Advances in Differential Equations and Control Processes 2026, 33(1), 3584.
doi: https://doi.org/10.59400/adecp3584

Article

Physics-informed neural networks for solving steady-state heat conduction inverse problems

Tangwei Liu[1,*], Qiong Zou[1], Xiaoqing Ruan[1,2], Dingding Yan[1], Jeevan Kafle[3], Zhongzhou Lan[4]

[1] School of Science, East China University of Technology, Nanchang 330013, China
[2] Mathematics Teaching and Research Group, Sheshan School, Shanghai 201602, China
[3] Central Department of Mathematics, Tribhuvan University, Kathmandu 44600, Nepal
[4] School of Computer Information Management, Inner Mongolia University of Finance and Economics, Hohhot 010070, China
*** Corresponding author:** Tangwei Liu, twliu@ecut.edu.cn

CITATION

Liu T, Zou Q, Ruan X, et al. Physics-informed neural networks for solving steady-state heat conduction inverse problems. Advances in Differential Equations and Control Processes. 2026; 33(1): 3584. https://doi.org/10.59400/adecp3584

ARTICLE INFO

Received: 30 July 2025
Revised: 6 February 2026
Accepted: 13 February 2026
Available online: 5 March 2026

Abstract: This paper investigates numerical methods for a class of three-dimensional steady-state heat conduction inverse problems. By employing Physics-Informed Neural Networks (PINNs), the three-dimensional heat conduction inverse problems are reformulated as optimization problems with respect to a properly defined loss function. Two cases with different additional conditions are considered: one incorporates an additional boundary temperature gradient condition, while the other involves an additional partial internal temperature measurement. Corresponding efficient algorithms are developed to solve the resulting optimization problems. To optimize the performance of the proposed numerical framework, systematic sensitivity analyses are performed to rigorously justify the selection of key hyperparameters (e.g., activation functions and network architecture). Additionally, to validate the mathematical effectiveness and noise robustness of the algorithm, this study primarily employs synthetic data with controlled noise levels for quantitative evaluation. Numerical results demonstrate that the proposed method can efficiently and accurately approximate the solutions to the three-dimensional (3D) heat conduction inverse problems for both polynomial and non-polynomial cases. In addition, a theoretical analysis is provided to interpret the method's stability against noise amplification. Future work will focus on applying the proposed framework to real-world field data to further validate its practical engineering value.

Keywords: physics-informed neural networks; three-dimensional heat conduction; inverse problem; loss function; numerical simulation

1. Introduction

Solving forward problems for partial differential equations (PDEs) remains a fundamental task in mathematical physics. Recent advances have significantly enhanced our ability to address such problems, particularly when dealing with multi-dimensional spaces, intricate geometries, and coupled multi-physics systems [1, 2]. Meanwhile, theoretical analyses of regularity in degenerate equations provide a solid theoretical foundation for these numerical explorations [3]. Advances in computational mathematics have facilitated the application of diverse analytical and numerical techniques, such as the domain decomposition method [4] and conventional approaches including the finite element method (FEM), finite volume

schemes, and spectral techniques, which have established well-developed theoretical frameworks encompassing multi-grid acceleration strategies and adaptive mesh refinement algorithms. Contemporary research frontiers predominantly concentrate on unstructured grid manipulation, multi-scale coupled computations, and emerging solution paradigms rooted in Physics-Informed Neural Networks (PINNs) [5].

Inverse problems are difficult to solve due to fundamental issues such as ill-posedness and non-uniqueness [6]. Consequently, research across various disciplines has focused on using regularization techniques to stabilize these ill-posed systems [7]. Although Tikhonov regularization remains the standard practice, Luo et al. introduced a novel approach specifically designed for handling 3D transient inverse heat conduction in thin plates. Their approach transforms the inverse problem into a direct formulation that only requires reconstructing Dirichlet boundary conditions, thereby facilitating the computation of approximate solutions [8]. Furthermore, Wang et al. proposed the Fast Bayesian Parallel Sampling (FBPS) framework, which significantly improves computational efficiency compared to traditional Bayesian methods [9]. To efficiently handle complex geometric structures and characterize subsurface anomalies, a new computational framework was introduced that significantly improves both precision and efficiency [10]. Traditionally, these inverse problems have been solved using mesh-based methods—such as FEM or Finite Difference schemes—which frequently necessitate the use of iterative regularization techniques. Nevertheless, these techniques have notable limitations. They rely heavily on mesh quality, which constitutes a major bottleneck for solving problems in complex 3D domains. By contrast, statistical methods such as Bayesian inference [9] excel at quantifying uncertainty. However, these methods often struggle with multi-dimensional parameter spaces and necessitate exhaustive sampling techniques like Markov Chain Monte Carlo. Consequently, the resulting computational burden is often prohibitive for real-time applications. Here, the PINNs framework represents a fundamental departure from traditional methods. It recasts the inverse problem as a joint optimization task where parameters and solutions are solved simultaneously, thereby eliminating the reliance on mesh generation and iterative forward solvers.

The PINNs framework distinguishes itself from conventional approaches by eliminating the need for mesh generation. This mesh-free nature facilitates robust solutions for intricate geometries, while also offering superior adaptability and generalization potential [11–14]. As neural network architectures advanced, researchers, including Lee and Kang [15], Wang and Mendel [16], and Yentis and Zaghloul [17], were among the first to apply them to PDEs. These architectures have since been refined by treating the network as a parametric function approximator and minimizing a residual-based loss function to find numerical solutions [18]. Continuous methodological improvements have spurred the development of numerous PINNs variants tailored for specific needs. Notable examples include fractional-order PINNs (fPINN) [19], conservative PINNs (cPINN) [20], and graph-based frameworks (gPINN) [21]. The uniform convergence of PINNs generated numerical approximations to their analytical counterparts has been rigorously demonstrated through asymptotic analysis [22].

Heat conduction constitutes a critical problem in practical engineering, with typical scenarios involving the inversion of unknown boundary temperature distributions from observational data of thermosensitive plates under moving heat sources [23]. Consequently, it has become a core research domain across scientific investigation and industrial applications [24]. Taking geothermal resource exploitation as an example, the demand for accurate analysis and prediction of subsurface and boundary temperature distributions gives rise to the side boundary value inverse problem for the heat conduction equation, the primary research focus of this study. For the formulation of its mathematical model, the local region of the target geothermal formation is rationally simplified to a cuboid with homogeneous thermal physical properties in the present work.

2. Mathematical model

The governing equation for 3D steady-state heat conduction can be expressed as

$$\begin{aligned} &\frac{\partial}{\partial x}\left(k\frac{\partial u(x,y,z)}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial u(x,y,z)}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial u(x,y,z)}{\partial z}\right) \\ &=F(x,y,z),(x,y,z)\in\Omega=(0,a)\times(0,b)\times(0,c), \end{aligned} \tag{1}$$

where k denotes the temperature-dependent thermal conductivity. While this general form accommodates spatially varying conductivity, the equation simplifies under the assumption of a constant thermal conductivity to

$$\Delta u=f(x,y,z),(x,y,z)\in\Omega=(0,a)\times(0,b)\times(0,c), \tag{2}$$

where $f(x,y,z)=\frac{F(x,y,z)}{k}$.

As shown in **Figure 1**, this study considers a three-dimensional cubic domain for heat conduction.

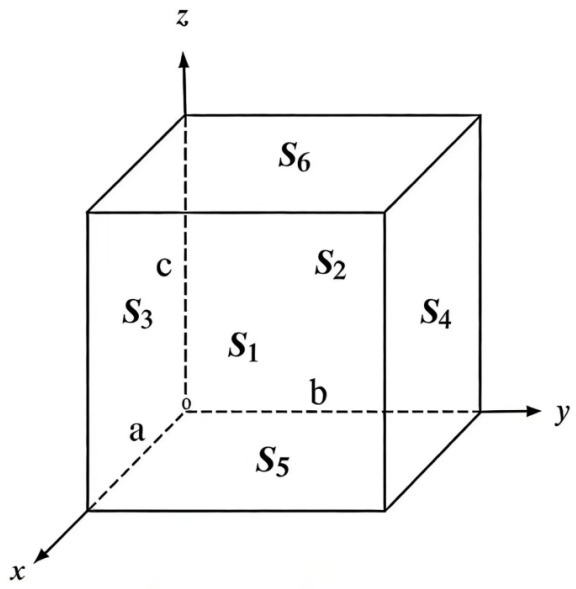


Figure 1. Schematic illustration of the three-dimensional cubic domain for heat conduction.

$$\begin{cases} \Delta u(x,y,z) = f(x,y,z), (x,y,z) \in \Omega, \\ u|_{S_1} = u(0,y,z) = g_1(y,z),\ y \in [0,b],\ z \in [0,c], \\ u|_{S_2} = u(1,y,z) = g_2(y,z),\ y \in [0,b],\ z \in [0,c], \\ u|_{S_3} = u(x,0,z) = g_3(x,z),\ x \in [0,a],\ z \in [0,c], \\ u|_{S_4} = u(x,1,z) = g_4(x,z),\ x \in [0,a],\ z \in [0,c], \\ u|_{S_5} = u(x,y,0) = g_5(x,y),\ x \in [0,a],\ y \in [0,b]. \end{cases} \tag{3}$$

The boundary of the domain Ω is defined as $\partial\Omega = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6$ with the condition that $S_1 \cap S_2 \cap S_3 \cap S_4 \cap S_5 \cap S_6 = \varnothing$. The constants *a*, *b* and *c*, denote the edge lengths of the cuboid domain. The function $u(x,\ y,\ z)$ represents the temperature distribution within the domain Ω, where $g_1(y,\ z), g_2(y,\ z), g_3(x,\ z), g_4(x,\ z)$ and $g_5(x,\ y)$ denote the temperature function on a portion of the cube's boundary surface. It is necessary to obtain the temperature function $h(x,\ y)$ on the upper surface S_6. This problem is formulated as a sideways problem. Additional constraints must be incorporated to solve the corresponding inverse problem, and two distinct types of supplementary conditions are discussed separately below.

