Stability in Training PINNs for Stiff PDEs: Why Initial Conditions Matter

Baoli Hao, Ulisses Braga-Neto, Chun Liu, Lifan Wang, Ming Zhong

Illinois Institute of Technology

August 15, 2026

Scientific Machine Learning

Physics Informed Machine Learning

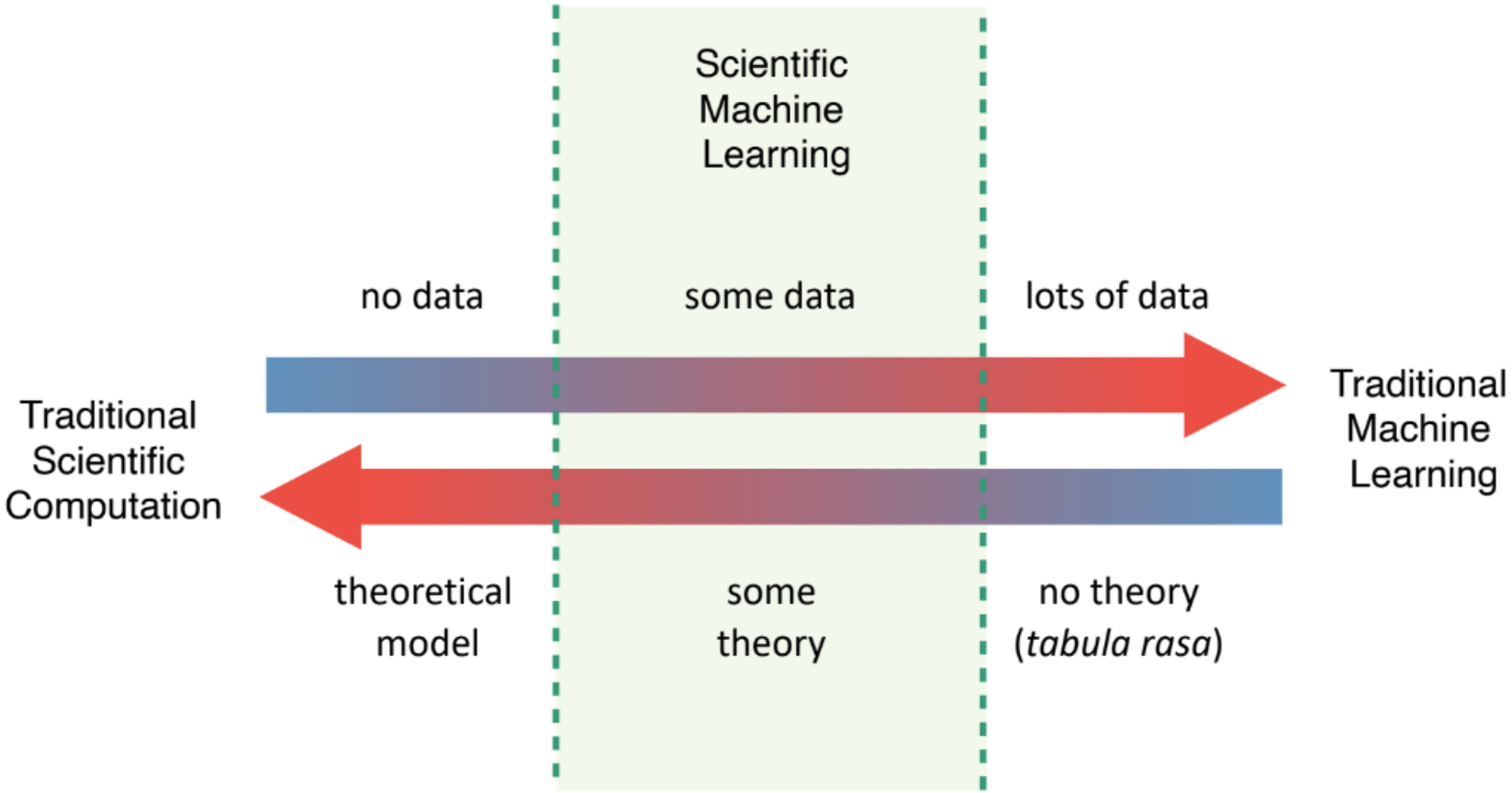

Figure: New Era of Machine Learning: Balance between Data and Theory.

