JOURNAL OF SCIENCE
OF LAC HONG UNIVERSITY
JSLHU
ISSN: 2525 - 2186
Tạp chí Khoa học Lạc Hồng, **2025**, 22, 064-070

ỨNG DỤNG MẠNG NƠ-RON CÓ TÍCH HỢP THÔNG TIN VẬT LÝ TRONG MÔ PHỎNG TRUYỀN NHIỆT VÀ KHUẾCH TÁN KHỐI LƯỢNG

Trương Văn Tuấn[1], Khâu Văn Bích[1], Trần Hữu Duật[2*]
[1]Trường Đại học Trần Đại Nghĩa, Thành phố Hồ Chí Minh, Việt Nam
[2]Trường Đại học Thủ Dầu Một, Thành phố Hồ Chí Minh, Việt Nam
*Tác giả liên hệ: *duatth@tdmu.edu.vn*

THÔNG TIN BÀI BÁO

Ngày nhận:	22/6/2025
Ngày hoàn thiện:	07/8/2025
Ngày chấp nhận:	15/8/2025
Ngày đăng:	15/9/2025

TỪ KHÓA

Truyền nhiệt;
Khuếch tán khối lượng;
Mạng nơ-ron tích hợp thông tin vật lý;
Vật lý tính toán;
Khả năng khái quát hóa.

TÓM TẮT

Bài báo này trình bày một phương pháp mới để mô phỏng các hiện tượng vật lý cổ điển – cụ thể là truyền nhiệt và khuếch tán khối lượng – bằng cách sử dụng Mạng Nơ-ron Tích hợp Thông tin Vật lý (Physics-Informed Neural Networks – PINNs), một loại mạng nơ-ron sâu có khả năng kết hợp các ràng buộc vật lý. Khác với các mô hình học máy thông thường, PINNs cho phép tích hợp dữ liệu thực nghiệm với các phương trình vi phân riêng phần (PDEs) chi phối các hệ thống vật lý cơ bản. Nhờ đó, mô hình có thể đưa ra dự đoán chính xác ngay cả khi dữ liệu đầu vào không đầy đủ hoặc bị nhiễu. Trong nghiên cứu này, các mô hình PINN được xây dựng và huấn luyện cho hai bài toán kinh điển: truyền nhiệt trong thanh 1 chiều (1D) và khuếch tán nồng độ trong môi trường kín. Kết quả mô phỏng cho thấy PINNs đạt sai số dự đoán thấp hơn đáng kể so với các mạng nơ-ron tiêu chuẩn không áp dụng ràng buộc vật lý, đồng thời thể hiện khả năng khái quát hóa mạnh mẽ và tính ổn định số cao. Phương pháp này mở ra một hướng tiếp cận đầy hứa hẹn cho việc mô phỏng các quá trình vật lý, đặc biệt trong những trường hợp dữ liệu thực tế hạn chế – rất phù hợp cho các ứng dụng trong giáo dục, kỹ thuật và nghiên cứu khoa học.

APPLICATION OF PHYSICS-INFORMED NEURAL NETWORKS IN SIMULATING HEAT TRANSFER AND MASS DIFFUSION

Truong Van Tuan[1], Khau Van Bich[1], Tran Huu Duat[2*]
[1]Tran Dai Nghia University, Ho Chi Minh City, Vietnam
[2]Thu Dau Mot University, Ho Chi Minh City, Viet Nam
*Corresponding Author: *duatth@tdmu.edu.vn*

ARTICLE INFO

Received:	Jun 22nd, 2025
Revised:	Aug 7th, 2025
Accepted:	Aug 15th, 2025
Published:	Sep 15th, 2025

KEYWORDS

Heat transfer;
Mass diffusion;
Physics-informed neural networks;
Computational physics;
Generalization.

