	PINN+RBA	30300	Adam	2.0e5	2.06e-3	7.1
	cPIKAN+RBA	15840	Adam	2.0e5	1.60e-3	7.4
(c)	PINN	82304	Adam	5.0e5	4.30e-2	10.3
	cPIKAN	20960	Adam	5.0e5	N/A	8.0
	PINN+RBA	82304	Adam	5.0e5	1.72e-2	11.4
	cPIKAN+RBA	20960	Adam	5.0e5	3.81e-3	8.4

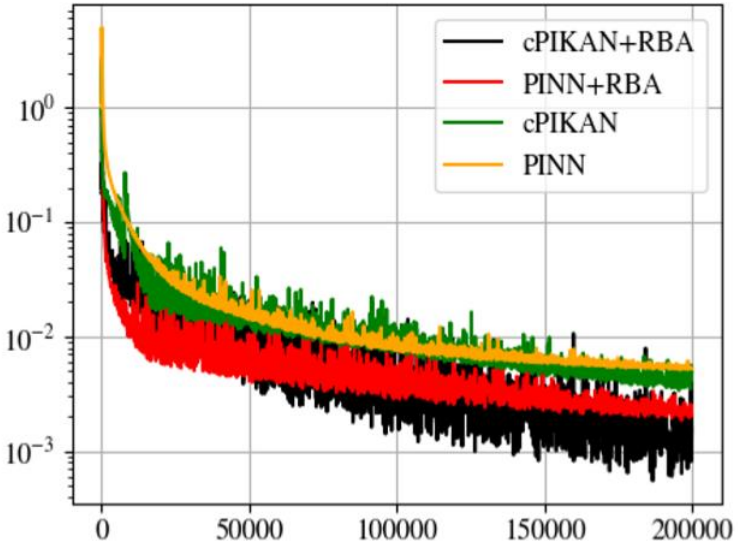
Table 2: Relative L^2 and computational time (ms/it) comparison between different models.(a) Parameter-based analysis for solving Helmholtz Equation $(a_1 = 1, a_2 = 4)$ using LBFGS optimizer and a biased loss function that downscale the PDE contribution. (b) Computation time-based comparison for solving Helmholtz Equation $(a_1 = 1, a_2 = 4)$ using ADAM optimizer with an unbiased loss function. (c) Complexity-based analysis using ADAM optimizer with no global weights for solving the Helmholtz equation with a higher wave number $(a_1 = a_2 = 6)$. For the cPIKAN model, N/A represents "not applicable" since loss became undefined after the initial iterations.

2D Helmholtz equation (Comparison)

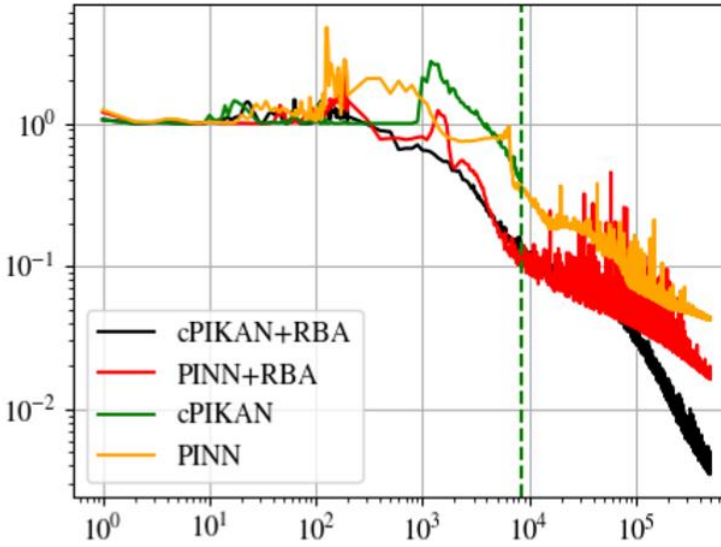
Parameter-based Comparison
$(a_1 = 1, a_2 = 4)$