(i) Neumann boundary conditions are additionally imposed.

The Neumann boundary conditions imposed on S_5 are given by the following formula,

$$\frac{\partial u}{\partial n}|_{S_5} = q_5. \tag{4}$$

Following a methodology widely used to nonsmooth optimal control in complex systems [25], the inverse problem formulated in Equations (3) and (4) is recast as an optimal control problem [26]. The formulation of the optimization problem is given as follows,

$$\begin{cases} \min J(u(w,b);w,b) \\ J(u) = \|\Delta u - f\|^2_{L_2(\Omega)} + \sum_{i=1}^{5} \|u - g_i\|^2_{L_2(S_i)} + \left\| \frac{\partial u}{\partial n} - q_5 \right\|^2_{L_2(S_5)}, \\ u = NET(x,y,z;w,b), \end{cases} \tag{5}$$

where, *NET*: $\Omega \to R$ represents the mapping of the network, *w* and *b* are the parameters in the neural network.

(ii) Temperature measurements are incorporated in an internal region D.

The expression for the additional temperature measurements in the internal domain is given as follows,

$$u(x,y,z)|_D = u_D, D \subseteq \Omega. \tag{6}$$

Correspondingly, the solution to the inverse problems (3) and (6) is equivalent to

the subsequent optimal control problem.

$$\begin{cases} \min\limits_{w,b} \ J(u(w,b);w,b) \\ J(u) = \|\Delta u - f\|^2_{L_2(\Omega)} + \sum_{i=1}^{5} \|u - g_i\|^2_{L_2(S_i)} + \|u - u_D\|^2_{L_2(D)}, \\ u = NET(x,y,z;w,b) \end{cases} \tag{7}$$

where u_D denotes the measurement data within the interior domain. By addressing two distinct optimization tasks, the temperature value associated with the unknown boundary condition can be derived. Specifically, the approximate function $\hat{u}(x,y,z;\theta)$ is obtained via the PINNs is utilized to approximate the target unknown function $u(x,y,z)$, with θ standing for the network's parameter set. Guided by the constructed loss function, the optimization procedure is designed to minimize in the function $J(u)$. The L-BFGS algorithm is adopted to iteratively update the network parameters θ, enabling $\hat{u}(x,y,z;\theta)$ to eventually function as an approximate solution to the inverse problem. The overall workflow of PINN-based PDE solving is presented in **Figure 2**.

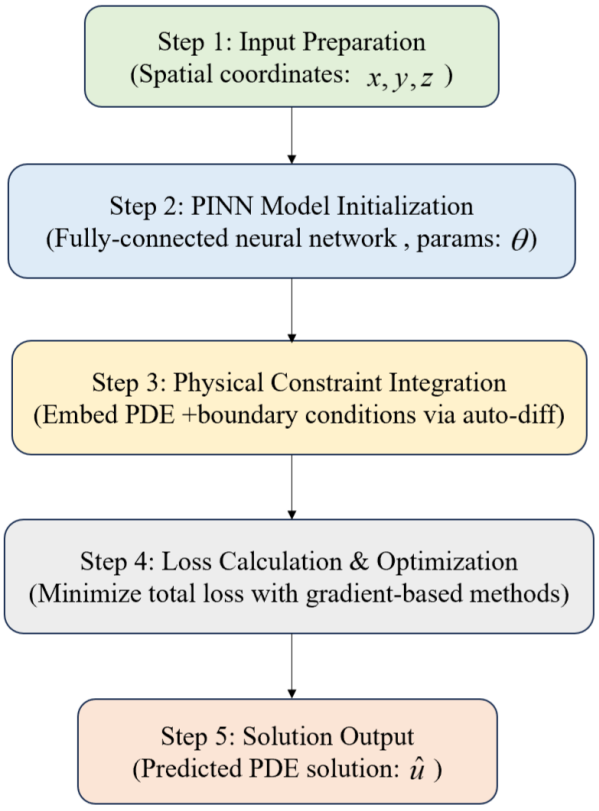


Figure 2. Workflow of PDE solving via PINNs.

3. PINNs for 3D steady-state heat conduction inverse problems

3.1. Additional Neumann boundary conditions on S_5

First, establish a deep neural network, where PDE information is integrated as constraint conditions, and computations are carried out via the network's loss function. Here, a fully connected feedforward network—one of the most prevalent neural network architectures—is employed. The choice of activation function is not a

mere heuristic selection; instead, it is inherently bounded by the differential order of the governing PDE.

The functional in Equation (5) is expressed as the following loss function

$$\mathcal{L}(\theta)_{S_5} = \omega_1 \mathcal{L}_{u_b}(\theta) + \omega_2 \mathcal{L}_f(\theta) + \omega_3 \mathcal{L}_{data}^{S_5}(\theta), \tag{8}$$

where $\mathcal{L}_f(\theta)$ is the residual of the partial differential equation,

$$\mathcal{L}_f(\theta) = \frac{1}{N_f} \left\| \Delta \hat{u}(x, y, z; \theta) - f \right\|_{L^2(\Omega)}^2, \tag{9}$$

$\mathcal{L}_{u_b}(\theta)$ is the residual of the boundary condition,

$$\mathcal{L}_{u_b}(\theta) = \frac{1}{N_b} \sum\nolimits_{i=1}^{5} \left\| \hat{u}(x, y, z; \theta) - g_i \right\|_{L^2(S_i)}^2, \tag{10}$$

$\mathcal{L}_{data}^{S_5}(\theta)$ is the residual of the observation data,

$$\mathcal{L}_{data}^{S_5}(\theta) = \frac{1}{N_{data}} \left\| \frac{\partial \hat{u}(x, y, z; \theta)}{\partial n} - q_5 \right\|_{L^2(S_5)}^2, \tag{11}$$

where $q_5 = \frac{\partial u(x,y,0)}{\partial z}$, N_f denotes the numbers of collocation point data, N_b denotes the numbers of boundary nodes, N_{data} denotes the numbers of observation observational data sample. L^2 denote the Hilbert space of square-integrable functions. The partial derivatives are computed via automatic differentiation, ω_i are the weights and $\omega_i > 0$. The training set for the loss function is partitioned into three distinct components, boundary condition sampling points set, collocation points set for evaluating PDE residuals, and observational condition sampling points set.

In typical computer vision or natural language processing (NLP) tasks, the Rectified Linear Unit (ReLU) is frequently favored, as it helps alleviate the vanishing gradient issue. However, for Physics-Informed Neural Networks aimed at solving second-order PDE like the steady-state heat equation, ReLU is theoretically sub-optimal. Mathematically, the ReLU function is piecewise linear, meaning its second-order derivative is zero everywhere. This leads to a vanishing second derivative' issue, where the physics-informed loss term,which relies on the Laplacian operator Δu, cannot effectively back-propagate physical constraints to update the network weights. By contrast, the Hyperbolic Tangent (Tanh) function constitutes a smooth and C^∞ continuous function. Its non-zero, continuous higher-order derivatives allow the neural network to accurately approximate the curvature and smooth variations inherent in harmonic-like temperature distributions. Moreover, in contrast to the Sigmoid function, the Tanh function is zero-centered, boasting an output interval of $[-1,1]$. This zero-centering property helps prevent the 'zig-zagging' dynamics of gradient updates often associated with non-zero mean activation, thereby significantly accelerating the convergence of the L-BFGS optimizer during the fine-tuning phase. Therefore, Tanh provides the necessary mathematical foundation for modeling the smooth diffusion process required by the heat conduction equation. The PINNs network structure is shown in the following **Figure 3**.