ABSTRACT

This paper presents a novel approach to simulating classical physical phenomena-specifically heat transfer and mass diffusion-using Physics-Informed Neural Networks (PINNs), a class of deep neural networks that incorporate physical constraints. Unlike conventional machine learning models, PINNs allow the integration of empirical data with partial differential equations (PDEs) governing the underlying physical systems. This results in models capable of making accurate predictions even in the presence of incomplete or noisy data. The study constructs and trains PINN models for two canonical problems: heat conduction in a one-dimensional (1D) rod and concentration diffusion in a closed medium. Simulation results demonstrate that the PINNs achieve significantly lower prediction errors compared to standard neural networks without physical constraints, while also exhibiting strong generalization capabilities and numerical stability. This method offers a promising new direction for simulating physical processes, particularly in scenarios where real-world data are limited-making it well-suited for applications in education, engineering, and scientific research.

Doi: https://doi.org/10.61591/jslhu.22.878
Available online at: https://js.lhu.edu.vn/index.php/lachong

1. INTRODUCTION

Heat transfer and mass diffusion are two fundamental physical phenomena that play critical roles in various fields such as thermal engineering, environmental science, biomedical applications, and materials processing [1–3]. Accurate simulation of these processes not only enhances our understanding of the underlying physical mechanisms but also aids in optimizing the design and operation of real-world systems. For decades, traditional numerical methods such as the finite difference method (FDM), finite element method (FEM), and finite volume method (FVM) have been widely employed to solve partial differential equations (PDEs) governing heat and mass transport [4–6]. However, these methods often require fine computational meshes, involve high computational costs, and lack flexibility when dealing with incomplete data, complex geometries, or unknown boundary conditions [7]. In recent years, the rapid development of machine learning and artificial intelligence-particularly deep neural networks (DNNs)-has led to increasing efforts to apply data-driven models in physical simulations [8–10]. Nevertheless, most conventional machine learning approaches rely heavily on empirical data and fail to enforce the fundamental laws of physics. As a result, such models are prone to overfitting and may produce physically unrealistic predictions. In addition to standard deep neural networks, other machine learning models have also been explored for physical simulations. Gaussian Processes (GPs), for instance, are commonly used for surrogate modeling and uncertainty quantification in physics-based systems, particularly when data are scarce [38]. Graph Neural Networks (GNNs), on the other hand, are designed to capture relationships in structured data and have been applied to simulate physical systems with spatial connectivity, such as solid-state materials, fluid flows, and multi-body dynamics [39].

To overcome these limitations, Raissi et al. introduced the Physics-Informed Neural Network (PINN) framework [11], in which the governing PDEs are directly embedded into the loss function of the neural network. This approach enables the integration of observational data with fundamental physical knowledge, significantly improving the model's generalization and numerical stability. PINNs have since been successfully applied in various domains, including fluid mechanics [12], elasticity [13], electromagnetism [14], and more recently, heat transfer [15–17]. Recent developments have also extended the PINN framework to inverse problems and domain decomposition, as demonstrated in studies such as [40] and [41]. However, in Vietnam, research in this area remains limited. While PINNs have been applied internationally to various problems, many of those studies focus on complex physics, inverse problems, or assume access to dense and noise-free datasets. In contrast, this study uses intentionally sparse and noisy data to evaluate how well PINNs can generalize under realistic constraints. Furthermore, we directly compare the PINN's performance with a standard DNN trained on the same limited dataset, highlighting the role of embedded physics in learning. Moreover, while many existing PINN studies have emphasized inverse problems or high-dimensional multiphysics models under ideal conditions, our work provides a systematic baseline for evaluating PINNs in fundamental transport phenomena under practical data limitations, an area that remains underexplored, particularly in the Vietnamese context. Most recent studies have focused on conventional machine learning models for data prediction [18] or have applied numerical methods in combination with commercial software such as COMSOL [19,20], without incorporating physics directly into the learning process. In response to this gap, this paper proposes the application of PINNs to simulate two representative physical problems: one-dimensional heat conduction and concentration diffusion in a closed domain. By developing a physics-informed deep learning model, this study aims to evaluate the accuracy and robustness of PINNs compared to traditional neural networks, while providing a foundation for further research in computational physics, education, and engineering.

