

HCP-PIGN: Efficient heat conduction prediction by physics-informed graph convolutional neural network

Jiang-Zhou Peng^a, Nadine Aubry^b, Yu-Bai Li^c, Zhi-Hua Chen^a, Mei Mei^{d,*}, Yue Hua^{d,*}

^a Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China

^b Department of Mechanical Engineering, Tufts University, Medford, MA 02155, USA

^c Key Laboratory of Ocean Energy Utilization and Energy Conservation of Ministry of Education, Dalian University of Technology, Dalian 116023, China

^d Sino-French Engineer School, Nanjing University of Science and Technology, Nanjing 210094, China



ARTICLE INFO

Keywords:

Heat conduction
Surrogate model
PINN
GCN

ABSTRACT

This work proposes a novel surrogate model (noted as HCP-PIGN) combining two groups of neural networks: i.e., the physics-informed and the graph convolutional neural networks (noted as PINN and GCN). It aims to tackle the existing challenges: pixelated pre-processing of data and large amounts of training data. For predicting 2D steady-state heat conduction, the GCN acting as the prediction module, considering the interdependence between unstructured and neighboring nodes. The PINN serving as the physical constraint module, embeds governing equations into the neural network's loss function. The HCP-PIGN model obtains precise predictions with diverse geometries and within milliseconds. The predictive performance of HCP-PIGN was further compared with three network structures: i.e., the physics-informed fully connected neural network (noted as FNN), purely data-driven based FNN, and GCN. The results indicate that HCP-PIGN has the lowest error of temperature field predictions, which are below 3 % and 1.3 % for the max and mean relative errors, respectively. The improvements of 28.1% and 34.6% in accuracy are achieved over the pure data-driven GCN, and the physics-driven FNN, respectively. Therefore, the proposed HCP-PIGN model improves the physical prior knowledge and model's adaptabilities to geometry variations, resulting in superior performances.

1. Introduction

As one of the fundamental methods of thermal energy transport, heat conduction is ubiquitous in daily life. It has broad applications in the design (Liu et al., 2022; Gao et al., 2008) and implementation of industrial heat exchangers (Lohan et al., 2020). To accurately analyze the heat conduction, researchers have extensively studied related mechanisms and described the temperature variation in the heat transfer region by a partial differential equation (noted as PDE). The widely used traditional solvers include several algorithms: e.g., Finite Element (Bruch and Zvoloski, 1974), Finite Volume (Li et al., 2012), Boundary Element (Han et al., 1995) and other methods (Gao, 2017; Fic et al., 2005; Mishra et al., 2009). But when applying the solvers, the complex pre-processing is generally involved, e.g., the differential derivations, integral equations, and modelization. The computation of large-scale or repetitive tasks leads to long computational time and high resources costs.

The latest developments in deep learning technology have achieved

considerable progress in fields such as healthcare (Zeng and Jia, 2024) and transportation (Du, 2023). Recently, this technology has also garnered significant attention in the field of simulation computing, i.e., acceleration in numerical simulation computation without compromising accuracy, numerous classical engineering approaches are transformed and suitable surrogate models are proposed to solve heat conduction problems (Hua et al., 2023; Hua et al., 2023). Because the deep learning technology has the benefits of hierarchical feature learning and the scalability of numerous data, the deduced data-driven models can obtain the approximate temperature field under given geometric conditions. Li et al. (Li et al., 2022) developed a model (hidden temperature method, noted as HTM) for solving temperature field. Based on a data-driven approaches and artificial neural networks, the HTM established the corresponding temperature relationship of nodes during iterative processes. They showed that rigorous mathematical derivation of differential equations is not necessary for the learning process of neural networks, but the problem is solved by learning the underlying regularities from numerous data with physical information.

* Corresponding authors.

E-mail addresses: meimeij@njust.edu.cn (M. Mei), yhua@njust.edu.cn (Y. Hua).