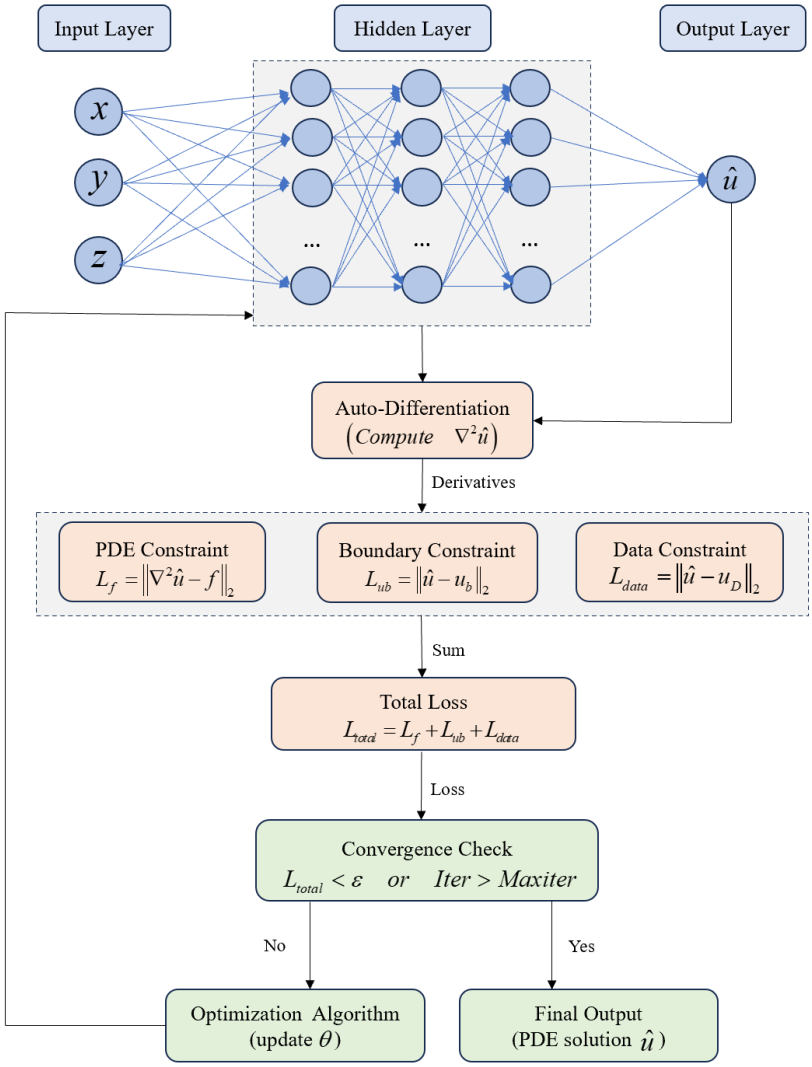


Figure 3. Schematic of the PINNs fully connected neural network.

Optimization algorithms (such as the Adam algorithm and L-BFGS algorithm) are employed to numerically minimize the loss function through iterative updates, resulting in an approximate solution to the problem.

Algorithm 1 formally presents the neural network algorithm for solving the inverse boundary problem of the 3D steady-state heat conduction equation.

Algorithm 1 PINNs algorithm for solving sideways inverse problems

Step 1: Select numbers N_b, N_f and N_{data}, for different training point set, where N_b denotes the number of boundary sampling points, N_f denotes the number of equation collocation points, and N_{data} denotes the number of observation sampling points.

Step 2: Construct a network structure with one neuron in the output layer.

Step 3: Initialize the parameters θ, which include the weights w and biases b of the neural network.

Step 4: Use automatic differentiation techniques to compute the physical constraints $\mathcal{L}_f(\theta)$ and boundary residuals $\mathcal{L}_{u_b}(\theta)$.

Step 5: Combine with observation residuals $\mathcal{L}_{data}^{S_5}(\theta)$ to form the loss function $\mathcal{L}(\theta)_{S_5}$.

Step 6: Use an optimization algorithm to update the neural network parameters θ to minimize the loss function $\mathcal{L}(\theta)_{S_5}$.

Step 7: The training process is terminated to obtain the final approximate solution $\hat{u}(x, y, z; \theta)$ once the loss function reaches a certain level of accuracy or the maximum number of training steps is attained.

3.2. Additional observational data on the interior domain D

The loss function is formulated by integrating the residuals derived from the governing equations, boundary conditions, and observational constraints, with its formal expression given as follows:

$$\mathcal{L}(\theta)_D = \omega_1 \mathcal{L}_{u_b}(\theta) + \omega_2 \mathcal{L}_f(\theta) + \omega_3 \mathcal{L}_{data}^D(\theta). \tag{12}$$

Here, $\mathcal{L}_{data}^D(\theta)$ represents the residual of the observation data, $\mathcal{L}_f(\theta)$ represents the residual of the PDE, $\mathcal{L}_{u_b}(\theta)$ represents the residual of the boundary condition, and

$$\begin{aligned} \mathcal{L}_{data}^D(\theta) &= \frac{1}{N_{data}} \left\| \hat{u}(x,y,z;\theta) - u_D \right\|_{L^2(D)}^2, \\ \mathcal{L}_f(\theta) &= \frac{1}{N_f} \left\| \Delta \hat{u}(x,y,z;\theta) - f \right\|_{L^2(\Omega)}^2, \\ \mathcal{L}_{u_b}(\theta) &= \frac{1}{N_b} \sum\nolimits_{i=1}^{5} \left\| \hat{u}(x,y,z;\theta) - g_i \right\|_{L^2(S_i)}^2. \end{aligned} \tag{13}$$

The symbols $N_f,\ N_b$ and N_{data} respectively denote the numbers of collocation point dataset, boundary nodes, and observational data sample set. $L^2(D)$ represents the Hilbert space of square-integrable functions defined on the measurable set D.

The neural network algorithm for solving the inverse boundary value problem, where observational data are prescribed on the interior domain D, similar to **Algorithm 1**. It suffices to replace the observation $\mathcal{L}_{data}^{S_5}(\theta)$ with $\mathcal{L}_{data}^D(\theta)$ and the loss function $\mathcal{L}(\theta)_{S_5}$ with $\mathcal{L}(\theta)_D$ in **Algorithm 1**.

4. Numerical example

Prior to applying this method to complex geological scenarios, it is imperative to validate its accuracy and noise robustness against well-established analytical solutions. To this end, this study primarily employs synthetic datasets to conduct a rigorous quantitative assessment.

Consider the inverse problem of the 3D steady-state heat conduction equation in the cubic domain $\Omega = (0,1) \times (0,1) \times (0,1)$. Given the temperatures on the five surfaces $S_1,\ S_2,\ S_3,\ S_4$ and S_5, along with additional observational conditions, temperature values across the whole domain Ω and on the surface S_6 is inferred.

Consider the following problem,

$$\begin{cases} \Delta u(x,y,z) = 2xy + 2yz + 2xz,\ (x,y,z) \in \Omega, \\ u|_{S_1} = u(0,y,z) = z + 5,\ y \in [0,1],\ z \in [0,1], \\ u|_{S_2} = u(1,y,z) = yz + y^2 z + yz^2 + z + 6,\ y \in [0,1],\ z \in [0,1], \\ u|_{S_3} = u(x,0,z) = x + z + 5,\ x \in [0,1],\ z \in [0,1], \\ u|_{S_4} = u(x,1,z) = x^2 z + xz + xz^2 + x + z + 5,\ x \in [0,1],\ z \in [0,1], \\ u|_{S_5} = u(x,y,0) = x + 5,\ x \in [0,1],\ y \in [0,1]. \end{cases} \tag{14}$$

The exact solution of the problem is as follows,

$$u(x, y, z) = x^2yz + xy^2z + xyz^2 + x + z + 5. \tag{15}$$

The expression for the simulation conducted using observational data with added random errors is given as follows,

$$u^\delta = u \times (1 + \delta \times r). \tag{16}$$

Here, δ represents the noise intensity, and $r \in [-1, 1]$ is a random number.

Numerical calculation accuracy is quantified via the relative error, which is defined mathematically as follows,

$$E_r(u) = \frac{\|u - \hat{u}\|_2}{\|u\|_2}. \tag{17}$$

The computations were performed on a laptop equipped with an AMD Ryzen 7 6000 Series CPU and an NVIDIA GeForce RTX GPU. The code uses the [TensorFlow/PyTorch] framework. The computation time for the numerical experiments ranges from 200 to 700 s.

4.1. Numerical example for additional Neumann boundary conditions on S_5

Given the additional condition as $u_z(x, y, 0) = x^2y + xy^2 + 1$, identify the temperature within the domain Ω and on the surface S_6. The boundary element method was used to solve the Cauchy inverse problem in a rectangular domain. For this method, the relative error level of $\hat{u}$ is 10^{-2}, and the relative error level of the inversion approximation on the boundary is 10^{-1} [27].

In the present study, the neural network was subjected to validation using architectures with 2, 3, 4, and 6 hidden layers respectively. Each hidden layer was configured with 40 neurons, while the activation functions incorporated in the experiments included the ReLU, Logistic, and Tanh functions. The symbols e_u (characterizing the global temperature approximation accuracy) and e_{S_6} (quantifying the top surface temperature approximation performance) derived from these different structural configurations are tabulated in **Table 1**.