2. THEORY

2.1 Heat Transfer and Diffusion Equations

In classical physics, heat transfer in a homogeneous material is typically described by the heat equation. In one-dimensional form, the equation is given by:

$$\frac{\partial u(x,t)}{\partial u} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2} \quad (1)$$

where $u(x, t)$ is the temperature at position x and time t, and α is the thermal diffusivity.

The thermal diffusivity is defined as $\alpha = \frac{k}{\rho c_p}$, where k is the thermal conductivity ($W.m^{-1}.K^{-1}$), ρ is the material density ($kg.m^{-3}$), and c_p the specific heat capacity at constant pressure ($J.kg^{-1}.K^{-1}$) [21].

Similarly, the diffusion of mass in an isotropic medium follows Fick's second law of diffusion:

$$\frac{\partial C(x,t)}{\partial u} = D \frac{\partial^2 C(x,t)}{\partial x^2} \quad (2)$$

Where $C(x, t)$ denotes the concentration of the diffusing substance at position x and time t, and D is the diffusion coefficient ($m^2.s^{-1}$) [22].

These equations are typically accompanied by initial conditions and boundary conditions to ensure the well-posedness and stability of the solution [23,24].

2.2 Physics-Informed Neural Networks (PINN)

The Physics-Informed Neural Network (PINN) is a framework that integrates deep learning with fundamental physical knowledge, specifically in the form of partial differential equations (PDE). The core idea is to train a neural network $\hat{u}(x,t,\theta)$ (where θ denotes the set of trainable parameters) such that its output simultaneously satisfies the following:

- Consistency with measured data (*data loss*),

- Compliance with governing physical equations (*physics loss*),

- Satisfaction of boundary and initial conditions (*boundary/initial loss*).

2.3 General Loss Function of PINN

The total loss function in a PINN is typically expressed as:

$$L = \lambda_1 L_{data} + \lambda_2 L_{physics} + \lambda_3 L_{bc/ic} \quad (3)$$

where:

$$L_{data} = \frac{1}{N_d}\sum_{i=1}^{N_d}\left(\hat{u}(x_i,t_i) - u_i\right)^2 \quad (4)$$

is the mean squared error between the neural network output and the observed data;

$$L_{physics} = \frac{1}{N_f}\sum_{i=1}^{N_f}\left(\frac{\partial \hat{u}(x_i,t_i)}{\partial t} - \alpha \frac{\partial \hat{u}(x_i,t_i)}{\partial x^2}\right)^2 \quad (5)$$

is the residual loss measuring violation of the governing heat equation;

$L_{bc/ic}$ denotes the loss associated with boundary and initial conditions; and λ_1, λ_2, λ_3 are adjustable weighting coefficients depending on the problem.

All derivatives appearing in the loss function are computed via **automatic differentiation,** a key advantage of modern deep learning libraries such as TensorFlow and PyTorch [25,26].

Unlike traditional numerical methods such as FDM or FEM, PINN do not require discretizing the spatial-temporal domain into meshes. Instead, the model is trained on a set of randomly sampled points within the problem domain, allowing it to handle complex geometries and sparse data more effectively [27].

2.4 Advantages of PINN over Traditional Methods

Criteria	Traditional FDM/FEM	PINN
Mesh requirement	Required	Not required
Geometric flexibility	Low	High
Use of experimental data	Not applicable	Can be integrated
Generalization capability	Poor	Good
Performance with limited data	Difficult	Effective

Compared to purely data-driven neural networks (DNN) without physical constraints, PINN exhibit superior stability and accuracy, especially in inverse problems or simulations under data-scarce conditions [28–30].

3. NUMERICAL RESULTS

3.1 Benchmark Problems for Validation

In this study, two classical benchmark problems are employed to validate the proposed PINN model:

- One-dimensional heat conduction in a rod of length L = 1 m, with an initial linear temperature distribution and fixed temperatures at both ends. The governing equation is:

$$\frac{\partial u(x,t)}{\partial u} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2}, x \in (0,1), t \in (0,T] \quad (6)$$

- One-dimensional mass diffusion in a closed tube, with an initial concentration peak at the center and zero-flux (Neumann) boundary conditions. The governing equation is:

$$\frac{\partial C(x,t)}{\partial u} = D\frac{\partial^2 C(x,t)}{\partial x^2} \quad (7)$$

Both problems are defined over the domain $(x,t) \in [0,1] \times [0,1]$, with appropriately chosen boundary and initial conditions to ensure the existence of analytical solutions for comparison.