Mohammad et al. (Edalatifar et al., 2021) utilized deep neural network (noted as DNN) to explore the heat conduction in 2D geometry conditions. A large data set is constructed to train the DNN, consisting of various geometry and temperature boundary conditions. The trained DNN can be used to solve new heat transfer problem directly with no iteration or construction of algebraic equations. They highlighted the capability of deep learning methods to learn and solve physical problems even without prior knowledge of the underlying governing equations. Based on deep feedforward neural networks (noted as DFNN), Liu et al. (Liu et al., 2018) developed a data-driven solution, aiming at the complex physical process of boiling. Local features near wall are used to predict the boiling heat transfer. After training the model, the DFNN is capable to accurately capture the boiling patterns even for extrapolation cases, indicating a good generalization property. While data-driven models derived from deep learning demonstrate impressive predictive capabilities, they are subject to two limitations when it comes to practical applications. The first limitation is the requirement for a substantial amount of training data to effectively learn the underlying patterns within data structures. The data is typically obtained through traditional numerical solvers. Therefore, it fails to solve the heavy cost of numerical computation (Ma et al., 2020). The second limitation is the lack of physical interpretability in these models. In physics and engineering domains, prior knowledge is often embedded within training data, but purely data-driven deep learning algorithms do not explicitly incorporate this prior knowledge.

To tackle these limitations, researchers have integrated the information, e.g., balance equations and physical laws, with traditional deep learning algorithms for solving the problems. Raissi et al. (Raissi et al., 2017; Raissi et al., 2017) developed a model using PINN approach to solve the PDEs. They constructed a loss function incorporating physical information. By minimizing this loss function, the neural network is trained to ensure that the predicted results adhere to the underlying physical laws. Additionally, the modified PINN, deduced from Raissi's framework, was successfully applied to various heat transfer conditions, e.g., the heat conduction problems explored by He et al. (He et al., 2021). Adaptive activation functions containing multiple activation layers were applied for direct analysis of the system, which is found to accelerate the convergence speed and improve the solution accuracy. They found that high computational accuracy and efficiency of the framework can be obtained under different conditions. Zobeiry, N. and Humfeld, K.D. (Zobeiry and Humfeld, 2021) utilized the PINN framework to solve heat transfer PDEs, which showed that the heat conduction beyond the training dataset can also be accurately predicted by PINN. For the work of Bararnia, H. and Esmaeilpour, M. (Bararnia and Esmaeilpour, 2022), the PINNs are implemented for the thermal-fluid at the boundary layer. They built a loss function for the network which satisfied the governing equations and boundary conditions. Reasonable prediction results were obtained for three benchmark works, i.e., Natural convection, Falkner-Skan, and Blasius-Pohlhausen. Therefore, PINN is found well suitable for systems which are described by PDEs, and providing numerous advantages compared with other purely data-driven models. Combining the physical laws in partial differential forms with the machine learning, PINN is capable to learn both the patterns and underlying physical laws of the data. Furthermore, PINN is trained for fulfilling governing equations in specific regions, it can achieve high performance prediction in scenarios where observable data is limited or unavailable. PINN models face two significant challenges: high computational training cost and limited scalability of the model (Gao et al., 2021). Automatic differentiation operations (evaluation of PDE residuals) on numerous collocation points are required for the point-wise formulation, large sizes of tensor will be generated by the back-propagation process in the chain rule (Baydin et al., 2018). In the literature of (Jin et al., 2021) and (Jagtap et al., 2020), the effect of various collocation point numbers on PINN training is investigated in the computational domain: more collocation point numbers gave out better model performance. This indicates that longer training time and

more Graphics Processing Unit memories (Karniadakis et al., 2021) are required for PINN. Furthermore, slight modification of inference problem would make the trained network get trapped. For example, most existing PINN frameworks use FNN as the model structure (Li et al., 2022; Cai et al., 2021; Zobeiry and Humfeld, 2021), which are not informed of the geometry characteristics of the computational domain. By this way, the PINN neglected the spatial structure comprised of the point sets and solely computed the gradient for the input point sets, indicating that the PINN was exclusively trained under fixed geometries. But the PINN needs to be retrained if the geometry changes. Therefore, the model sensitivity to geometry changes is needed to enhance, i.e., the model scalability.