It can therefore be deduced that while the fitting performance exhibits a certain degree of improvement with an increasing number of hidden layers, such enhancement will not persist indefinitely as the layer count continues to rise. As indicated in **Table 1**, the ReLU activation function yields inferior performance compared to the Logistic function and Tanh function for this inverse sideways boundary problem. Although the accuracy of the approximate solutions obtained using the Logistic activation function and the Tanh activation function is comparable, experimental observations show that the network with the Tanh activation function achieves faster training convergence and a shorter experimental runtime in practical experiments. Considering various factors comprehensively, the Tanh function is selected as the activation function in this numerical test.

Table 1. Experimental results of different activation functions under varying numbers of layers.

Number of hidden layers	Activation function	e_u	e_{S_6}
2	ReLU	6.082×10^{-3}	7.485×10^{-3}
2	Logistic	2.541×10^{-4}	3.982×10^{-3}
2	Tanh	1.326×10^{-4}	2.203×10^{-4}
3	ReLU	5.918×10^{-3}	9.276×10^{-3}
3	Logistic	2.717×10^{-3}	4.231×10^{-4}
3	Tanh	1.166×10^{-4}	1.537×10^{-4}
4	ReLU	4.821×10^{-3}	8.778×10^{-3}
4	Logistic	2.758×10^{-3}	3.582×10^{-4}
4	Tanh	6.851×10^{-4}	1.198×10^{-3}
6	ReLU	6.055×10^{-3}	1.015×10^{-2}
6	Logistic	2.724×10^{-3}	4.932×10^{-3}
6	Tanh	5.164×10^{-3}	1.041×10^{-3}

We proceed to investigate the influence of varying network layer numbers and neuron counts on the performance of PINNs. The approximation errors derived from different network configurations—specifically, 2-layer, 3-layer and 4-layer networks paired with 10, 20, 40, 50 and 60 neurons respectively—are tabulated in **Table 2**.

Table 2. Prediction error e_u of PINNs with different numbers of network layers and neurons.

Number of neurons/number of network layers	10	20	40	50	60
2	2.742×10^{-3}	1.776×10^{-4}	1.326×10^{-4}	1.301×10^{-4}	1.251×10^{-4}
3	2.237×10^{-3}	1.289×10^{-4}	1.166×10^{-4}	1.062×10^{-4}	1.191×10^{-4}
4	2.018×10^{-3}	1.153×10^{-4}	6.851×10^{-4}	1.061×10^{-4}	1.118×10^{-4}

As the number of neurons and network layers increases, the error e_u decreases. When the number of neurons is 40 or 50, and the number of network layers is 3 or 4, it is observed that the impact on approximation accuracy is negligible. As can be seen from the table, both the network configurations with 3 layers with 50 neurons, 4 layers with 50 neurons yield favorable results. However, the experimental runtime of the former is 605 s, while that of the latter is 3373 s. By comprehensively balancing the trade-off between accuracy and computational time cost, the network architecture with 3 layers and 50 neurons is adopted for the numerical examples in the present case.

To begin with, a neural network is established, which consists of an input layers, output layers, and three hidden layers with 50 neurons allocated to each layer, while the Tanh function is designated as the activation function. A total of 2600 training data points are selected for the experiment: $5N_b = 400 \times 5$, $N_f = 200$, and $N_{data} = 400$. The loss function is given by Equation (8), where the boundary condition residual $\mathcal{L}_{u_b}(\theta)$, the PDE residual $\mathcal{L}_f(\theta)$, and the observation data residual $\mathcal{L}_{data}^{S_5}(\theta)$ as follows,

$$\begin{aligned}\mathcal{L}_{u_b}(\theta) = &\tfrac{1}{N_b}\textstyle\sum_{i=1}^{N_b} |\hat{u}(0, y_i, z_i; \theta) - u(0, y_i, z_i)|^2 + \tfrac{1}{N_b}\sum_{i=1}^{N_b} |\hat{u}(1, y_i, z_i; \theta) - u(1, y_i, z_i)|^2 \\ &+\tfrac{1}{N_b}\textstyle\sum_{i=1}^{N_b} |\hat{u}(x_i, 0, z_i; \theta) - u(x_i, 0, z_i)|^2 + \tfrac{1}{N_b}\sum_{i=1}^{N_b} |\hat{u}(x_i, 1, z_i; \theta) - u(x_i, 1, z_i)|^2 \\ &+\tfrac{1}{N_b}\textstyle\sum_{i=1}^{N_b} |\hat{u}(x_i, y_i, 0; \theta) - u(x_i, y_i, 0)|^2,\end{aligned} \tag{18}$$

$$\mathcal{L}_f(\theta) = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|\frac{\partial^2\hat{u}(x_i,y_i,z_i;\theta)}{\partial x^2} + \frac{\partial^2\hat{u}(x_i,y_i,z_i;\theta)}{\partial y^2} + \frac{\partial^2\hat{u}(x_i,y_i,z_i;\theta)}{\partial z^2} - f(x_i,y_i,z_i)\right|^2 \tag{19}$$

$$\mathcal{L}_{data}^{S_5}(\theta) = \frac{1}{N_{data}}\sum_{i=1}^{N_{data}}\left|\frac{\partial\hat{u}(x_i,y_i,0;\theta)}{\partial z} - \frac{\partial u\,(x_i,y_i,0)}{\partial z}\right|^2. \tag{20}$$

The temperature distribution within the domain Ω can be predicted using **Algorithm 1**. In this study, numerical experiments are conducted using synthetically generated observational data contaminated with noise of varying intensity levels, yielding approximate solutions corresponding to different noise levels and quantifying the deviation between these approximate solutions and the analytical solution. The approximate solutions under distinct noise levels are presented in **Figure 4**.

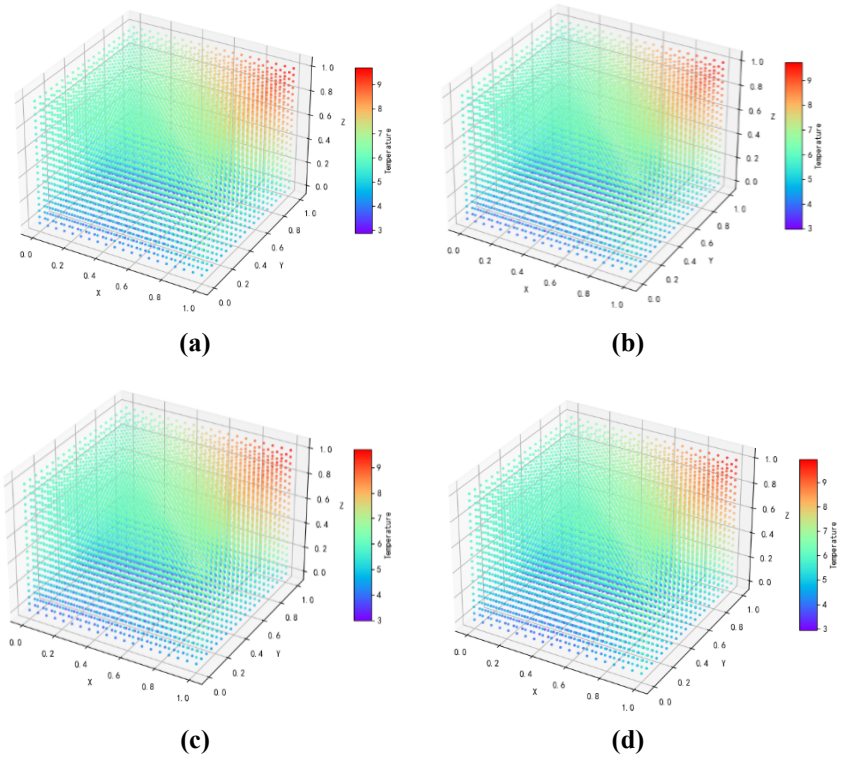
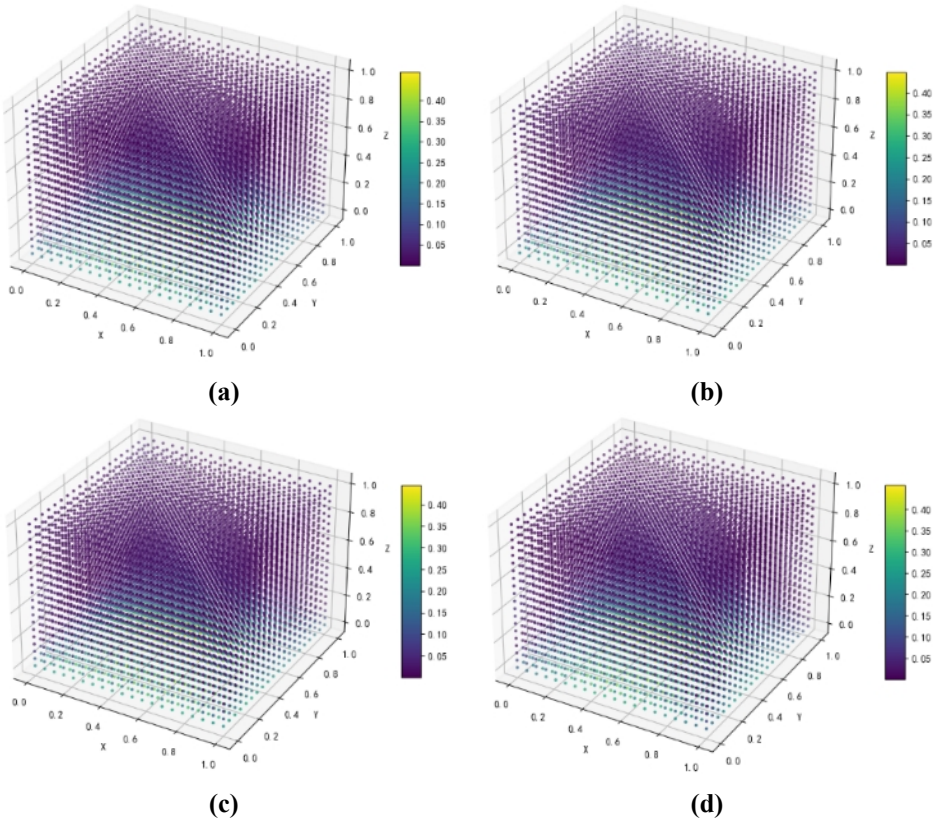