Physics-Informed Neural Network[3]

- PINNs integrate PDE constraints into deep learning

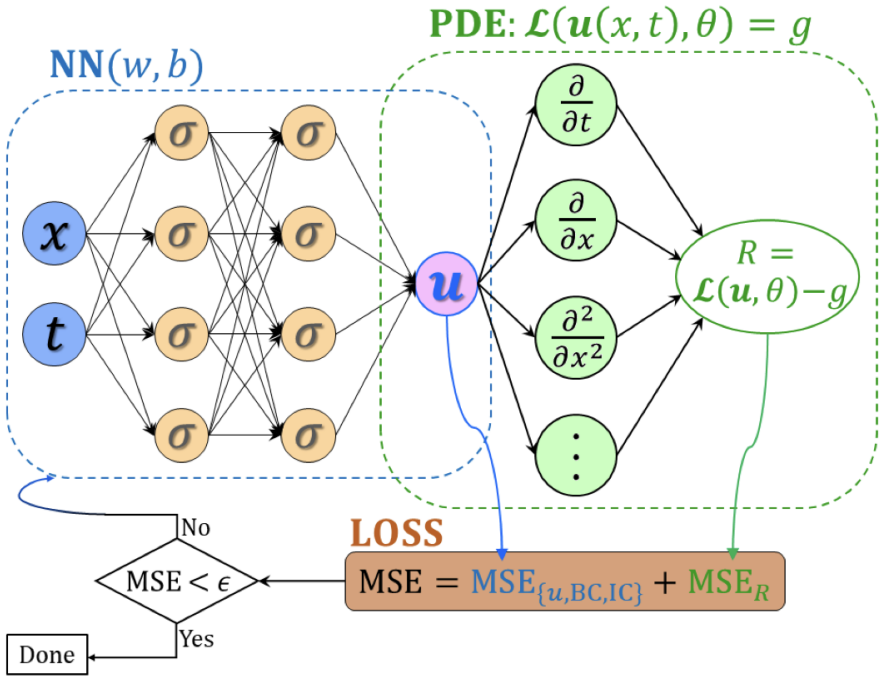

Figure: Schematic of PINN, where the loss function of PINN contains a mismatch in the given data on the state variables or boundary and initial conditions. The hyperparameters of PINN can be learned by minimizing the total loss $\mathcal{L} = MSE_{IC} + MSE_{BC} + MSE_R$.

Motivation

- Time-dependent PDEs are essential for modeling physics and engineering phenomena.
- Training PINNs on stiff time-dependent PDEs is often unstable.
- Key question: Why do PINNs often fail to train stably on stiff PDEs?
- Key challenge: propagating information from initial conditions (ICs).

Main Contributions

- Identify IC enforcement as critical for PINN stability.
- Provide theoretical analysis (well-conditioning + NTK perspective).
- Ablation Studies on 7 stiff PDEs, 2D Allen–Cahn, sensitivity studies.

[2] **Hao, B., Braga-Neto, U., Liu, C., Wang, L., & Zhong, M. (2024). Stability in Training PINNs for Stiff PDEs: Why Initial Conditions Matter. arXiv preprint arXiv:2404.16189.**

Methodology

- PDE setup: $u_t + \mathcal{P}(u) = f$ on $(0, T] \times \Omega$.
- **Hard constraint transformation:**

$$\tilde{u}(t, \boldsymbol{x}) = \psi(t, \boldsymbol{x}) + \phi(t, \boldsymbol{x}) u_{\mathrm{NN}}(t, \boldsymbol{x}),$$

  with $\psi(0, \boldsymbol{x}) = u_0(\boldsymbol{x}), \phi(0, \boldsymbol{x}) = 0$ ensuring exact IC satisfaction.
- Training reduces to minimizing PDE residual only.
- Standard PINN loss combines:

$$\mathcal{L} = \mathcal{L}_{\mathrm{IC/BC}} + \mathcal{L}_{\mathrm{R}}.$$

- HC-PINN loss:

$$\mathcal{L} = \mathcal{L}_{\mathrm{R}}.$$

Well-Conditioning of the Transformation

Key Idea

If ψ, φ are smooth and $\varphi_t(0, \boldsymbol{x}) \neq 0$, the transformation is well-posed.

- Hard-constraint transformation requires mild regularity:
  - ψ, ϕ are smooth (C^1 in t and K-order in x).
  - $\phi_t(0, x) \neq 0$ ensures no blow-up at $t = 0$.
- Then u_{nn} still satisfies a well-posed PDE.
- The transformation introduces a stabilizing "reaction term" and updated forcing. Initial conditions are built in, ensuring stable PINN training.
- i.e., $\phi(t, \boldsymbol{x}) = 1 - e^{-Ct}$ with $\psi(t, \boldsymbol{x}) = e^{-Ct} u_0(\boldsymbol{x})$.