A neural network is expected to handle both the spatial point coordinates and the geometric characteristics of the computational region. Fortunately, the limitations are effectively overcome through the convolutional neural network (noted as CNN) (Kim and Kim, 2017) comprising of a network structure with shared-weights (Bodapati and Veeranjaneyulu, 2019), which obtains the spatial characteristics from original data. Based on CNN models, numerous CNN-based geometric adaptive prediction frameworks were developed, such as the one in (Peng et al., 2021; Rajabi, 2022; Lin et al., 2018; Tadeparti and Nadigana, 2022). However, the exploration of PINN-based frameworks with geometric adaptation has been relatively limited compared to pure data-driven models. This is due to the requirement of pixelating the input data into a regular matrix form in traditional CNNs, resulting in the displacement of coordinate information in the original data and thus making it difficult to directly couple the residuals of the differential equations (Morningstar, 2019). To address these limitations, a physical information embedding scheme based on graph neural networks is proposed. Unlike conventional CNNs, the GCN does not require pixelated data preprocessing, thereby preserving the inherent relationships among nodes (Kipf and Welling, 2016). Additionally, the input data in GCN retains the spatial coordinate information for observation points, allowing for the direct calculation of spatial derivatives in PDEs solutions (Peng et al., 2023). In our previous work, this solution was utilized to address the issue of thermal convection (Peng et al., 2023). The model's prediction of the flow field achieved performance comparable to numerical simulation results, demonstrating that the GCN structure effectively enhances the prediction accuracy of the framework by preserving the spatial information of nodes. In this paper, we primarily focus on the structural characteristics of neural networks, explore the enhancement of predictive performance of GCN models, and comprehensively analyze the predictive performance of models trained with pure data-driven and physics-informed approaches. The model is named HCP-PINN, which utilizes GCN and PINN to solve heat conduction PDEs with unstructured grids. The HCP-PINN model goes beyond traditional PINNs by incorporating a data-driven approach. In the training process of HCP-PINN, the physics laws are utilized to constrain the neural network parameters updates, while the deviation between results obtained by network model and numerical simulation is incorporated to enhance the training process. This integrated approach enhances the effectiveness of the model.

In this paper, the feasibility of the model is first verified by solving the Burgers' equation. Subsequently, for the accuracy and effectiveness evaluation of HCP-PINN adapted to various geometries, the heat conduction is investigated for heat sources at different locations. The predicted model results are compared with calculated numerical results to analyze the model performance. Moreover, the results of the physics-informed FNN, purely data-driven GCN, and FNN are compared in terms of error. This work is organized as follows: the basic methods and approaches are described in Section 2, comprised of dataset preparation and training methods. The obtained results and discussions are presented in Sections 3 and 4 for test cases predicted by different trained network models. A summary of the current work is given at last.

2. Method

Three key aspects are presented in this section: the training data preparation (i.e., selection and generation), network structures, and the training methodology. The present prediction model for thermal field comprises two components: i.e., GCN and PINN. The GCN plays a crucial role in identifying and analyzing geometric structural properties, enabling the model adaption with geometric positions variations. On the other hand, the PINN formulates a learning strategy with physical information, and constructs a loss function based on PDEs. The loss function, represented by PDE residuals, gradually converges during training processes. This convergence ensures that the prediction results by the proposed model are closely consistent with physical equations, imparting physical interpretability to the prediction model.

2.1. Mathematical methodology

The selected cases involve two-dimensional steady-state heat conduction scenarios in a square bounded computational region, featuring circular boundaries that can be moved. All boundaries are Dirichlet boundary conditions. The outer square edge boundary and the inner circular boundary are the walls with low and high temperatures, respectively. It is noted that the movable circular boundary is not treated as a moving heat source but rather as a thermal boundary. Consequently, the heat conduction within the computational region is expressed by a second-order PDE (the Laplace equation):

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

where $T(x, y)$ is the temperature at the position (x, y) , with boundary conditions as follows: $T(x, 0) = T(x, 1) = T(0, y) = T(1, y) = T_{min}$; $T(x_i, y_i) = T_{max}$; $x_i, y_i \in \phi_c$. ϕ_c is the circular boundary in the computational domain, T_{min}, T_{max} are low temperature and high temperature, respectively. The temperature and coordinates are normalized as follows:

$$T^* = \frac{T}{T_{max}}, x^* = \frac{x}{D}, y^* = \frac{y}{D} \quad (2)$$

where the temperature is normalized by the high temperature and the characteristic length D is used to make coordinates dimensionless. Based on above descriptions, the whole physical field is presented in Fig. 1.