Figure 4. Approximate solutions $\hat{u}$ under varying noise levels: **(a)** Noiseless, **(b)** 1% Noise, **(c)** 5% Noise, **(d)** 10% Noise.

Figure 4 shows that as noise intensity rises, the solution surfaces gradually shift from nearly error-free profiles to localized deviations. Notably, the global spatial gradients and spatial distribution characteristics of high-temperature regions are accurately reconstructed across all noise intensity levels, illustrating the model's robust noise resistance in preserving the key thermophysical features of the steady-state heat conduction problem.

Figure 5 shows the error distribution maps comparing approximate solutions (reconstructed from noisy boundary observational data with varying amplitudes) to the exact solution, with each subfigure for a distinct noise amplitude.



(a) **(b)** **(c)** **(d)**

Figure 5. Comparison of errors between the exact solution u and the approximate solution $\hat{u}$ under varying noise levels: **(a)** Noiseless, **(b)** 1% Noise, **(c)** 5% Noise, **(d)** 10% Noise.

The PINNs enables the simultaneous reconstruction of both the approximate solution $\hat{u}$ and the temperature distribution on the boundary S_6. **Table 3** summarizes the relative L^2-norm errors, along with the relative errors of the temperature values on S_6, across a range of noise amplitudes.

Table 3. Relative errors of the approximate solution for the inverse boundary identification problem.

Different noise levels	Relative error of $\hat{u}$	Relative error of boundary temperature values
Noiseless	1.982593×10^{-4}	4.895225×10^{-4}
1% Noise	1.027173×10^{-4}	2.418572×10^{-4}
5% Noise	1.035969×10^{-3}	1.935097×10^{-3}
10% Noise	1.342753×10^{-3}	2.484777×10^{-3}

As evidenced by **Table 3**, the errors exhibit a bounded marginal increment under noise-contaminated conditions, demonstrating the model's inherent robustness against error propagation mechanisms.

To validate the capability of the proposed method in handling complex physical fields under source-free conditions, we introduce an analytical solution that satisfies the harmonic condition. This solution is constructed by coupling trigonometric and exponential functions, defined as

$$u(x, y, z) = sin(\pi x)cos(\pi y)e^{\sqrt{2}\pi z}. \tag{21}$$

Experimental Setup: The network architecture follows the aforementioned

numerical example, 3 hidden layers with 50 neurons, activation function is Tanh.

The convergence behavior of the loss function value during PINNs training is a key indicator of the model's stability and learning effectiveness. **Figure 6** illustrates the change in loss function value over 7000 training iterations, reflecting the progressive optimization of the proposed framework.

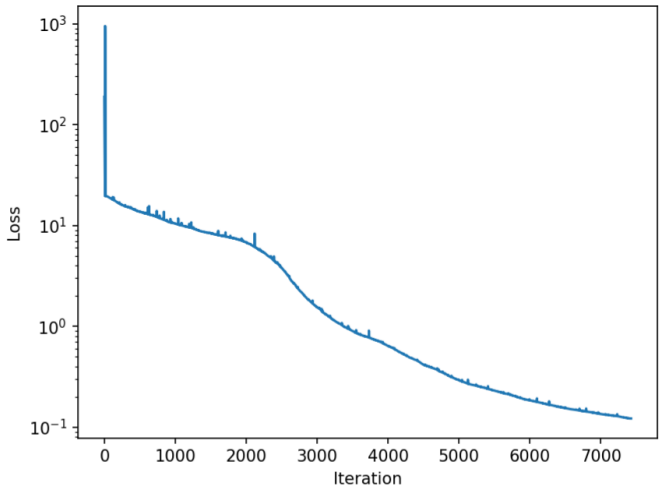


Figure 6. Convergence curve of the PINNs loss function.

From **Figure 6**, the loss value falls sharply in the first 1000 iterations, then declines steadily over 7000 iterations to a near-optimal level. This smooth convergence confirms the training process is numerically stable and demonstrates the physics-informed loss function effectively drives the model to optimal convergence.

As shown in **Figure 7**, the predicted temperature distribution is in good overall agreement with the exact distribution, both exhibiting a spatial pattern of high temperature at the bottom center and gradual cooling toward the surroundings. This indicates that the model can effectively capture the core characteristics of the temperature field.

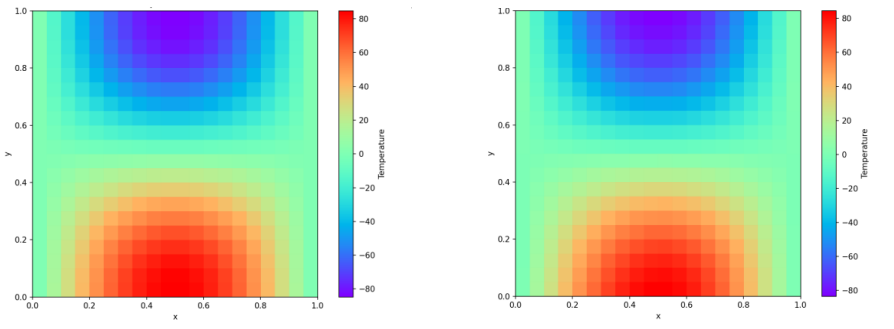
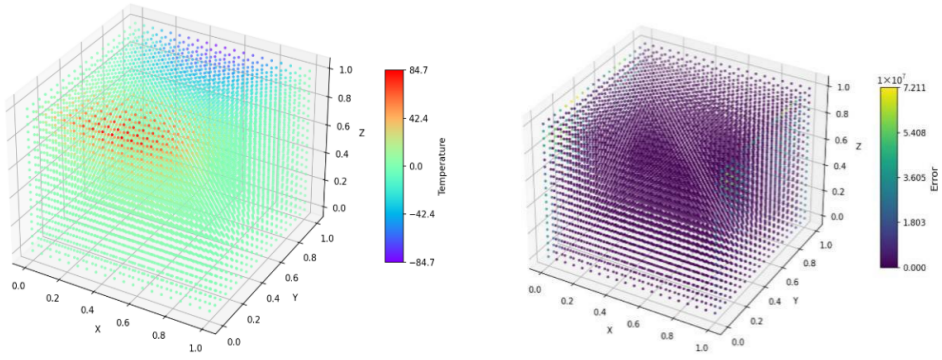


Figure 7. Comparison of **(a)** exact temperature and **(b)** predicted temperature distributions on S_6.

From **Figure 8**, it can be seen that high temperatures concentrate in the central region and gradually decrease outward, consistent with the expected spatial thermal pattern; the relative errors are generally at a low level, with only minor fluctuations in local areas. This indicates that the prediction model can accurately capture the 3D temperature field, and the results are highly reliable.



(a) **(b)**

Figure 8. **(a)** Predicted temperature distribution; **(b)** Relative errors.

The **Table 4** presents the relative errors obtained using internal observational data contaminated with different noise levels.

Table 4. Relative errors under different noise levels.

Different noise levels	Minimum relative error of $\hat{u}$	Relative error of boundary temperature values
Noiseless	6.625358×10^{-2}	7.601113×10^{-2}
1% Noise	4.362917×10^{-2}	5.168691×10^{-2}
5% Noise	4.313745×10^{-2}	5.123986×10^{-2}
10% Noise	6.243320×10^{-2}	7.366219×10^{-2}

The results demonstrate that the proposed PINNs framework exhibits robust noise resilience: errors are controllable and even reduced under low noise levels (1%–5%), and remain below the noise-free level even with 10% high noise intensity.