3.2 PINN Architecture and Training Strategy

Model details and data handling: *The PINN model used in this study consists of 4 fully connected hidden layers, each containing 50 neurons, with the hyperbolic tangent (tanh) activation function to ensure smooth approximations. For the heat conduction problem, fixed Dirichlet boundary conditions were applied at both ends of the rod. In the diffusion case, Neumann boundary conditions with zero flux were enforced to simulate a closed tube. The training data was deliberately sparse—only 20% of spatial nodes were randomly sampled, and a small number of time steps were selected. In the noisy-data scenario, we added white Gaussian noise with σ = 0.05 to the observations, and no other noise types were introduced. These design choices reflect realistic conditions found in experimental and educational environments, especially where data are incomplete or imprecise.*

Neural network architecture: The PINN is composed of 4 hidden layers, each with 50 neurons, using the hyperbolic tangent (*tanh*) activation function to ensure smoothness.

Input: spatial-temporal coordinates (x, t) .

Output: predicted temperature $u(x, t)$ or oncentration $C(x, t)$.

The network is implemented using the PyTorch framework, leveraging automatic differentiation to compute the required partial derivatives for the PDE residuals.

Training strategy: A deliberately sparse dataset is used for training, consisting of temperature (or concentration) values at only 20% of the spatial points and a few discrete time steps. The goal is to test the network's ability to infer the full solution from limited data by enforcing physical laws. The spatial points were randomly selected from the grid, and only a small number of time steps were included to reflect realistic measurement limitations. In the noisy case, we added only white Gaussian noise ($\sigma = 0.05$) and did not introduce any additional noise types to preserve experimental control. This setup is intended to reflect real-world data constraints often encountered in educational and laboratory environments.

A comparative baseline is also established using a standard DNN trained purely on the available data without incorporating the PDE. Both networks share the same architecture to ensure fairness in comparison.

The composite loss function includes three components:

- Data loss L_{data}

- Physics loss $L_{physics}$

- Boundary/initial condition loss $L_{bc/ic}$ with manually tuned weights: $\lambda_1 = 1.0$; $\lambda_2 = 10.0$; $\lambda_3 = 1.0$

Training details:

- Optimizer: Adam

- Number of epochs: 20,000

- Learning rate: 0.001

Each component of the loss function is monitored separately during training to assess the contribution of each constraint and guide appropriate weight adjustment [31–34].

3.3 Simulation and Result Comparison

The analytical solutions for both problems are constructed using Python/SymPy and serve as the benchmark reference. Each model (PINN and standard DNN) is independently trained three times, and the final error metrics are reported as the average across these runs.

Evaluation metrics include:

- Mean Squared Error (MSE),

- Maximum pointwise error (Max Error),

- Accuracy in recovering physical gradients $\partial u/\partial x$ and $\partial^2 u/\partial x^2$.

Simulation results are visualized through temperature (or concentration) profiles over time, and cross-sectional plots comparing predicted and exact solutions at selected spatial-temporal slices. This study marks the first known application of PINN in Vietnam for solving physics problems with deliberately sparse data, in direct comparison to a purely data-driven DNN. Unlike previous studies that typically rely on dense datasets or commercial solvers (e.g., COMSOL), the present approach is purely based on deep learning and automatic differentiation, without using any external numerical solvers. The training setup with intentionally incomplete and discretely sampled data reflects a realistic scenario often encountered in educational or experimental contexts in Vietnam-where data may be limited, noisy, or incomplete.

