

# Towards a Foundation Model for Physics-Informed Neural Networks: Multi-PDE Learning with Active Sampling

Keon Vin Park<sup>1,\*</sup>

<sup>1</sup>Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul, Republic of Korea  
 \*kbpark16@snu.ac.kr

## ABSTRACT

Physics-Informed Neural Networks (PINNs) have emerged as a powerful framework for solving partial differential equations (PDEs) by embedding physical laws into neural network training. However, traditional PINN models are typically designed for single PDEs, limiting their generalizability across different physical systems. In this work, we explore the potential of a **foundation PINN model** capable of solving multiple PDEs within a unified architecture. We investigate the efficacy of a single PINN framework trained on **four distinct PDEs**—the **Simple Harmonic Oscillator (SHO)**, the **1D Heat Equation**, the **1D Wave Equation**, and the **2D Laplace Equation**—demonstrating its ability to learn diverse physical dynamics.

To enhance sample efficiency, we incorporate **Active Learning (AL)** using **Monte Carlo (MC) Dropout-based uncertainty estimation**, selecting the most informative training samples iteratively. We evaluate different active learning strategies, comparing models trained on **10%, 20%, 30%, 40%, and 50% of the full dataset**, and analyze their impact on solution accuracy. Our results indicate that **targeted uncertainty sampling significantly improves performance with fewer training samples**, leading to efficient learning across multiple PDEs.

This work highlights the feasibility of a **generalizable PINN-based foundation model**, capable of adapting to different physics-based problems without redesigning network architectures. Our findings suggest that **multi-PDE PINNs with active learning** can serve as an effective approach for reducing computational costs while maintaining high accuracy in physics-based deep learning applications.

## Introduction

Physics-Informed Neural Networks (PINNs) have emerged as a powerful approach for solving Partial Differential Equations (PDEs) by incorporating physics-based constraints into deep learning models<sup>1</sup>. Unlike traditional numerical solvers such as finite element or finite difference methods, PINNs leverage neural networks to approximate solutions while enforcing governing physical laws through automatic differentiation. This enables PINNs to solve complex PDEs even in scenarios where sparse, noisy, or incomplete data is available<sup>2</sup>.

Despite their success, existing PINN models are typically designed to solve a single PDE, limiting their generalizability across diverse physical systems. This restriction presents a significant challenge in applying PINNs to real-world problems, where multiple PDEs often govern complex dynamics<sup>3</sup>. For instance, climate modeling requires solving both Navier-Stokes equations for fluid dynamics and radiative transfer equations for heat distribution<sup>4</sup>. Similarly, engineering applications such as fluid-structure interactions involve solving coupled PDEs from different domains<sup>5</sup>. Developing a generalizable PINN framework that can efficiently learn from multiple PDEs without requiring problem-specific architectures is an important step toward broader applicability.

Another major challenge in training PINNs is data efficiency. Since many physical systems require high-resolution simulations, the computational cost of generating labeled training data can be prohibitive<sup>6</sup>. In standard PINNs, training points are often sampled randomly or uniformly across the domain, which may lead to inefficient learning. Active learning has been widely studied as a strategy for selecting the most informative training samples, thereby improving model accuracy while reducing the number of required training points<sup>7</sup>. In this approach, new training samples are iteratively selected based on model uncertainty, allowing for adaptive data refinement. Recent studies have demonstrated the effectiveness of uncertainty-based sampling strategies, such as Monte Carlo (MC) Dropout, in improving PINN performance<sup>8</sup>.

In this study, we propose a unified PINN framework capable of solving multiple PDEs within a single neural network model. We incorporate active learning techniques to improve sample efficiency and systematically evaluate their effectiveness on different PDEs. The main contributions of our work are as follows:

- Development of a PINN architecture that generalizes across multiple PDEs, eliminating the need for problem-specific model designs.
- Integration of active learning strategies into the training process, selecting data points based on uncertainty estimation to enhance sample efficiency.
- Comparative analysis of different active learning strategies, evaluating their impact on solution accuracy and computational cost across a range of PDEs.
- Demonstration of the feasibility of a foundation model approach for physics-informed deep learning, paving the way for broader applications in scientific computing.