To further quantify this performance, the relative errors for different network depths are summarized in **Table 5**.

Table 5. Relative errors with varying network depths.

Number of network layers	Minimum relative error of $\hat{u}$	Relative error of boundary temperature values
2	7.116510×10^{-2}	8.129906×10^{-2}
3	4.313745×10^{-2}	5.123986×10^{-2}
4	6.255944×10^{-2}	7.362182×10^{-2}
5	5.027186×10^{-2}	5.964556×10^{-2}

The results indicate that a 3-layer network yields the lowest reconstruction errors, suggesting an optimal network depth for the proposed PINNs framework, while deeper networks (4 or 5 layers) lead to increased errors due to potential overfitting.

4.2. Numerical example for additional observational data on the interior domain D

Given the observation temperature as $u\left(\frac{1}{2}, y, z\right) = \frac{1}{4}yz + \frac{1}{2}y^2z + \frac{1}{2}yz^2 + z + \frac{11}{2}$, identify the unknown temperature within the domain Ω and on the surface S_6. The expression of the analytical solution to the problem (14) is given as follows,

$$u(x, y, z) = x^2yz + xy^2z + xyz^2 + x + z + 5. \tag{22}$$

The expression of the $h(x,y)$ is given by

$$h(x,y) = x^2y + xy^2 + xy + x + 6. \tag{23}$$

First, establish a neural network with inputs, outputs, and six hidden layers, each containing 80 neurons, and use Tanh as activation function. We select training data set consisting of 2600 points for the PINNs model training, $5N_b = 400 \times 5$, $N_f = 200$, and $N_{data} = 400$. The loss function is given by Equation (12), where residual $\mathcal{L}_{u_b}(\theta)$, the residual $\mathcal{L}_f(\theta)$, and the residual $\mathcal{L}^D_{data}(\theta)$ as follows,

$$\begin{aligned}\mathcal{L}_{u_b}(\theta) = &\tfrac{1}{N_b}\textstyle\sum_{i=1}^{N_b}\left|\hat{u}(0,y_i,z_i;\theta) - u(0,y_i,z_i)\right|^2 + \tfrac{1}{N_b}\textstyle\sum_{i=1}^{N_b}\left|\hat{u}(1,y_i,z_i;\theta) - u(1,y_i,z_i)\right|^2 \\ &+\tfrac{1}{N_b}\textstyle\sum_{i=1}^{N_b}\left|\hat{u}(x_i,0,z_i;\theta) - u(x_i,0,z_i)\right|^2 + \tfrac{1}{N_b}\textstyle\sum_{i=1}^{N_b}\left|\hat{u}(x_i,1,z_i;\theta) - u(x_i,1,z_i)\right|^2 \\ &+\tfrac{1}{N_b}\textstyle\sum_{i=1}^{N_b}\left|\hat{u}(x_i,y_i,0;\theta) - u(x_i,y_i,0)\right|^2,\end{aligned} \tag{24}$$

$$\mathcal{L}_f(\theta) = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|\left(\frac{\partial^2\hat{u}(x_i,y_i,z_i;\theta)}{\partial x^2} + \frac{\partial^2\hat{u}(x_i,y_i,z_i;\theta)}{\partial y^2} + \frac{\partial^2\hat{u}(x_i,y_i,z_i;\theta)}{\partial z^2}\right) - f(x_i,y_i,z_i)\right|^2, \tag{25}$$

$$\mathcal{L}^D_{data}(\theta) = \frac{1}{N_{data}}\sum_{i=1}^{N_{data}}\left|\hat{u}\left(\frac{1}{2},y_i,z_i;\theta\right) - u\left(\frac{1}{2},y_i,z_i\right)\right|^2, \tag{26}$$

where (x_i, y_i, z_i) denotes the coordinates of the internal nodes. The loss function weights are selected as $\omega_1 = 9$, $\omega_2 = 1$, $and\ \omega_3 = 1$ to balance the data fidelity term, physical residual term, and boundary constraints, respectively. The network is trained with the L-BFGS algorithm to minimize the composite loss function, producing a model that optimally approximates the solution under given physical constraints. The trained network then computes the unknown boundary conditions and full-field solution.

To validate the noise robustness of our method, we introduce training conditions for different noise levels into the training process, along with initial network parameters and physical constraints. Loss function value variations are shown in **Figure 9**.

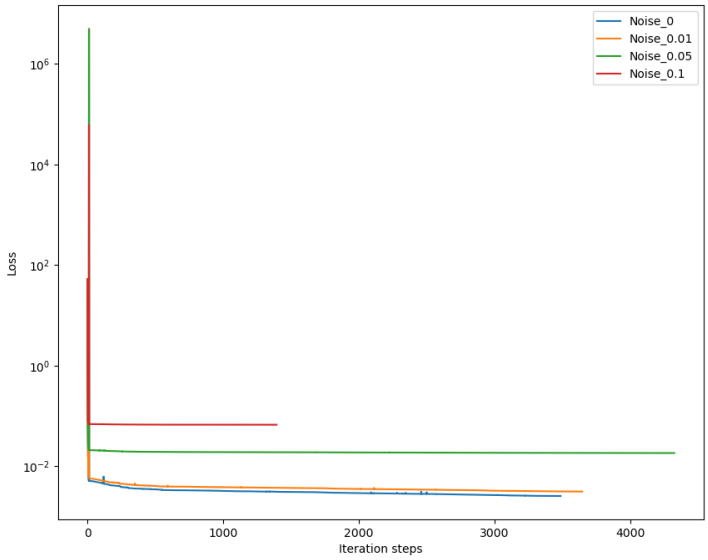


Figure 9. Convergence curve of the PINNs loss function.

Numerical experiments under multi-level noise conditions show that the loss

function converges to a specific accuracy after 1000 iterations, and this accuracy is correlated with the noise intensity.

Figure 10 illustrates the comparative results of the approximate solutions derived within the cubic domain by employing internal partial temperature observation data under varying noise levels.

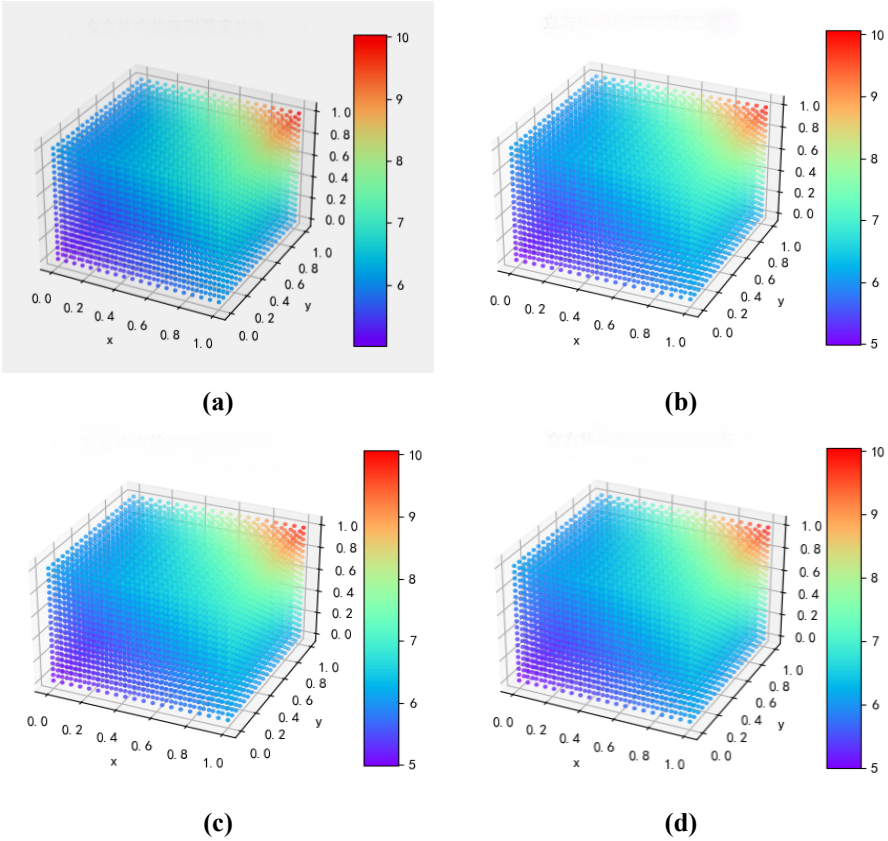


Figure 10. The approximate solutions $\hat{u}$ under varying noise levels, **(a)** Noiseless; **(b)** 1% noise; **(c)** 5% noise; **(d)** 10% noise.

Figure 10 shows that although the approximate solutions exhibit localized oscillatory perturbations with increasing noise intensity, the spatial distribution pattern of the global thermal field remains unchanged, which verifies the robustness of the proposed method against noise intensity.