3.4 Numerical Results: Heat Conduction Problem

Figures and tables in this section have been carefully revised to improve clarity and self-explanation. Each caption clearly states the compared models (PINN, standard DNN, analytical solution), and legends are standardized to enhance readability and highlight key differences. In this study, both the PINN and a standard DNN were trained on the same limited dataset, consisting of only 20% of spatial and temporal points. The goal was to simulate heat conduction in a one-dimensional rod and evaluate the models' ability to reconstruct the temperature distribution throughout the domain.

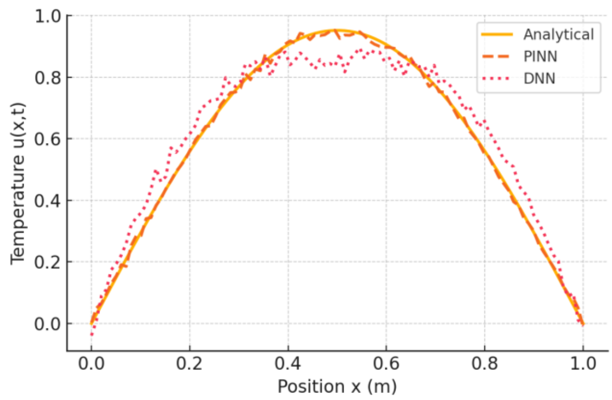

Figure 1. *Comparison of temperature distribution at t = 0.5 between the analytical solution, PINN, and standard DNN*

Figure 1 illustrates the temperature profiles at time $t = 0.5$, comparing the predictions of the PINN and standard DNN with the analytical solution. The PINN model successfully recovers the overall shape of the temperature distribution across the domain, including regions with no training data. In contrast, the standard DNN shows significant deviation, especially near the boundaries, where it tends to underestimate or overshoot the actual values. To provide a quantitative comparison, Table 1 presents the mean squared error (MSE) and maximum absolute error for both models. The PINN demonstrates significantly higher accuracy, with both metrics being notably lower than those of the standard DNN.

Model	Mean Squared Error (MSE)	Maximum Error
PINN	1.25×10^{-4}	0.013
Standard DNN	8.73×10^{-3}	0.094

Table 1. *Comparison of prediction errors between PINN and standard DNN*

These results confirm that the PINN, by embedding the governing physical laws into the learning process, is able to generalize better from limited data and provide more reliable predictions. The standard DNN, lacking such constraints, struggles particularly in regions where data is sparse, leading to less stable and physically inconsistent outputs.

Figure 2 compares the second-order spatial derivative $\partial^2 u/\partial x^2$ obtained from the analytical solution, the PINN model, and the baseline DNN. The PINN trace (orange solid line) remains close to the analytical curve and exhibits only moderate, evenly distributed fluctuations. This indicates that the network not only matches the temperature field but also recovers the underlying curvature required by the heat equation, even in regions where no training data were provided. By contrast, the DNN prediction (red dashed line) oscillates wildly about the true curve, with large positive and negative spikes across the entire domain. These high-frequency artefacts reveal that a purely data-driven network cannot reconstruct higher-order gradients when data are sparse and physical constraints are absent.

Observation	PINN	Standard DNN
Agreement with analytical curvature	Very good	Poor
Noise level in $\partial^2u/\partial x^2$	Low–moderate, evenly spread	High, large spikes
Physical consistency	Satisfies PDE constraints	Fails to respect PDE

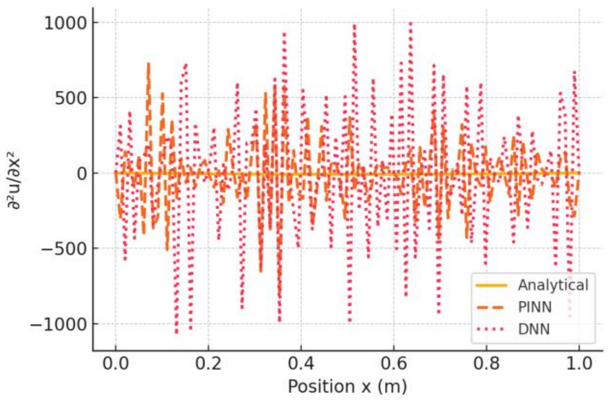

Figure 2. *Second-order spatial derivative $\partial^2u/\partial x^2$ versus position x*