2.2. Data pre-processing

The graph is represented by $G = (V, E)$, where V and E ($(v_i, v_j) \in E, i, j \in N$) are graph nodes and edges, respectively. The relationship between graphs and grids is described as follows: vertices V are mesh nodes, edges E represent the mesh connectivity, the set $\mathcal{N}(v_i)$ denotes the neighbors of node v_i . The schematic process of generating mesh into

graph data is displayed in Fig. 2.

As shown in Fig. 2, the relationships among points are complex. To facilitate the computation of graph data, the commonly used representation is an adjacency list. For all vertices, all connected vertices in the graph are stored. Additionally, considering the feature properties for all vertices, a feature matrix is employed to save these features. The mesh data is preprocessed by transforming node connections into graph data, which aligns better with the structural characteristics of graph data and enables effective model training. For mesh generation, the gmsh library in Python is utilized. The generated grids are unstructured and non-uniform. To accurately simulate the boundary layer effect, a denser mesh is employed near boundaries. In addition, the generated mesh file is examined to extract the individual node index and its neighboring nodes. This information is then used to build adjacency lists for graph data.

Furthermore, the probe node information is processed to capture temperature data at specific positions in computational regions. The probe coordinates, along with their corresponding temperature information, are organized into the feature matrix and the training label, respectively. The feature matrix $X \in R^{n \times m}$ consists of the coordinates of n nodes, which has m features individually. Labels for training the model are the temperature results. It is highlighted that, during numerical calculations, a sufficiently large mesh size (nodes number) is utilized to ensure high-quality labels with enhanced accuracy. Conversely, for generating the adjacency table and probes, a relatively smaller mesh size is employed to reduce the computational cost for the neural network training.

Following data pre-processing, the raw data is divided into two distinct sets: the training and the test sets. This division enables the performance evaluation of the model during training. 20 cases are applied in the training set, where each case involves a circle (i.e., the high temperature boundary) placed at different locations within the computational region. These cases are utilized for model training, enabling the adaptive prediction of geometric transformations. The test set also serves as an independent dataset used to assess various aspects of the model's performance after completing the training, e.g., prediction accuracy, robustness, and extrapolation capabilities, etc.

2.3. Network structures

The HCP-PIGN model is utilized to predict two-dimensional heat conduction phenomena with thermal wall surfaces at any positions in cavities. Since multi-geometric position transformation is involved, more advanced GCN is used to learn data rules, enabling an adaption to changes of geometric position. Furthermore, a classic FNN is built to analyze and compare the GCN performance. Specifically, the GCN and the FNN is introduced one by one. Finally, the physical information method embedding neural network is discussed.

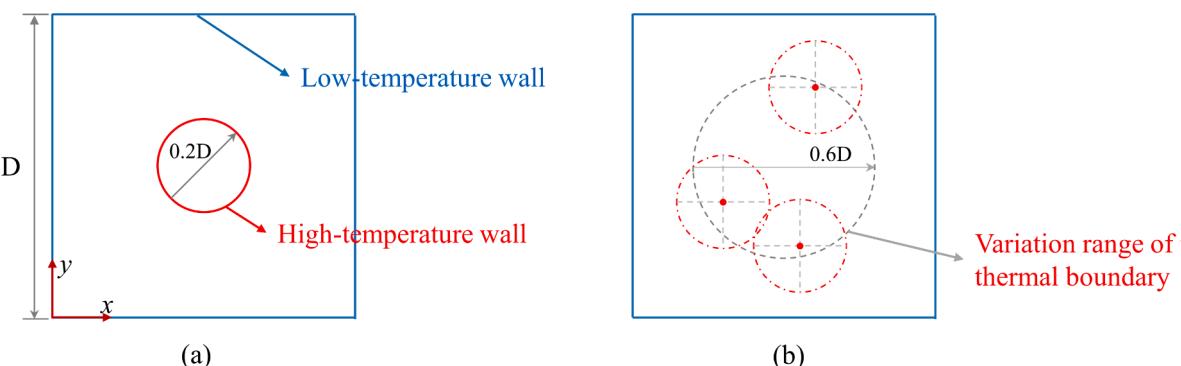


Fig. 1. (a) Boundary conditions in Physical field; (b) Distribution range of thermal boundaries in computational domain.