Why Initial Conditions Matter?

- Stiff PDEs evolve on **widely varying time scales**.
- Explicit solvers struggle $\rightarrow$ stability hinges on **initial conditions**.

Summary

- Hard ICs enforce the time direction in training, reducing spectral bias.
- Mimics traditional numerical solvers (exact ICs at $t = 0$).
- Behaves like an implicit time integrator $\rightarrow$ more stable.

How Hard Constraints Work – NTK View

1. Under NTK linearization:

$$\dot{\theta} = -\nabla_\theta \mathcal{L}, \quad \dot{u}(\boldsymbol{z}) \approx -K(\theta_0)\nabla_u \mathcal{L},$$

where K is the NTK matrix.

2. **Soft case:** Loss = IC + PDE residual. Residual dynamics:

$$\dot{r}_s = -K_*^s r_s, \quad K_*^s = \begin{bmatrix} K_{11}^s & K_{12}^s \\ K_{21}^s & K_{22}^s \end{bmatrix}.$$

3. Decay depends on the smallest eigenvalue of K_*^s, sensitive to imbalance between IC and PDE parts ("spectral bias").

4. **Hard case:** IC removed via transformation, residual = PDE only:

$$\dot{r}_h = -K_*^h r_h,$$

decay depends only on a single kernel K_*^h.

5. **Conclusion:** Hard constraints collapse multi-block NTK spectrum into one block $\Rightarrow$ faster & more stable decay, reduced spectral bias.