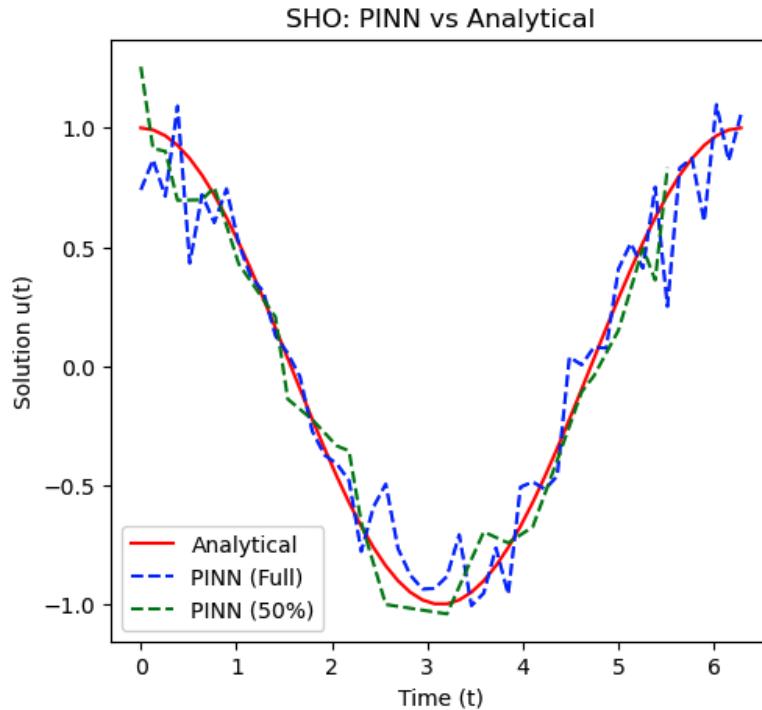
Our experimental results show that integrating active learning into PINNs significantly reduces the number of required training samples while maintaining high accuracy. By enabling a single PINN model to learn from multiple PDEs, this work provides a step toward generalizable physics-informed machine learning, bridging the gap between deep learning and traditional numerical methods.

## Results

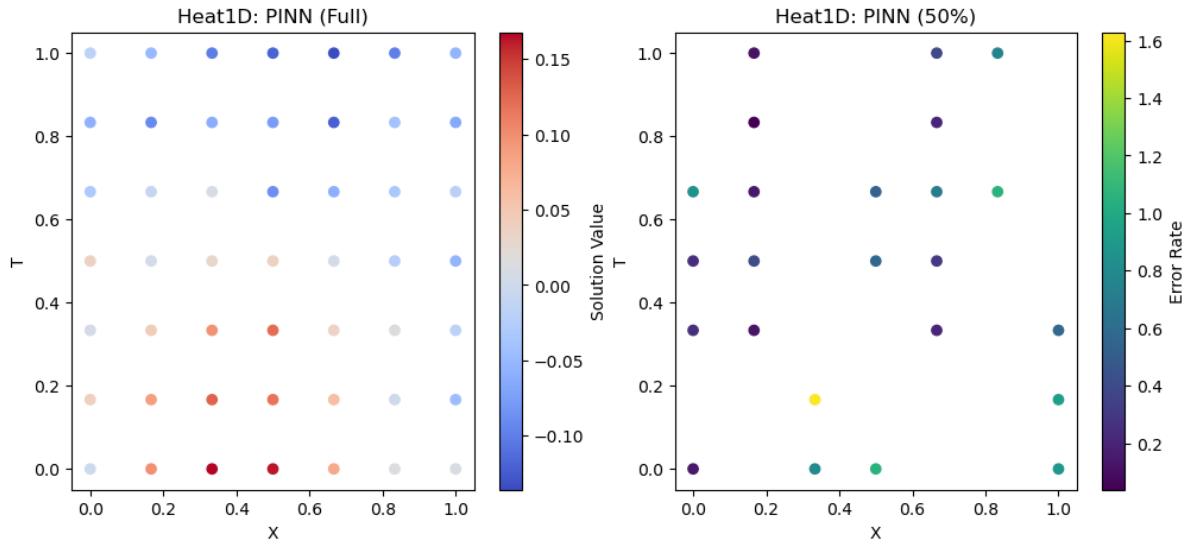
In this section, we present the results of our experiments on solving various Partial Differential Equations (PDEs) using Physics-Informed Neural Networks (PINNs). We evaluate the performance of PINNs under different active learning percentages, comparing the solutions obtained using the full dataset with those obtained using only 50% of the data.