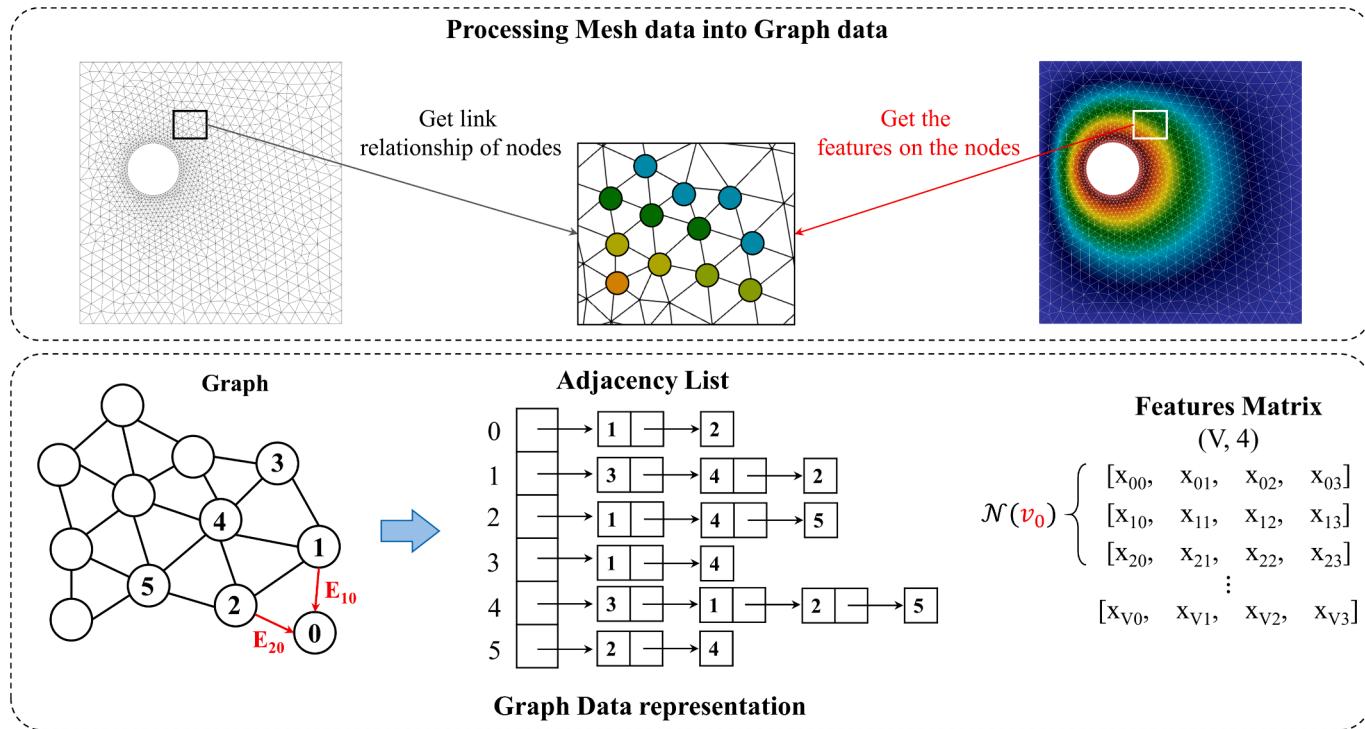


Fig. 2. The process of generating mesh data into graph data (containing adjacency and feature matrixes).

2.3.1. GCN

GCN is a neural network specifically designed to work with graph structures, allowing it to handle non-Euclidean data effectively. The GCN operation involves two main components: node feature propagation between nodes and node feature propagation between layers. The overall structure of GCN is depicted in Fig. 3.