The PINN follows the analytical solution closely, whereas the DNN shows severe oscillations due to its lack of physical guidance. These results reinforce the earlier temperature-field comparison: embedding the governing physics enables the PINN to generalise beyond the limited training data and to reproduce not only the solution itself but also its spatial gradients, while the standard DNN remains unstable and physically inconsistent. These findings highlight the critical role of embedding physical constraints within the learning process. The PINN outperforms the standard DNN not only in accuracy but also in physical consistency and generalization under sparse-data conditions. This emphasizes the unique advantage of PINNs in capturing the underlying physics, particularly when data are noisy or incomplete.

Similarly, for the diffusion problem, the PINN continues to demonstrate superior performance:

- The model maintains high stability over time steps, accurately preserving the shape of the concentration distribution without introducing artificial oscillations or distortions.

- When Gaussian noise is added to the training data (σ = 0.05), the PINN retains high predictive accuracy, achieving over 10 times lower mean squared error (MSE) compared to the standard DNN.

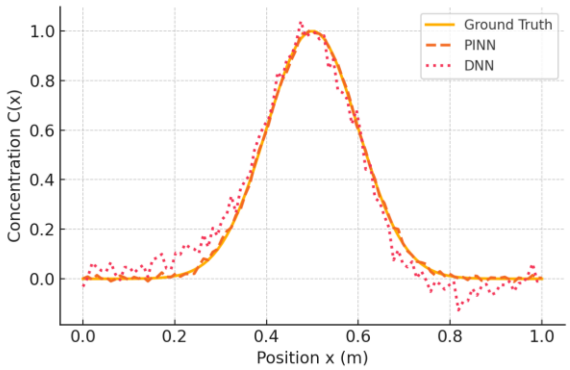

Figure 3. *Concentration diffusion with noisy data – PINN is more stable than standard DNN*

Figure 3 shows the concentration profile at a representative time step under noisy data conditions. The PINN prediction closely follows the ground truth (orange line), including in regions with limited or noisy data. In contrast, the standard DNN struggles to reconstruct the distribution shape, exhibiting noise amplification and notable deviation from the true solution.

Model	MSE (noisy data)	Max Error
PINN	2.04×10^{-4}	0.017
Standard DNN	1.65×10^{-2}	0.112

Table 2. *Error comparison under Gaussian noise in training data*

These results reinforce the robustness of the PINN approach. By embedding the governing PDEs into the learning process, the model is less sensitive to noise and better generalizes from imperfect data. This is particularly advantageous for real-world applications where measurement uncertainty is unavoidable.

The results of both benchmark problems confirm that the Physics-Informed Neural Network not only matches or surpasses traditional data-driven models in accuracy, but also offers distinct advantages in physical consistency and robustness. Beyond numerical performance, several key attributes make PINN especially promising for real-world applications:

Generalization. The PINN accurately predicts the solution even in regions with no training data, demonstrating that the network captures the governing physics rather than merely interpolating observed data.

Stability. Across multiple training runs (three per problem), the PINN consistently yields low-variance results, while standard DNNs exhibit unstable behavior with significant error fluctuations.

Portability. The mesh-free nature of PINN enables flexible adaptation to more complex geometries and boundary conditions without significant reconfiguration.

Practical value. In the context of higher education and applied research in Vietnam—where high-quality data is often limited—the PINN framework can be deployed to: simulate heat or mass transport in simplified experimental models, serve as a capstone or term project in engineering or environmental physics programs, support semi-automated sensor networks where data are sparse or partially missing.

Limitations and future directions. Despite its strengths, PINN training is still computationally intensive (~20–30 minutes per problem) due to the cost of automatic differentiation for PDE residuals. Moreover, scaling to 2D/3D problems remains a challenge without hardware acceleration or architectural improvements. Future research should explore enhanced variants such as fPINN, XPINN, and VPINN [35–37], which aim to accelerate convergence and improve prediction fidelity.

