COMPETITIVE PHYSICS INFORMED NETWORKS

ICLR2023

Various real-world applications are usually described by partial differential equations (PDEs).

- Navier-Stokes Eq.
- Schrödinger Eq.
- Maxwell Eq.

- numerical algorithms
  - computationally expensive
  - face severe challenges in multi-physics and multi-scale systems
  - data assimilation introduces additional uncertainties
- machine learning
  - Physics-informed Neural Networks (PINNs): encode PDEs into the loss function of neural networks

$$\begin{aligned} \mathcal{A}[u] = f, \quad & \text{in}\Omega \\ u = g, \quad & \text{on}\partial\Omega, \end{aligned}$$

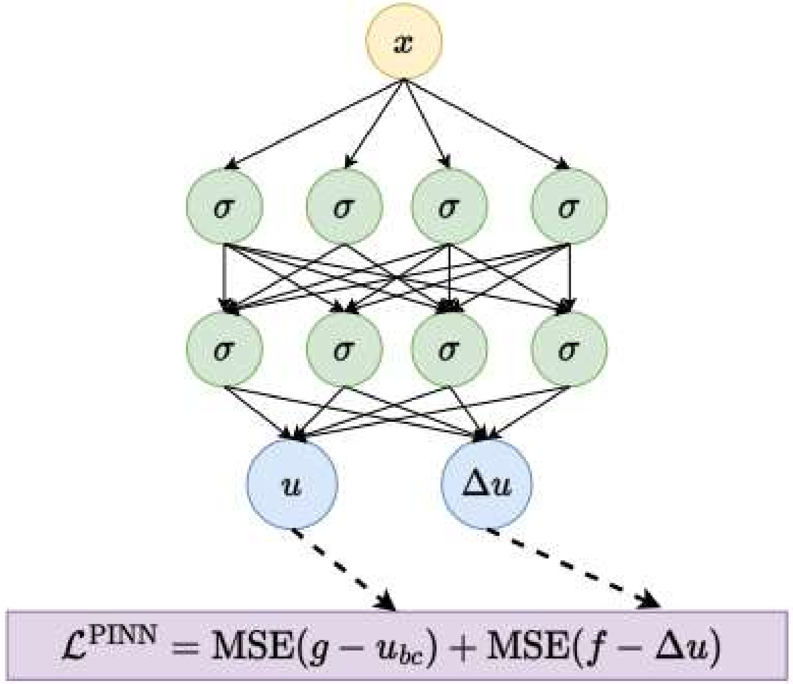
use neural network to approximate u

$$\mathcal{L}^{\mathrm{PINN}}(\mathcal{P}, \boldsymbol{x}, \overline{\boldsymbol{x}}) = \mathcal{L}_{\Omega}^{\mathrm{PINN}}(\mathcal{P}, \boldsymbol{x}_{\Omega}) + \mathcal{L}_{\partial\Omega}^{\mathrm{PINN}}(\mathcal{P}, \overline{\boldsymbol{x}}),$$

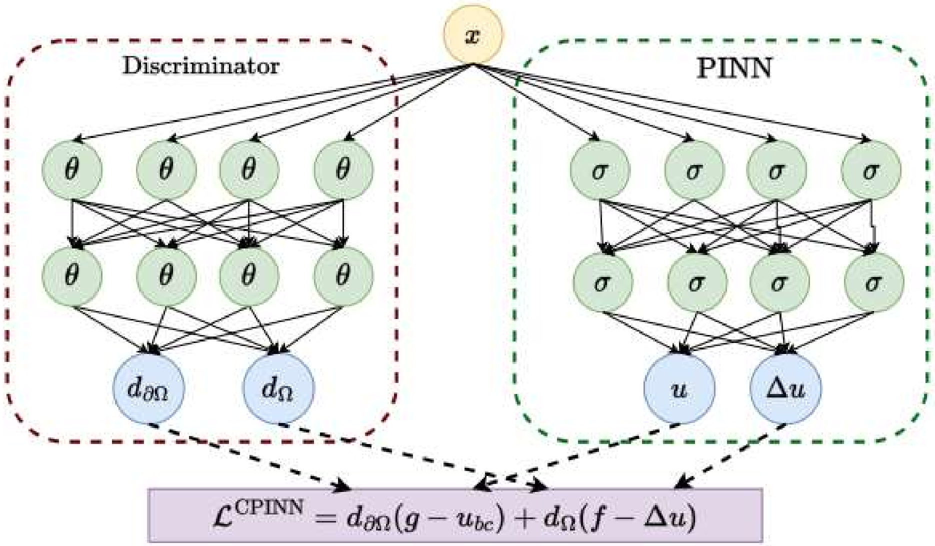
$$\mathcal{L}_{\Omega}^{\mathrm{PINN}}(\mathcal{P}, \boldsymbol{x}) = \frac{1}{N_{\Omega}} \sum_{i=1}^{N_{\Omega}} \left(\mathcal{A}[\mathcal{P}](x_i) - f(x_i)\right)^2$$

$$\mathcal{L}_{\partial\Omega}^{\mathrm{PINN}}(\mathcal{P}, \overline{\boldsymbol{x}}) = \frac{1}{N_{\partial\Omega}} \sum_{i=1}^{N_{\partial\Omega}} \left(\mathcal{P}\left(\overline{x}_i\right) - g\left(\overline{x}_i\right)\right)^2$$