As shown in **Figure 11**, errors are concentrated primarily in high-gradient interior regions, with their magnitude rising as noise levels increase. Critically, the global error field's topological structure stays unchanged, confirming the method's noise-immunity properties.

Table 6 indicates that the method demonstrates measurable robustness against low-level noise perturbations (1%). However, a nonlinear deterioration in predictive accuracy emerges with increasing noise intensity, accompanied by progressive degradation of solution fidelity under elevated noise conditions.

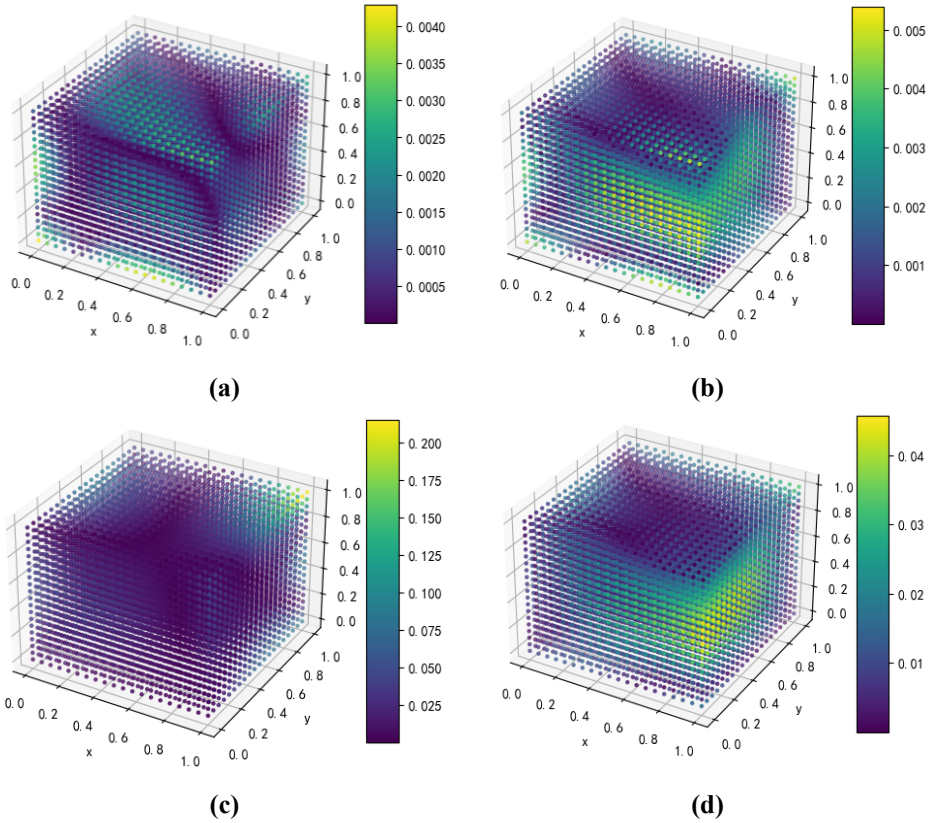


Figure 11. Comparison of errors between the exact solution u and the approximate solution $\hat{u}$ for the Cauchy problem in a cubic region under varying noise levels: **(a)** Noiseless, **(b)** 1% noise, **(c)** 5% noise, **(d)** 10% noise.

Table 6. Relative errors of the results for the boundary identification inverse problem.

Different noise levels	Minimum relative error of $\hat{u}$	Relative error of boundary temperature values
Noiseless	1.062329×10^{-4}	1.378126×10^{-4}
1% Noise	2.559945×10^{-4}	1.968565×10^{-4}
5% Noise	1.114833×10^{-3}	6.772392×10^{-4}
10% Noise	2.201698×10^{-3}	1.073202×10^{-3}

4.3. Theoretical analysis of stability and noise robustness

The numerical results in **Tables 3** and **6** demonstrate that the proposed method exhibits robust stability, where the reconstruction error grows linearly rather than exponentially with increasing noise intensity. This robustness can be theoretically justified by framing the PINNs training process as a Tikhonov-regularized optimization problem, which aligns with the core framework for investigating the generalization error of PINNs in solving PDE inverse problems [28].

In the Hadamard sense, the inverse problem of reconstructing boundary temperatures from measurement data is inherently ill-posed. The defining characteristic of Hadamard ill-posedness lies in the solution's extreme sensitivity to observational noise: even infinitesimal perturbations in the measured data can lead to arbitrarily large deviations in the reconstructed solution [6,28].

To rigorously guarantee the stability of the PINNs-based inverse heat conduction method, three core assumptions must be satisfied [28]:

(i) Bounded Observational Noise

The discrepancy between noisy measurements and exact values satisfies $\left\|u_{\text{meas}}^{\delta}-u_{\text{exact}}\right\|_{L^2(D\cup\partial\Omega)}\leq\delta$, where D denotes the internal measurement region and $\partial\Omega$ denotes the boundary, δ denotes the noise level, In numerical experiments, this assumption is validated by introducing controlled noise ($\delta=1\%,5\%,10\%$) to synthetic data, which mimics real-world measurement errors while maintaining boundedness. u_{meas}^{δ} denotes the noisy observational data and u_{exact} denotes the exact solution. This boundedness ensures that noise-induced perturbations are quantifiable and controllable.

(ii) Solution Space Constrained to Sobolev Space $H^1(\Omega)$

The imposition of Neumann boundary conditions or internal temperature measurements constrains the solution space to the Sobolev space $H^1(\Omega)$—the Neumann boundary condition is mathematically expressed as $\left\|\frac{\partial u}{\partial n}-q_S\right\|_{L^2(S_5)}^2\leq\epsilon$; internal measurements $u|_D=u_D$ similarly restrict the solution's spatial variations, as the measured values act as "anchors" to limit arbitrary fluctuations in the temperature field. This constraint guarantees bounded first derivatives of the solution, rendering the set of admissible solutions a compact function space.

(iii) PDE Residual as a Seconder-Order Tikhonov Regularizer

The PDE residual term $\|\Delta u-f\|_{L^2(\Omega)}^2$ in the PINNs loss function is minimized during training. This term acts as a second-order Tikhonov regularizer, suppressing high-frequency noise fluctuations.

To quantify the stability of the proposed method, we derive a stability estimate for the PINNs approximation error. Let $\mathcal{L}:H^1(\Omega)\to L^2(D\cup\partial\Omega)$ denote the measurement operator, such that the exact measured data is $g=\mathcal{L}(u_{\text{exact}})$. In the presence of noise, the observed data is $g^{\delta}=\mathcal{L}(u_{\text{exact}})+\eta$, where η denotes the noise term satisfying $\|\eta\|_{L^2(D\cup\partial\Omega)}\leq\delta$ [28]. The PINNs loss function can be rewritten in a unified Tikhonov-regularized form.

$$\mathcal{J}(\theta)=\|\mathcal{L}(u_\theta)-g^{\delta}\|_{L^2(D\cup\partial\Omega)}^2+\lambda\|\Delta u_\theta-f\|_{L^2(\Omega)}^2, \tag{27}$$

where θ denotes the PINNs parameter set (weights and biases), $u_\theta=\text{NET}(x,y,z;\theta)$ is the network's output, and $\lambda>0$ balances the data fidelity term and the PDE residual term [28].

The PDE residual term $||\Delta u_\theta-f||_{L^2(\Omega)}^2$ acts as an implicit regularization operator, which is analogous to the second-order Tikhonov regularizer. For the steady-state heat equation, the Laplacian operator Δ acts as a high-pass filter for spatial frequencies. Random noise η typically contains high-frequency oscillations, which would lead to unbounded growth in the Laplacian norm. This smoothing effect aligns exactly with the core philosophy of Tikhonov regularization—by penalizing the higher-order derivatives of the solution, it constrains the degree of fluctuation in the solution, thereby preventing the model from overfitting to noise in the observational data. For the steady-state heat conduction equation, this constraint directly corresponds to minimizing the residual of the physical law, ensuring that the solution satisfies both

the data fidelity and physical consistency requirements simultaneously. By forcing the neural network to minimize the PDE residual $||\Delta\hat{u} - f||$, the optimization process effectively suppresses these high-frequency noise components, ensuring the recovered solution $\hat{u}$ remains within the manifold of smooth physical solutions [28].

This constraint restricts the minimization occurs to the Sobolev space $H^1(\Omega)$, rather than the broader space $L^2(\Omega)$. According to inverse problems stability theory, confining admissible solutions to a compact function space of functions with bounded derivatives restores the continuous dependence of the solution on input data [28]. Consequently, the approximation error of the PINNs solution $\hat{u} = u_{\hat{\theta}}$ (where $\hat{\theta}$ minimizes $\mathcal{J}(\theta)$) satisfies the following stability estimate,

$$\|\hat{u} - u_{\text{exact}}\|_{L^2(\Omega)} \leq C_1\delta + C_2\left(N_{\text{int}}^{-\frac{\alpha}{2}} + N_{\text{data}}^{-\frac{\alpha_d}{2}}\right) + C_3\epsilon_{\text{approx}}, \tag{28}$$

where C_1 is the stability constant (Lipschitz constant) dependent on the domain and operator; N_{int} and N_{data} are the numbers of interior collocation points and data points, respectively; α and α_d are the convergence rates of the quadrature rules for the PDE residual and data residual; ϵ_{approx} represents the neural network approximation error. C_2 and C_3 are constants dependent on the regularity of the residuals and the network architecture [28]. This stability estimate theoretically confirms that the reconstruction error grows linearly with noise intensity δ, rather than exponentially. This explains the robust noise resistance observed in numerical experiments, where errors remain controllable even under 10% noise perturbation.