Results

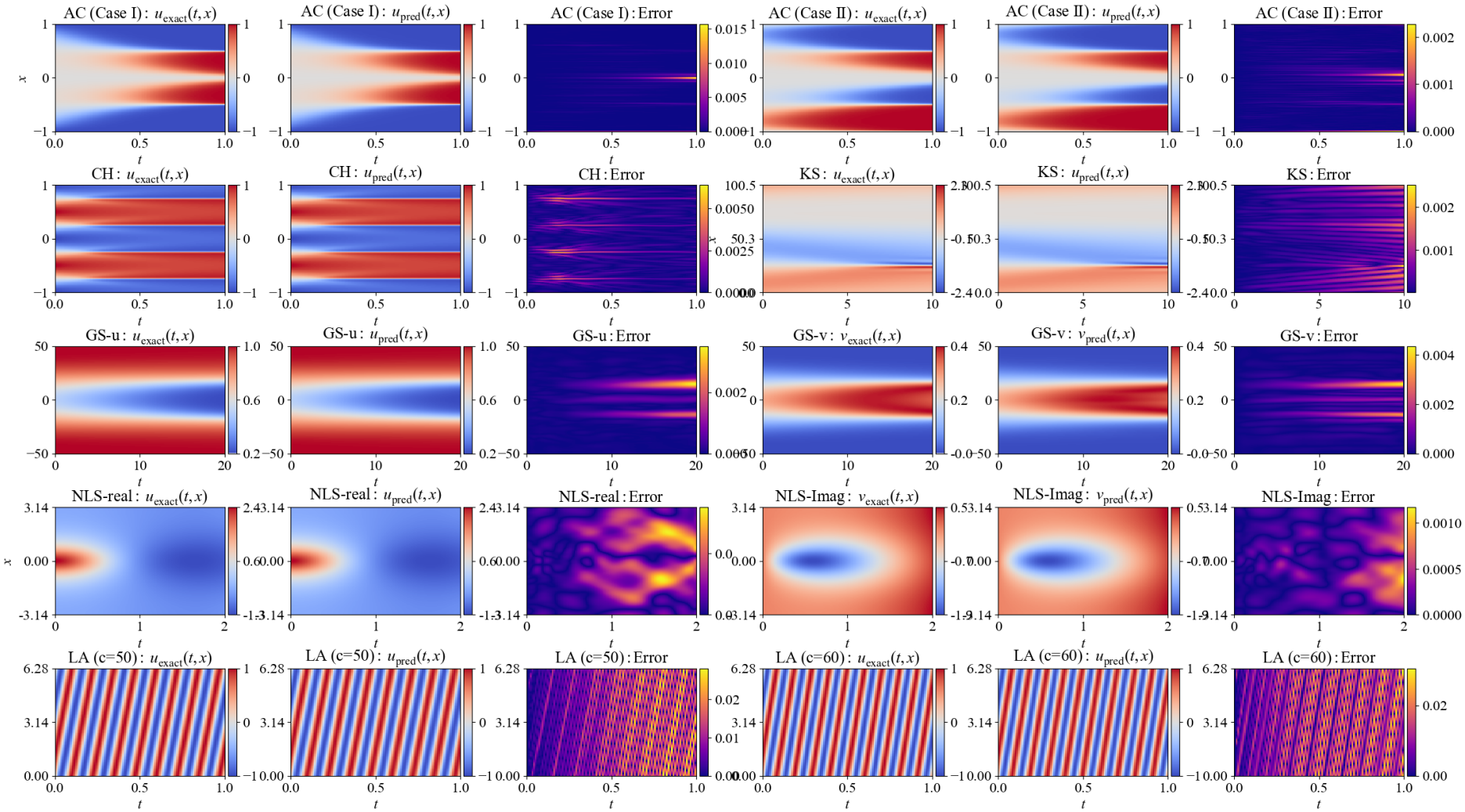

Figure: Benchmark Results: Truth vs HC-PINN vs Absolute Error.

Results: Comparison

	Causal training (MLP)[4]	Enhanced RBA[1]	HC-PINNs
AC (Case I)	6.95×10^{-2}	2.62×10^{-3}	8.29×10^{-4}
AC (Case II)	1.78×10^{-2}	1.08×10^{-3}	2.14×10^{-4}
CH	3.49×10^{-1}	9.83×10^{-2}	5.61×10^{-4}
KS	2.72×10^{-1}	3.64×10^{-2}	5.37×10^{-4}

Table: Relative L^2 errors after $50k$ training iterations. All methods are implemented under the same $50k$ iteration.

- HC-PINNs achieve much lower errors with fewer iterations.