Physics-Informed Neural Networks



(a) PINN



(b) CPINN

$$\max_{\mathcal{D}} \min_{\mathcal{P}} \mathcal{L}_\Omega^{\mathbf{CPINN}}(\mathcal{P}, \mathcal{D}, \boldsymbol{x}) + \mathcal{L}_{\partial\Omega}^{\mathbf{CPINN}}(\mathcal{P}, \mathcal{D}, \overline{\boldsymbol{x}}),$$

$$\mathcal{L}_\Omega^{\text{CPINN}}(\mathcal{D}, \mathcal{P}, \boldsymbol{x}) = \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \mathcal{D}_\Omega(x_i)\left(\mathcal{A}[\mathcal{P}](x_i) - f(x_i)\right) \qquad \mathcal{L}_{\partial\Omega}^{\text{CPINN}}(\mathcal{D}, \mathcal{P}, \overline{\boldsymbol{x}}) = \frac{1}{N_{\partial\Omega}} \sum_{i=1}^{N_{\partial\Omega}} \mathcal{D}_{\partial\Omega}(\overline{x}_i)\left(\mathcal{P}(\overline{x}_i) - g(\overline{x}_i)\right)$$

The Nash equilibrium of this game is: $\mathcal{P} \equiv u$ and $\mathcal{D} \equiv 0$

$$\mathcal{P}(x) = \sum_{i=1}^{\dim(\boldsymbol{\pi})} \pi_i \psi_i(x), \quad \mathcal{D}(x) = \sum_{i=1}^{\dim(\boldsymbol{\delta})} \delta_i \phi_i(x),$$

$$A \in \mathbb{R}^{N_\Omega \times \dim(\boldsymbol{\pi})}$$

$$f \in \mathbb{R}^{N_\Omega}$$

$$\Longrightarrow \quad A_{ij} := \mathcal{A}[\psi_j](x_i), \quad f_i := f(x_i)$$

$$A\pi = f$$

For PINNs: $$\min_{\boldsymbol{\pi}} \|A\pi - f\|^2$$

$$\boldsymbol{\pi} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{f} \qquad \kappa(\boldsymbol{A}^\top \boldsymbol{A}) = \kappa(\boldsymbol{A})^2$$

For CPINNs: $$\min_{\boldsymbol{\pi}} \max_{\boldsymbol{\delta}} \boldsymbol{\delta}^\top (\boldsymbol{A\pi} - \boldsymbol{f})$$

$$\begin{bmatrix} 0 & \boldsymbol{A}^\top \\ \boldsymbol{A} & 0 \end{bmatrix} \begin{bmatrix} \pi \\ \delta \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix}, \quad \text{with} \quad \kappa\left(\begin{bmatrix} 0 & \boldsymbol{A}^\top \\ \boldsymbol{A} & 0 \end{bmatrix}\right) = \kappa(\boldsymbol{A})$$