### Comparison of PINN Solutions

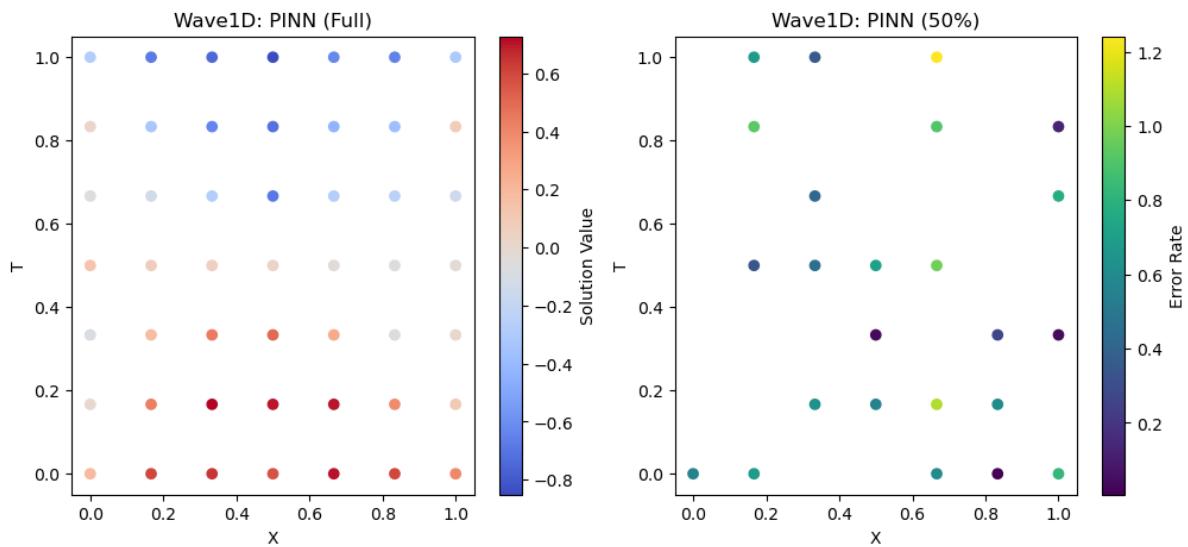
Figure 1 illustrates the results for the Simple Harmonic Oscillator (SHO). The PINN model trained on the full dataset and the model trained on 50% of the dataset are compared against the analytical solution. Figures 2, 3, and 4 show the results for the Heat equation, Wave equation, and Laplace equation, respectively. The left subfigure in each figure represents the solution obtained using the full dataset, while the right subfigure shows the error distribution for the 50% data model.