Results: 2D Allen-Cahn

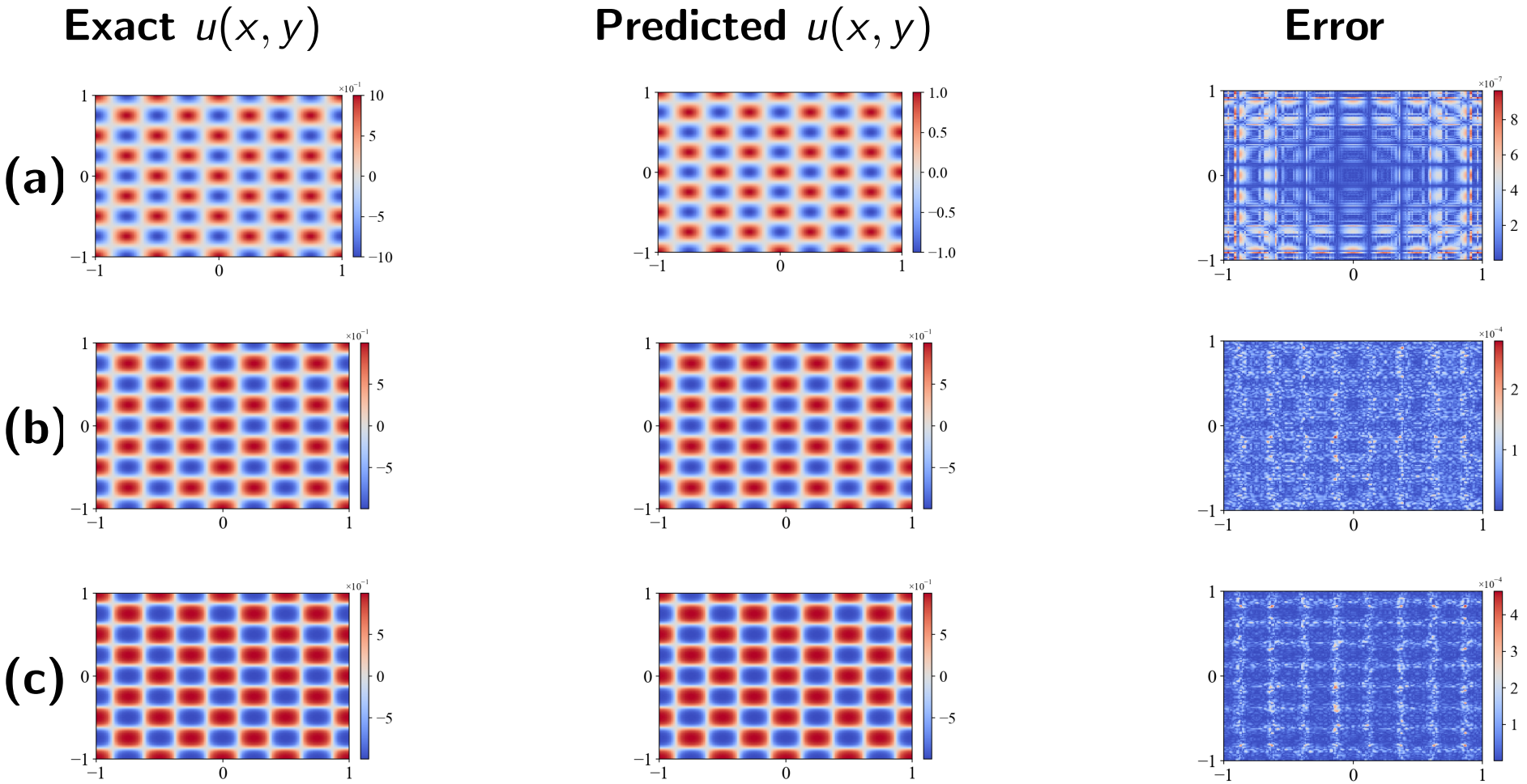

Figure: Reference, HC-PINN predicted solution, and absolute error of the 2D Allen Cahn equation at different time snapshots (a) $t = 0$, (b) $t = 0.5$, (c) $t = 1$.

Sensitivity Analysis

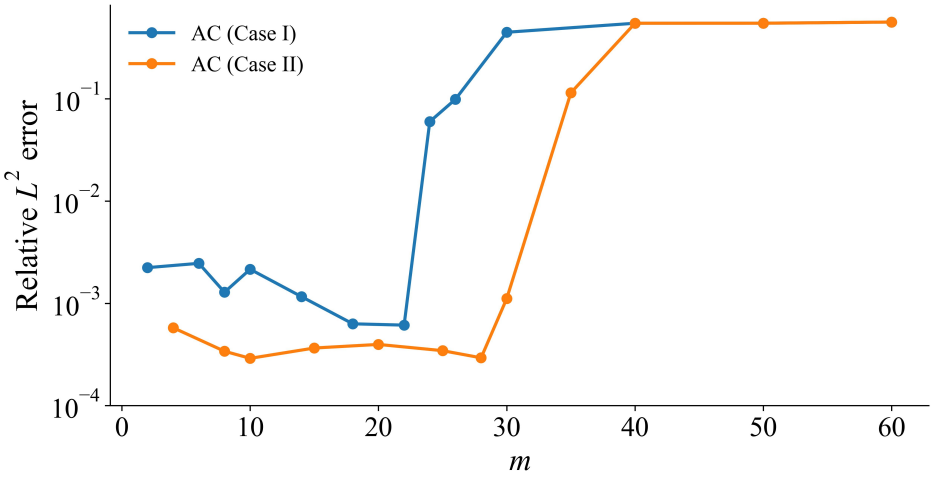

Figure: Sensitivity of HC-PINNs to the number of Fourier modes m in the Allen–Cahn equations. The plot shows the relative L^2 error as a function of m.

- Stable for small m, deteriorates when m exceeds threshold.
- For the decay rate C in ψ and ϕ , we compared $C \in \{1, 0.1\}$ and conducted ablation studies for each benchmark.

Conclusion

- Enforcing ICs via hard constraints is **essential** for stable PINN training on stiff PDEs.
- Ablation studies for HC-PINNs and comparison with other methods across diverse benchmarks.
- Sensitivity analysis on key hyperparameters.
- Future work: Higher dimensions, optimal transformations, efficiency improvements.

References I

[1] Sokratis J Anagnostopoulos, Juan Diego Toscano, Nikolaos Stergiopulos, and George Em Karniadakis. Residual-based attention and connection to information bottleneck theory in pinns. *arXiv preprint arXiv:2307.00379*, 2023.

[2] Baoli Hao, Ulisses Braga-Neto, Chun Liu, Lifan Wang, and Ming Zhong. Stability in training pinns for stiff pdes: Why initial conditions matter. *arXiv preprint arXiv:2404.16189*, 2024.

[3] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[4] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*, 2022.