POISSON EQUATION, NONLINEAR SCHRÖDINGER EQUATION,
BURGERS' EQUATION, ALLEN–CAHN EQUATION

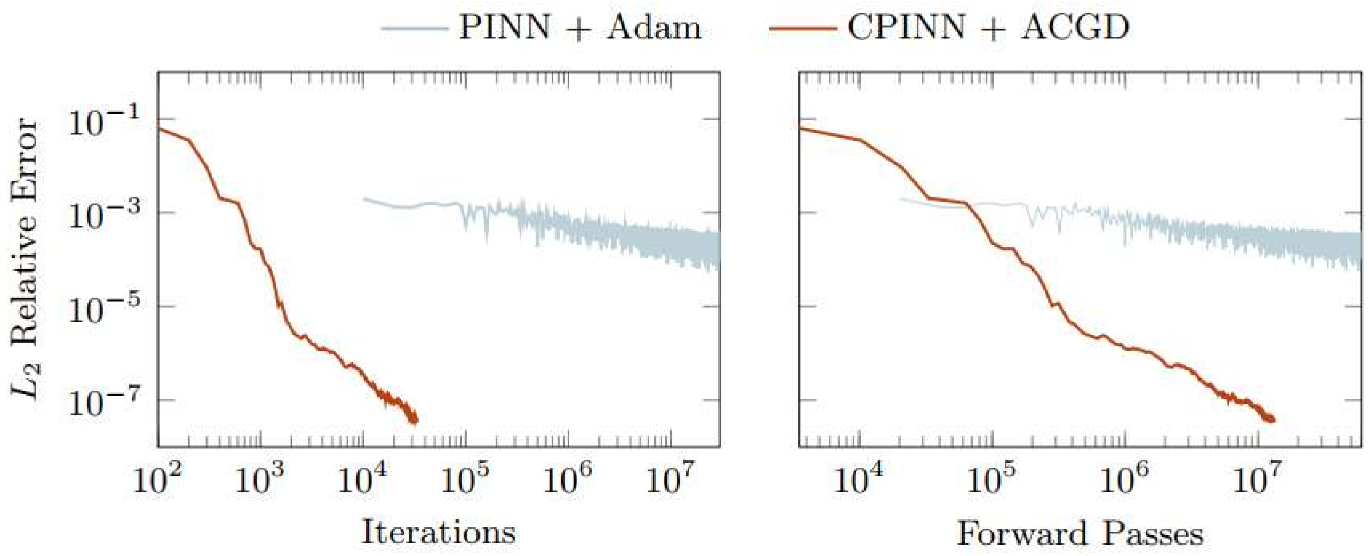


Figure 1: Comparison of CPINN and PINN on the Poisson problem of equation 15 in terms of relative error. CPINN has a faster convergence rate and reduces the L_2 error to 1.7×10^{-8}, whereas the PINN case has an L_2 error of 1.2×10^{-4} even with a larger computational budget.

RESULTS

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

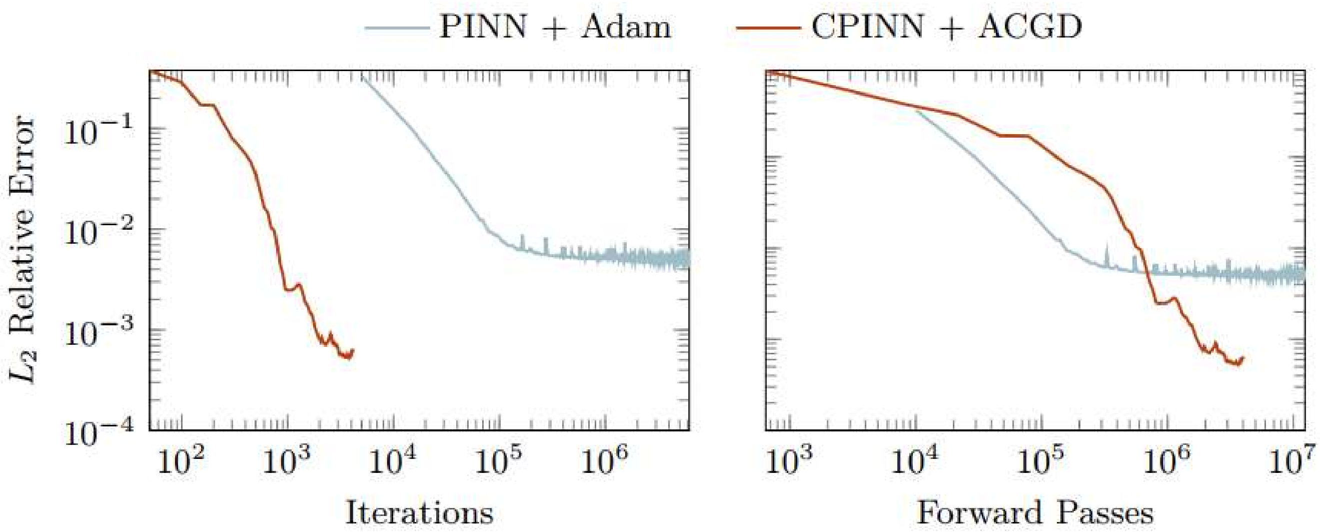


Figure 3: Comparison of CPINN and PINN on the nonlinear Schrödinger eq. (21) in terms of relative errors. After 200 000 training iterations, PINN cannot reduce the L_2 error further, plateauing about 4×10^{-3} , whereas CPINN reduces the error to 6×10^{-4} under a smaller computational budget.

Table 1: Performance of CPINNs and PINNs on the 2D Poisson problem of equation 15.

	Optimizer	Iterations	L_2 Rel. Error	$\mathcal{L}^{\text{PINN}}$	$\mathcal{L}_{\Omega}^{\text{PINN}}$	$\mathcal{L}_{\partial\Omega}^{\text{PINN}}$
PINN	Adam	3×10^{7}	1.2×10^{-4}	1.4×10^{-8}	8.4×10^{-8}	5.5×10^{-9}
	SGD	2.2×10^{7}	1.3×10^{-3}	1.1×10^{-6}	5.2×10^{-7}	6.3×10^{-7}
CPINN	ACGD	4.8×10^{4}	1.7×10^{-8}	2.1×10^{-14}	2×10^{-14}	6×10^{-16}
	CGD	6×10^{5}	3.5×10^{-4}	5.1×10^{-7}	3.5×10^{-7}	1.7×10^{-7}
	XAdam	8.5×10^{6}	1.6	1.6×10^{6}	1.6×10^{6}	0.2
	XSGD	8.5×10^{6}	1.2×10^{-4}	1.2×10^{-7}	9.2×10^{-8}	3×10^{-8}
WAN	Adam + Adagrad	1.3×10^{6}	1.2	2.6	2.6	1.2×10^{-3}
	ACGD	1.8×10^{5}	9×10^{-3}	1.1×10^{-3}	1.1×10^{-3}	2.7×10^{-5}