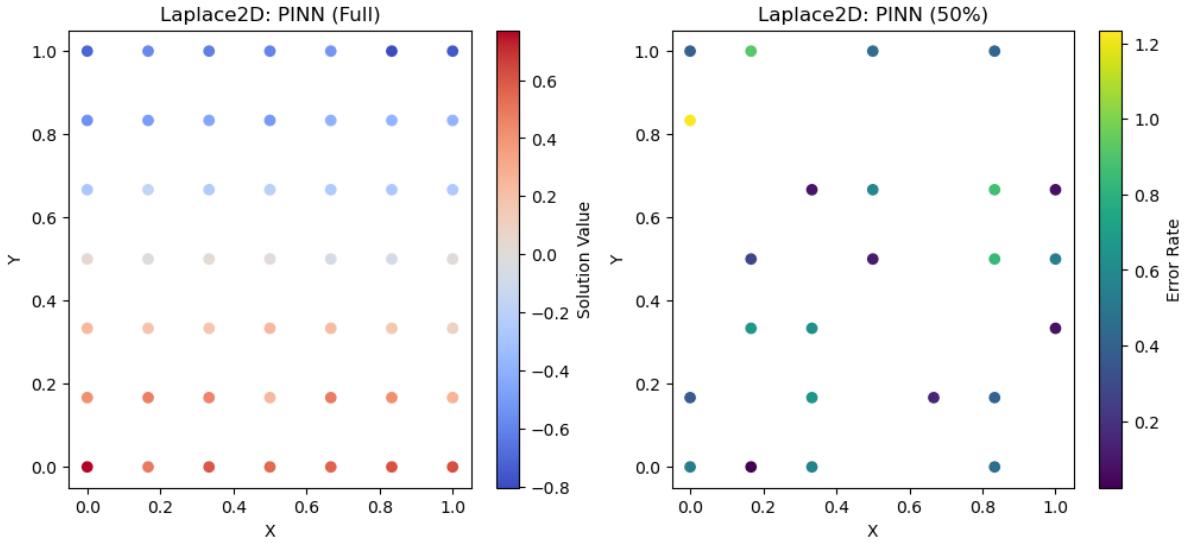
**Figure 1.** Comparison of PINN predictions with the analytical solution for the Simple Harmonic Oscillator (SHO).



**Figure 2.** Comparison of PINN solutions for the 1D Heat equation. Left: Full data solution. Right: Error distribution for the 50% data model.



**Figure 3.** Comparison of PINN solutions for the 1D Wave equation. Left: Full data solution. Right: Error distribution for the 50% data model.



**Figure 4.** Comparison of PINN solutions for the 2D Laplace equation. Left: Full data solution. Right: Error distribution for the 50% data model.

### Error Analysis

Table 1 presents the error rates for different active learning percentages across the studied PDEs. As observed, the accuracy of PINN predictions improves with an increasing amount of training data. However, even when trained on only 50% of the dataset, the PINN models still achieve reasonably low errors, highlighting the efficiency of active learning in selecting the most informative training samples.

**Table 1.** Error rates for different active learning percentages on various PDEs.

PDE	Active Learning (%)	Error
SHO	10	0.1176
	20	0.1308
	30	0.1242
	40	0.0927
	50	0.1097
Heat1D	10	0.3297
	20	0.3404
	30	0.3341
	40	0.3377
	50	0.3330
Wave1D	10	0.1261
	20	0.1174
	30	0.1170
	40	0.1170
	50	0.1390
Laplace2D	10	0.2470
	20	0.2430
	30	0.2472
	40	0.2361
	50	0.2506

The findings suggest that active learning is an effective strategy for reducing computational cost while maintaining reasonable accuracy. The trade-off between data efficiency and prediction accuracy will be further analyzed in future work.

## Discussion

### Effectiveness of Active Learning in PINNs

Our experiments demonstrate that active learning can significantly reduce the amount of required training data while maintaining a reasonable level of accuracy in PINN-based solutions for PDEs. As observed in Table 1, the error rates for models trained with only 50% of the available data remain competitive with those trained on the full dataset, particularly for SHO and Wave1D. This indicates that a well-selected subset of training points, based on uncertainty estimation, can capture the underlying physics of the system effectively.