5. Conclusion

Based on the PINNs model, this study examines the steady-state heat conduction problem in a 3D cubic domain by incorporating two distinct types of observational data. The loss function integrates residuals from boundary conditions, the governing physical equation, and observational data. Collocation points are only required within the computational domain, eliminating the need for mesh-based domain discretization. Numerical results demonstrate that network hyperparameters exert a significant impact on computational outcomes, highlighting the need to select hyperparameters tailored to the complexity of the target problem. Moreover, increasing the number of hidden layers and neurons improves the accuracy of computational results, while appropriate tuning of the weight coefficients for each term in the loss function can effectively enhance the solution accuracy of PINNs for heat conduction problems. Systematic sensitivity analysis identifies the hyperbolic tangent (Tanh) function as the optimal activation function due to its superior ability to model high-order derivatives, and an optimal network configuration is determined to balance computational efficiency and solution accuracy.

Furthermore, this study establishes a theoretical foundation for the robustness of the proposed method. Theoretical analyses confirm that the physics-informed loss function acts as an implicit Tikhonov regularizer, effectively suppressing high-frequency noise components. Consequently, the reconstruction error increases steadily and linearly (rather than exponentially) under noise-contaminated conditions.

In contrast to traditional mesh-based methods that decouple forward simulation and inverse parameter estimation, the PINNs framework proposed in this study marks a fundamental paradigm shift. Traditional approaches, such as the Finite Element Method (FEM), often face significant computational burdens in mesh generation for complex 3D geometries and require expensive sensitivity analyses. In contrast, the PINNs method is inherently mesh-free, transforming the inverse heat conduction problem into a direct optimization task. By integrating physical laws directly into the loss function, the neural network effectively serves as an implicit regularizer. Both theoretical analysis and numerical experiments confirm that this approach prevents catastrophic noise amplification, ensuring a linear error growth even under noisy conditions—a distinct advantage over unregularized traditional solvers. Conceptually, compared with traditional mesh-based FEM solvers and stochastic Bayesian methods, the proposed framework features a streamlined mesh-free paradigm that obviates the need for complex iterative regularization and extensive sampling.

Notably, this study uses synthetic data to quantitatively validate the algorithm's accuracy and stability. While random noise is incorporated to simulate measurement errors, real geothermal environments involve more complex factors, such as heterogeneous media and irregular domain boundaries. Future work will apply the proposed PINNs framework to field-measured geothermal reservoir data to further verify its practical engineering value, with a focus on adapting the model to heterogeneous thermal conductivity and irregular geological boundaries.

Author contributions: Conceptualization, TL; methodology, TL; software, QZ and XR; validation, TL, QZ, and DY; formal analysis, JK; investigation, DY; resources, TL; data curation, TL; writing—original draft preparation, TL and DY; writing—review and editing, QZ, JK, and ZL; visualization, QZ; supervision, TL; project administration, TL; funding acquisition, TL and ZL. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Grant Nos. 42264007, 41962019, 12161061), the Natural Science Foundation of Inner Mongolia (Grant No. 2025MS01021), and the Fundamental Research Funds for the Inner Mongolia University of Finance and Economics (Grant No. NCYWT23036).

Institutional review board statement: Not applicable.

Informed consent statement: Not applicable.

Data availability statement: No new data were created in this study. All data supporting the findings are included within the article.

Conflict of interest: The authors declare no conflict of interest.

References

1. Yang Z, Wu X, He X, et al. A multiscale analysis-assisted two-stage reduced-order deep learning approach for effective thermal conductivity of arbitrary contrast heterogeneous materials. Engineering Applications of Artificial Intelligence. 2024; 136(Part A): 108916.

2. Ye CQ, Chung ET. Constraint energy minimizing generalized multiscale finite element method for inhomogeneous boundary value problems with high contrast coefficients. arXiv preprint. 2023; arXiv:2201.04834.
3. Yang Z, Xie J. Hölder regularity of solutions for certain degenerate parabolic integro-differential equation. Advances in Differential Equations and Control Processes. 2024; 31(3): 379–395.
4. Alsulami AS, Al-Mazmumy M, Alyami MA, et al. Application of Adomian decomposition method to a generalized fractional Riccati differential equation (Ψ-FRDE). Advances in Differential Equations and Control Processes. 2024; 31(4): 531–561.
5. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics. 2019; 378: 686–707.
6. Kabanikhin SI. Inverse and Ill-Posed Problems: Theory and Applications. De Gruyter; 2011.
7. Omurov TD, Dzhumagulov KR. Regularization of the inverse problem with the d'Alembert operator in an unbounded domain degenerating into a system of integral equations of Volterra type. Advances in Differential Equations and Control Processes. 2024; 31(4): 473–486.
8. Luo J, Yang QQ, Lu S, et al. A novel formulation and sequential solution strategy with time-space adaptive mesh refinement for efficient reconstruction of local boundary heat flux. International Journal of Heat and Mass Transfer. 2019; 141: 1288–1300.
9. Wang C, Heng Y, Luo J, et al. A fast Bayesian parallel solution framework for large-scale parameter estimation of 3D inverse heat transfer problems. International Communications in Heat and Mass Transfer. 2024; 155: 107409.
10. Higuera M, Perales JM, Rapún ML, et al. Solving inverse geometry heat conduction problems by postprocessing steady thermograms. International Journal of Heat and Mass Transfer. 2019; 143: 118490.
11. Grossmann TG, Komorowska UJ, Latz J, et al. Can physics-informed neural networks beat the finite element method? IMA Journal of Applied Mathematics. 2024; 89(1): 143–174.
12. Antonion K, Wang X, Raissi M, et al. Machine learning through Physics-Informed Neural Networks: Progress and challenges. Academic Journal of Science and Technology. 2024; 9(1): 46–49.
13. Feng X, Jiang Y, Qin JX, et al. Unsupervised learning method for the wave equation based on finite difference residual constraints loss. arXiv preprint. 2024; arXiv:2401.12489.
14. Hu H, Qi L, Chao X. Physics-informed Neural Networks (PINNs) for computational solid mechanics: Numerical frameworks and applications. Thin-Walled Structures. 2024; 205(Part B): 112495.
15. Lee H, Kang IS. Neural algorithm for solving differential equations. Journal of Computational Physics. 1990; 91(1): 110–131.
16. Wang L, Mendel JM. Structured trainable networks for matrix algebra. In: Proceedings of the 1990 International Joint Conference on Neural Networks; 17–21 June 1990; San Diego, CA, USA.
17. Yentis R, Zaghloul ME. VLSI implementation of locally connected neural network for solving partial differential equations. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. 1996; 43(8): 687–690.
18. Lagaris IE, Likas A, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks. 1998; 9(5): 987–1000.
19. Pang G, Lu L, Karniadakis GE. fPINNs: Fractional physics-informed neural networks. SIAM Journal on Scientific Computing. 2019; 41(4): A2603–A2626.
20. 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; 365: 113028.
21. Yu J, Lu L, Meng X, et al. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. Computer Methods in Applied Mechanics and Engineering. 2022; 393: 114823.
22. Shin Y, Darbon J, Karniadakis GE. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. Communications in Computational Physics. 2020; 28(5): 2042–2074.
23. Manthena VR, Srinivas VB, Lamba NK, et al. Fractional thermal response in a thermosensitive rectangular plate due to the action of a moving source of heat. Advances in Differential Equations and Control Processes. 2024; 31(3): 397–415.
24. Zhang B, Wu G, Gu Y, et al. Multi-domain physics-informed neural network for solving forward and inverse problems of steady-state heat conduction in multilayer media. Physics of Fluids. 2022; 34(11): 113601.
25. Ramazanova AT. Necessary conditions for optimality in one nonsmooth optimal control problem for

Goursat-Darboux systems. Advances in Differential Equations and Control Processes. 2024; 31(4): 673–681.
26. Li Y, Hu X. Artificial neural network approximations of Cauchy inverse problem for linear PDEs. Applied Mathematics and Computation. 2022; 414: 126678.
27. Bian BX, Zhou HL, Cheng CZ, et al. TSVD regularization method for inverse identification of two-dimensional potential boundary conditions. Journal of Hefei University of Technology (Natural Science Edition). 2014; 37(9): 1097–1101.
28. Mishra 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 Numerical Analysis. 2022; 42(2): 981–1022.