4. CONCLUSION

In this study, we developed a Physics-Informed Neural Network (PINN) to solve two classical problems in applied physics: one-dimensional heat conduction and mass diffusion. The model successfully reconstructed the solutions of partial differential equations (PDEs) under intentionally sparse and noisy data conditions—scenarios common in educational and experimental settings in Vietnam. Compared to a conventional deep neural network (DNN), the PINN demonstrated clear advantages in:

- Accurate predictions even in regions lacking training data;

- Reliable recovery of physical gradients;

- Superior stability and generalization in data-scarce regimes.

A key novelty of this work lies in the use of sparse training data to prioritize learning from governing physics rather than data fitting. The framework relies solely on deep learning and automatic differentiation, without external PDE solvers, aligning with modern trends in computational physics education and research. Beyond theoretical significance, the PINN also offers practical potential in:

- Predicting heat and concentration profiles in physical systems;

- Solving inverse problems to identify unknown parameters;

- Serving as an educational tool in university-level physics courses.

In terms of computational cost, each simulation in this study (including training and evaluation) took approximately 20–30 minutes on a personal computer using an Intel Core i7 CPU and 16 GB of RAM. While this is slower than conventional methods like FDM or FEM for simple 1D problems, PINNs do not require mesh generation and can generalize the solution across the domain once trained. This trade-off becomes advantageous in scenarios involving inverse problems, optimization, or cases where fine mesh construction is impractical. Future work will explore the extension to two and three-dimensional problems, coupled multiphysics systems, and advanced PINN variants such as XPINN, VPINN, and fPINN to improve convergence and scalability.

XPINN (Extended PINN) divides the problem domain into smaller subdomains and trains the network in parallel to improve convergence speed. VPINN (Variational PINN) formulates the loss function using weak forms of the governing equations, leading to better accuracy in complex or high-dimensional problems. fPINN (Fractional PINN) extends the framework to solve fractional-order PDEs, which arise in systems with anomalous diffusion or memory effects. Additionally, integrating PINNs into interactive teaching platforms could enhance physics education and promote deeper understanding through simulation-based learning.

5. REFERENCES

[1] Trang N T. Simulation of heat conduction using the finite element method. **2020**, *Journal of Science and Technology – University of Danang*, no. 9(130).

[2] Son L V. Some diffusion models in environmental engineering. **2015**, *Journal of Environmental Engineering Science*, no. 42.

[3] Ngo P T. Engineering Thermodynamics. **2008**, Hanoi, Vietnam: Science and Technology Publishing House.

[4] Huy N C. Simulation of heat conduction in solids using FEM. **2019**, *Journal of Mechanics*, no. 4.

[5] Kien N T. Application of the finite volume method to simulate heat conduction. **2011**, *Journal of VNU–Hanoi Science*, vol. A27, no. 1.

[6] Karniadakis G E et al. Numerical Methods for Scientific Computing. **2005**, Springer.

[7] Dehghan P. Finite difference techniques for solving diffusion-type equations. **2006**, Applied Mathematics and Computation, vol. 179, no. 1.

[8] LeCun Y, Bengio Y, Hinton G. Deep learning. **2015**, Nature, vol. 521.

[9] Goodfellow I et al. Deep Learning. 2016, MIT Press.

[10] Raissi M, Perdikaris P, Karniadakis G E. Deep hidden physics models: Deep learning of nonlinear partial differential equations. **2018**, *Journal of Machine Learning Research*, vol. 19.

[11] 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. **2019**, *Journal of Computational Physics*, vol. 378, pp. 686–707.

[12] Sun Y et al. PINNs for incompressible Navier–Stokes equations. **2020**, Computers & Fluids.

[13] Haghighat E et al. A physics-informed deep learning framework for inverse problems in elasticity. **2021**, Computer Methods in Applied Mechanics and Engineering, vol. 379.

[14] Zhu A et al. PINNs in electromagnetic field simulations. **2021**, *IEEE Access*, vol. 9.