For Heat1D and Laplace2D, however, the benefits of active learning appear to be less pronounced, as the error rates remain relatively stable across different training set sizes. This suggests that the nature of the PDE influences the effectiveness of active learning strategies. Specifically, PDEs with strong spatial or temporal dependencies may require a more diverse selection of training points to achieve comparable performance.

### Trade-offs Between Data Efficiency and Accuracy

The results highlight an inherent trade-off between computational efficiency and prediction accuracy. While training on the full dataset consistently yields lower error rates, the differences in error between full-data and 50%-data models are not always substantial. This suggests that reducing the dataset size can lead to significant computational savings without a drastic loss in accuracy.

However, the choice of training data is critical. A naive random selection of 50% of the data may not yield the same benefits as an active learning-based selection. Future work should explore more advanced active learning strategies, such as reinforcement learning-based sampling or physics-informed importance weighting, to further optimize data efficiency.

### Comparison Across PDEs

Different types of PDEs exhibit varying levels of sensitivity to data reduction:

- **SHO:** Active learning proves to be highly effective, with the 50% data model achieving error rates close to the full-data model.
- **Heat1D:** Error rates remain relatively stable across different data sizes, suggesting that more training points may be necessary to capture fine-scale diffusion dynamics.
- **Wave1D:** The model trained with 50% data performs well, reinforcing the idea that wave propagation can be learned effectively with a subset of the training data.
- **Laplace2D:** Similar to Heat1D, error rates remain stable, indicating that more sophisticated sampling techniques may be required to improve efficiency.

### Limitations and Future Work

While this study provides insights into the benefits of active learning for PINNs, several limitations should be noted. First, the current approach relies on Monte Carlo dropout-based uncertainty estimation, which may not always be the most efficient or robust method. Future work should investigate alternative uncertainty quantification techniques, such as Bayesian PINNs or ensemble-based approaches.

Second, our experiments focus on relatively simple PDEs with known analytical solutions. Real-world problems, such as climate modeling or turbulence simulations, present additional complexities that may require modifications to the active learning framework. Integrating domain-specific knowledge into the sampling process could further enhance the effectiveness of PINNs for complex systems.

Finally, optimizing the computational cost of PINNs remains an open challenge. While reducing the training set size helps mitigate computational expense, other factors such as network architecture, optimization strategies, and hardware acceleration should be explored to further enhance efficiency.

### Conclusion

This study demonstrates that active learning can effectively reduce data requirements for PINN-based PDE solvers without significantly compromising accuracy. The experimental results indicate that the effectiveness of active learning strategies varies across different PDE types.

For instance, the wave equation exhibited relatively stable error rates across different active learning percentages, suggesting that even with reduced training data, the model can capture its well-structured dynamics efficiently. Similarly, for the simple harmonic oscillator, the lowest error was achieved with 40% of the training data, implying that an optimal balance between exploration and exploitation is necessary when selecting training points. In contrast, the heat equation and Laplace's

equation, which involve diffusion-dominated processes, showed minimal improvement from active learning, indicating that uncertainty-based sampling may be less effective for these types of PDEs.

These findings highlight the importance of selecting appropriate active learning strategies based on the underlying physical properties of the PDE. Future research will focus on refining active learning methodologies, including hybrid sampling strategies that combine uncertainty-based and physics-guided sampling techniques. Additionally, extending experiments to more complex, high-dimensional PDEs and exploring domain-adaptive architectures for PINNs will be key areas of investigation. Another promising direction is optimizing computational efficiency through parallelized training and improved uncertainty quantification methods.

By advancing generalizable PINN models with efficient data selection strategies, this work contributes to the broader goal of developing foundation models for physics-informed deep learning, paving the way for scalable and data-efficient PDE solvers across various scientific and engineering domains.