Time-based Comparison
$(a_1 = 1, a_2 = 4)$



Complexity-based Comparison
$(a_1 = 6, a_2 = 6)$

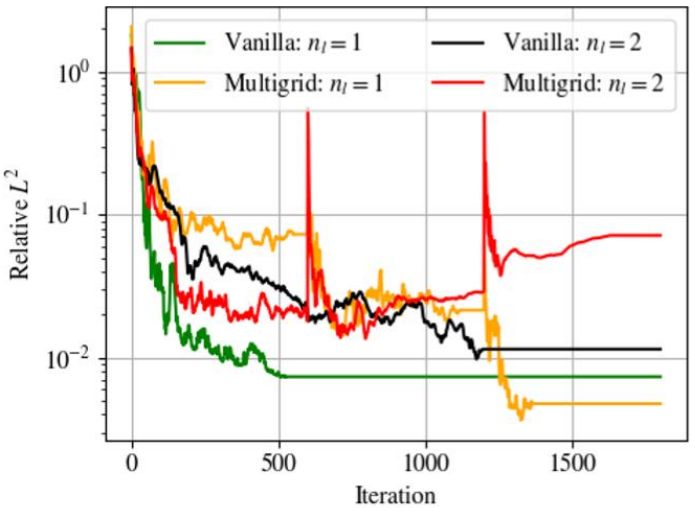


Sensitivity Analysis

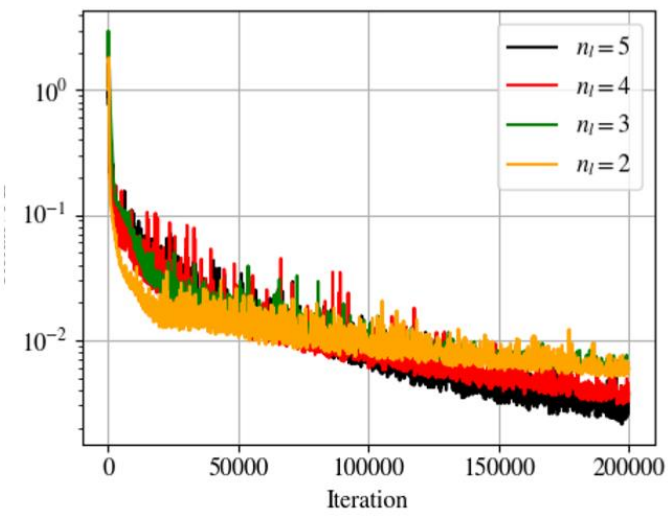
creasing the number of layers hinders PIKAN
erformance

creasing the number of layers or the
olynomial order improves cPIKAN performance

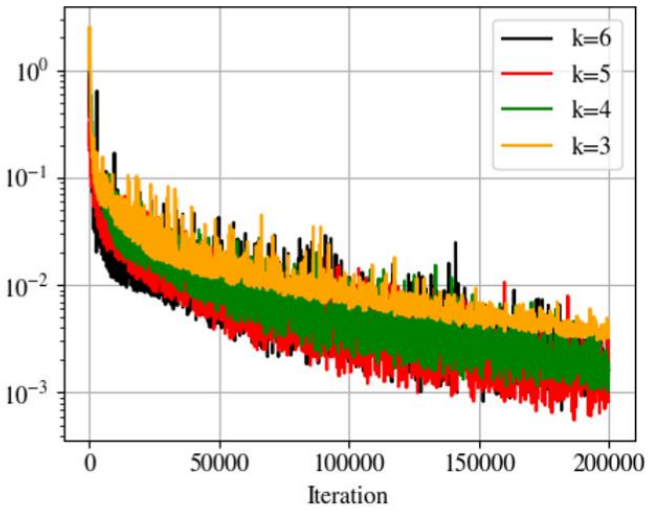
Helmholtz
$(a_1 = 1, a_2 = 4)$



Helmholtz
$(a_1 = 1, a_2 = 4)$



Helmholtz
$(a_1 = 1, a_2 = 4)$



Summary

re are many extensions of PINNs that can enhance accuracy and reduce training time.

e gradient enhanced PINN (gPINN) increases accuracy up to two orders of magnitude and requires only a small of residual points but it involves expensive extra automatic differentiation.

conservative PINN (cPINN) is based on domain decomposition and mimics the discontinuous Galerkin method. It can great parallelization in space.

extended PINN (XPINN) is also based on domain decomposition, but it is not limited to conservation laws and can lea parallelization in space and time. It can be applied to any partial differential equation, and it ensures that residual and state are continuous at the interfaces.

stochastic PINN (sPINN) can be used to solve stochastic PDEs. It is based on the Wasserstein GAN combined with a dient penalty term for stabilization. It scales to high dimensions and the cost is approximately quadratic h the dimension.