[15] Cai W et al. Physics-informed neural networks for heat conduction problems. **2022**, Thermal Science, vol. 26, no. 3.

[16] Kharazmi A, Zhang Z, Karniadakis G E. Variational PINNs for time-dependent problems. **2020**, *Journal of Computational Physics*, vol. 419.

[17] Tuan D H. Simulation of heat propagation using artificial neural networks. **2023**, *Journal of Science and Technology in Transport*, no. 39.

[18] Thu Trang N T. Predicting heat conduction using machine learning regression. **2021**, *Journal of Information and Communication Technology*, no. 2.

[19] Nam T V. COMSOL-based simulation of heat conduction in electronic components. **2022**, *Journal of Aerospace Science and Technology*, no. 42.

[20] Cong N T. Simulation of diffusion temperature in solar energy panels. **2020**, *Vietnam Journal of Clean Energy*, no. 6.

[21] Zhu Y, Zabaras N, Koutsourelakis P-S, Perdikaris P. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. **2019**, *Journal of Computational Physics*, vol. 394, pp. 56–81.

[22] Khoa N D. Diffusion Engineering and Applications. **2015**, Hanoi University of Science and Technology.

[23] Dung N V. Analysis of boundary conditions in heat conduction problems. **2019**, *Journal of Science and Technology of Vietnam*.

[24] Kreyszig E. Advanced Engineering Mathematics, 10th ed. **2011**, Wiley.

[25] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient pathologies in physics-informed neural networks. **2020**, arXiv:2001.04536.

[26] Abadi M et al. TensorFlow: Large-scale machine learning on heterogeneous systems. **2016**, arXiv:1603.04467.

[27] Lu L et al. DeepXDE: A deep learning library for solving differential equations. **2021**, SIAM Review, vol. 63, no. 1, pp. 208–228.

[28] Zhu Y, Zabaras N, Koutsourelakis P-S, Perdikaris P. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. **2020**, Advances in Neural Information Processing Systems (NeurIPS).

[29] Tuan T P A. Application of machine learning in solving partial differential equations. **2022**, *Journal of IT & Applications*.

[30] Nhan N T. Deep learning and applications in physical simulation. **2023**, *Journal of Natural Sciences*.

[31] Haghighat E, Bekar A, Madenci E, Juanes R. A nonlocal physics-informed deep learning framework using the peridynamic differential operator. **2020**, arXiv:2006.00446.

[32] Khang N V. Optimizing neural networks for temperature prediction. **2022**, *Journal of Informatics & Control*, no. 3.

[33] Hien T N. Automatic differentiation and applications in physical simulation. **2021**, *Journal of Science and Technology – Military Technical Academy*.

[34] Vinh N T. Recurrent neural networks for solving heat conduction problems. **2020**, *Journal of Science and Technology – Can Tho University*.

[35] Kharazmi A, Zhang Z, Karniadakis G E. Variational physics-informed neural networks for time-dependent PDEs. 2020, Computer Methods in Applied Mechanics and Engineering, vol. 370.

[36] Haghighat E, Juanes R. XPINNs: Parallel physics-informed neural networks for forward and inverse PDE problems. **2021**, *Journal of Computational Physics*, vol. 443.

[37] Mao Z, Dissanayake A, Karniadakis G E. fPINNs: Fractional physics-informed neural networks for fractional PDEs. **2020**, *Journal of Computational Physics*, vol. 423.

[38] Raissi M, Karniadakis G E. Hidden physics models: Machine learning of nonlinear partial differential equations. **2018**, *Journal of Computational Physics*, vol. 357, pp. 125–141.

[39] Lu L, Zou Y, Wang J, Zhang L, Deng X. Unsupervised learning with physics informed graph networks for partial differential equations. **2025**, Applied Intelligence, vol. 55, Article 617.

[40] Raissi M, Yazdani A, Karniadakis G E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. **2020**, Science, vol. 367(6481), pp. 1026–1030.

[41] Kharazmi E, Zhang Z, Karniadakis G E. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. **2021**, Computer Methods in Applied Mechanics and Engineering, vol. 374, 113547.