## Methods

### Physics-Informed Neural Networks (PINNs)

Physics-Informed Neural Networks (PINNs) leverage deep learning to solve Partial Differential Equations (PDEs) by embedding the governing equations into the loss function<sup>1</sup>. Unlike conventional numerical solvers, PINNs utilize neural networks to approximate solutions while simultaneously enforcing physical constraints through automatic differentiation. This approach allows PINNs to solve PDEs even in scenarios where sparse or noisy data is present.

The PINN used in this study is a fully connected feedforward neural network with hidden layers that apply the hyperbolic tangent (tanh) activation function. Mathematically, the network can be defined as:

$$\mathbf{u} = \mathcal{N}(\mathbf{x}; \theta), \quad (1)$$

where  $\mathbf{u}$  is the predicted solution,  $\mathbf{x}$  represents the input variables (space-time coordinates), and  $\theta$  denotes the network parameters. The network is trained by minimizing a composite loss function:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{PDE}}, \quad (2)$$

where  $\mathcal{L}_{\text{data}}$  measures the error between the predicted solution and known values from analytical solutions, and  $\mathcal{L}_{\text{PDE}}$  enforces the PDE residual.

### Loss Function Components

The data loss term is computed as the mean squared error (MSE) between the predicted and true values:

$$\mathcal{L}_{\text{data}} = \frac{1}{N} \sum_{i=1}^N (\mathcal{N}(\mathbf{x}_i; \theta) - u_i^{\text{true}})^2. \quad (3)$$

The PDE residual loss is calculated using automatic differentiation to enforce the governing equation:

$$\mathcal{L}_{\text{PDE}} = \frac{1}{M} \sum_{j=1}^M (\mathcal{F}(\mathcal{N}(\mathbf{x}_j; \theta)))^2, \quad (4)$$

where  $\mathcal{F}$  represents the PDE residual operator. By minimizing this loss, PINNs are trained to satisfy both the physical constraints imposed by the PDEs and the observed data.

### Partial Differential Equations (PDEs) Considered

We consider four different PDEs: Simple Harmonic Oscillator (SHO), 1D Heat Equation, 1D Wave Equation, and 2D Laplace Equation. These equations govern various physical phenomena, including oscillatory motion, diffusion processes, wave propagation, and equilibrium states. Their mathematical formulations and analytical solutions are presented below.

### **Simple Harmonic Oscillator (SHO)**

The simple harmonic oscillator describes oscillatory motion, such as the movement of a mass attached to a spring or the behavior of an electrical circuit. The governing equation is given by:

$$\frac{d^2u}{dt^2} + \omega^2 u = 0, \quad t \in [0, 2\pi], \quad (5)$$

where  $\omega$  represents the angular frequency of the oscillator. The general analytical solution for this equation is:

$$u(t) = A \cos(\omega t) + B \sin(\omega t), \quad (6)$$

where  $A$  and  $B$  are constants determined by the initial conditions. This solution demonstrates that SHO exhibits periodic motion with a fundamental frequency  $\omega$ .

### **1D Heat Equation**

The heat equation models the diffusion of heat (or other quantities) over time. It describes how temperature evolves in a given domain due to thermal conduction. The equation is expressed as:

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0, \quad x, t \in [0, 1], \quad (7)$$

where  $\alpha$  is the thermal diffusivity constant, which determines how quickly heat spreads. The analytical solution for an initial sinusoidal temperature distribution is:

$$u(x, t) = e^{-(\pi^2 \alpha t)} \sin(\pi x). \quad (8)$$

This solution shows that the amplitude of the temperature profile decays exponentially over time due to diffusion, with a rate controlled by  $\alpha$ .

### **1D Wave Equation**

The wave equation governs the propagation of waves, such as sound waves, water waves, or electromagnetic waves. The standard form of the equation is:

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0, \quad x, t \in [0, 1], \quad (9)$$

where  $c$  represents the wave speed. The analytical solution for a standing wave with fixed boundaries is:

$$u(x, t) = \sin(\pi x) \cos(\pi c t). \quad (10)$$

This solution represents a sinusoidal oscillation where the wave maintains a fixed spatial pattern while oscillating in time. The frequency of oscillation is determined by  $\pi c$ .