ANs present an alternative to PINNs that seems to be comparable in performance but more research is required.

ferences

- Blum AL, Rivest RL. Training a 3-node neural network is NP-complete. Neural Networks. 1992 Jan 1;5(1):117-27.
- Hu Z, Jagtap AD, Karniadakis GE, Kawaguchi K. When Do Extended Physics-Informed Neural Networks (XPINNs) Improve Generalization?. arXiv preprint arXiv:2109.09444. 2021 Sep 20.
- Jagtap AD, Karniadakis GE. Extended physics-informed neural networks (xPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. Communications in Computational Physics. 2020 Nov 1;28(5):2002-41.
- L. Lu, Y. Su, and G. E. Karniadakis. Collapse of Deep and Narrow Neural Nets. arXiv preprint arXiv:1808.04947. 2018 Aug 15.
- Lee JD, Simchowitz M, Jordan MI, Recht B. Gradient descent only converges to minimizers. InConference on learning theory 2016 Jun 6 (pp. 1246-1257). PMLR.
- Jagtap AD, Kharazmi E, Karniadakis GE. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering. 2020 Jun 15;365:113028.
- Pang G, Lu L, Karniadakis GE. fPINNs: Fractional physics-informed neural networks. SIAM Journal on Scientific Computing. 2019;41(4):A2603-26.
- Yang L, Zhang D, Karniadakis GE. Physics-informed generative adversarial networks for stochastic differential equations. SIAM Journal on Scientific Computing. 2020;42(1):A292-317.
- Yu J, Lu L, Meng X, Karniadakis GE. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. Computer Methods in Applied Mechanics and Engineering. 2022 Apr 1;393:114823.

ferences

ai G, Koley U, Mishra S, Molinaro R. On generalization error estimates of physics informed neural networks for approximating dispersive PDEs. SAM Research Report. 2021 3;2021.

e Ryck T, Lanthaler S, Mishra S. On the approximation of functions by tanh neural networks. arXiv preprint arXiv:2104.08938. 2021 Apr 18.

e Ryck T, Mishra S. Error analysis for physics informed neural networks (PINNs) approximating Kolmogorov PDEs. arXiv preprint arXiv:2106.14473. 2021 Jun 28.

opakumar V, Pamela S, Samaddar D. Loss landscape engineering via data regulation on PINNs. Machine Learning with Applications. 2023 Jun 15;12:100464.

endriks J, Jidling C, Wills A, Schön T. Linearly constrained neural networks. arXiv preprint arXiv:2002.01600. 2020 Feb 5.

agtap AD, Kharazmi E, Karniadakis GE. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse proble omputer Methods in Applied Mechanics and Engineering. 2020 Jun 15;365:113028.

in X, Cai S, Li H, Karniadakis GE. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. Journal of Computat hysics. 2021 Feb 1;426:109951.

harazmi E, Zhang Z, Karniadakis GE. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Computer Methods in Applied Mechanics and ngineering. 2021 Feb 1;374:113547.

harazmi E, Zhang Z, Karniadakis GE. Variational physics-informed neural networks for solving partial differential equations. arXiv preprint arXiv:1912.00873. 2019 Nov 27.

agaris PL, Tsoukalas LH, Safarkhani S, and Lagaris IE, Systematic construction ofneural forms for solving partial differential equations inside rectangular domains, subject to oundary and interface conditions, Int. J. Artif. Intell., 29 (2020), 2050009.

u Y, Chen H, Lu J, Ying L, Blanchet J. Machine Learning For Elliptic PDEs: Fast Rate Generalization Bound, Neural Scaling Law and Minimax Optimality. arXiv preprint rXiv:2110.06897. 2021 Oct 13.

u L, Pestourie R, Yao W, Wang Z, Verdugo F, Johnson SG. Physics-informed neural networks with hard constraints for inverse design. SIAM Journal on Scientific Computing 3(6), B1105–B1132, 2021.

cClenny L, Braga-Neto U. Self-adaptive physics-informed neural networks using a soft attention mechanism. arXiv preprint arXiv:2009.04544. 2020 Sep 7.

ishra S, Molinaro R. Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs. arXiv preprint arXiv:2006.16144. 2020 Jun 2

ishra S, Molinaro R. Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs. IMA Journal of Numer nalysis. 2021 Jun 17.

ishra S, Molinaro R. Physics informed neural networks for simulating radiative transfer. Journal of Quantitative Spectroscopy and Radiative Transfer. 2021 Aug 1;270:10770

aissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial ifferential equations. Journal of Computational Physics. 2019 Feb 1;378:686-707.

hin Y, Darbon J, Karniadakis GE. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. arXiv preprint rXiv:2004.01806. 2020 Apr 3.

hin Y, Zhang Z, Karniadakis GE. Error estimates of residual minimization using neural networks for linear PDEs. arXiv preprint arXiv:2010.08019. 2020 Oct 15.

Wang S, Yu X, Perdikaris P. When and why PINNs fail to train: A neural tangent kernel perspective. Journal of Computational Physics. 2022 Jan 15;449:110768.

Deep Learning for Science and Engineering Teaching Kit

Thank You