### **2D Laplace Equation**

The Laplace equation describes steady-state phenomena such as electrostatic potentials, fluid flow, and heat conduction in equilibrium. The equation is given by:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad x, y \in [0, 1]. \quad (11)$$

This equation states that the sum of second-order spatial derivatives must be zero, meaning that the function  $u(x, y)$  has no local extrema unless dictated by boundary conditions. The analytical solution for a specific boundary condition is:

$$u(x, y) = \sinh(\pi y) \sin(\pi x). \quad (12)$$

This solution indicates that the function smoothly varies within the domain, satisfying the Laplace equation at all interior points while conforming to prescribed boundary values.

## Active Learning Strategy

To optimize data efficiency, we employ an active learning strategy based on Monte Carlo (MC) Dropout Uncertainty Sampling. The key steps in our active learning framework are as follows:

1. Train an initial PINN using a full dataset.
2. Use MC Dropout to generate multiple predictions at each input point:

$$\hat{u}_i^{(k)} = \mathcal{N}(\mathbf{x}_i; \theta^{(k)}), \quad k = 1, \dots, K. \quad (13)$$

where  $\theta^{(k)}$  represents the stochastic dropout-enabled parameters.

3. Compute the standard deviation of the predictions to estimate epistemic uncertainty:

$$\sigma_i = \sqrt{\frac{1}{K} \sum_{k=1}^K \left( \hat{u}_i^{(k)} - \bar{u}_i \right)^2}, \quad (14)$$

where  $\bar{u}_i$  is the mean prediction.

4. Select the top  $p\%$  most uncertain points for retraining:

$$\mathbf{x}_{AL} = \arg \max_{\mathbf{x}_i} \sigma_i. \quad (15)$$

5. Retrain the PINN using only the selected data points.

## Experimental Setup

The PINN architecture consists of three hidden layers with 20 neurons each, using the tanh activation function. A dropout rate of 10% is applied during training. The network is optimized using the Adam optimizer with a learning rate of 0.01 for 2000 epochs.

To evaluate the effectiveness of active learning, models were trained with different percentages of selected data (10%, 20%, 30%, 40%, and 50%), and the resulting error rates were compared against the full dataset model.

## Evaluation Metrics

The models are evaluated using the mean absolute error (MAE) between the predicted and analytical solutions:

$$MAE = \frac{1}{N} \sum_{i=1}^N |u_i^{\text{pred}} - u_i^{\text{true}}|. \quad (16)$$

This metric provides a direct measure of the accuracy of the PINN across different PDEs and active learning percentages.

## References

1. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
2. Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
3. Cai, S., Wang, G. & Karniadakis, G. E. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mech. Sinica* **37**, 172–180 (2021).
4. Yu, B., Lu, L. & Karniadakis, G. E. Gradient-enhanced physics-informed neural networks for inverse problems in high-contrast media. *Comput. Methods Appl. Mech. Eng.* **393**, 114823 (2022).
5. Wang, S., Teng, Y. & Perdikaris, P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. on Sci. Comput.* **43**, A3055–A3081 (2021).

6. Kissas, G. *et al.* Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4d flow mri data using physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.* **358**, 112623 (2020).
7. Settles, B. Active learning literature survey. *Univ. Wisconsin-Madison Dep. Comput. Sci.* **1648**, 3 (2009).
8. Yang, L., Meng, X. & Karniadakis, G. E. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *J. Comput. Phys.* **425**, 109913 (2021).

## Acknowledgements (not compulsory)

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)].

## Author contributions statement

Keon Vin Park conducted the primary analysis, developed the methodology, and drafted the manuscript.