In GCN, the individual node information in the subsequent layer is determined by combining its own information from the previous layer with its neighboring nodes' information, which is achieved by the weighted summation, followed by a non-linear transformation applied to the summation. The information propagation among layers can be summarized as follows:

$$H^{l+1} = \sigma \left(D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} H^l W^l \right) \quad (3)$$

where \tilde{A} ($\tilde{A} = D - A$) is the Combinatorial Laplacian matrix, which is calculated by the adjacency matrix A (converted by the adjacency list) and the degree matrix D , respectively. The matrix \tilde{A} effectively addresses the issue of self-transmission by considering the influence of each node

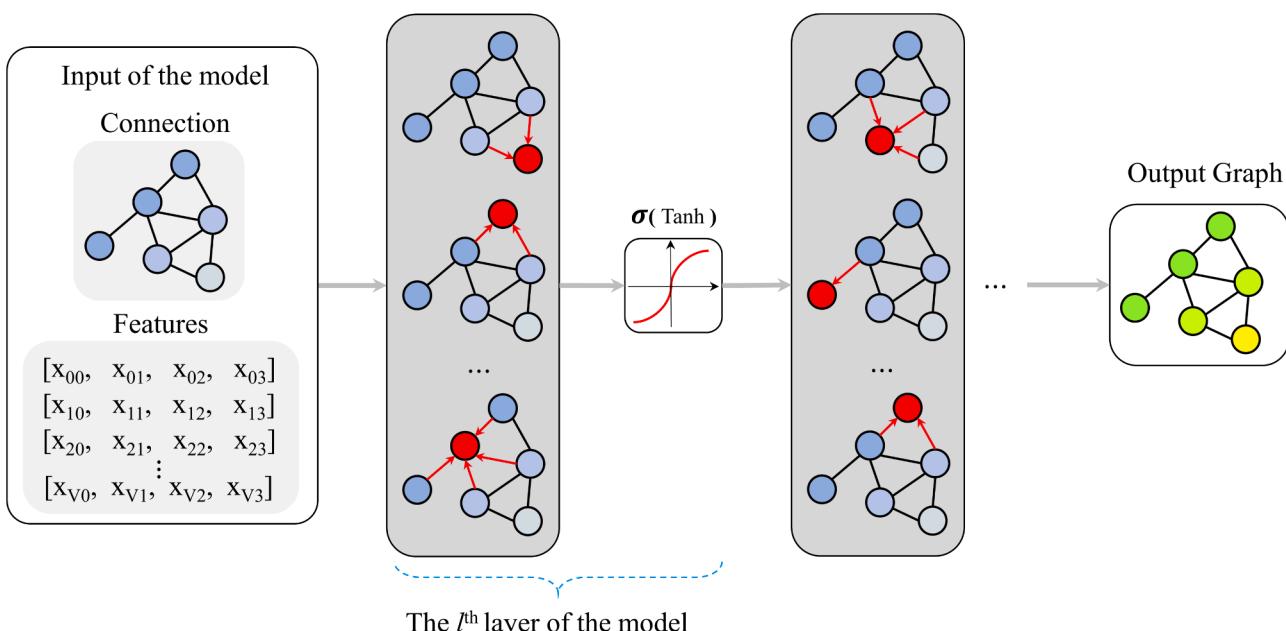


Fig. 3. The GCN architectures.

on itself. And $D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}$ represents the normalization of adjacency matrices which balances the influence among nodes. H^l is the hidden feature at the l^{th} layer. For the 0th and L^{th} layers, $H^0 = X$ and $H^L = Z$. Both the input and output matrices, i.e., X and Z , consists of n nodes, while they have m and c features as the input and output, respectively. The weight matrix W is learnable and undergoes continuous updates during training. The activation function $\sigma(\bullet)$, such as $\tanh(\bullet)$, is applied. The structural parameters of the GCN and the hyperparameters of the training model are listed in detail in [Table 1](#).

The critical difference of GCN from previous CNN model is its utilization of a message passing scheme to compute interconnections among nodes. This scheme applies GCN for information-passing between directly-connected nodes. The message passing scheme is composed of two stages: message passing and update. The message passing process can be defined as:

$$h_v^{t+1} = U_t \left(h_v^t, \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t, e_{vw}) \right) \quad (4)$$

where h_v^t represents the feature vector, and subscript and superscript are the node v and the t^{th} iteration, respectively. e_{vw} denotes the edge feature vector between nodes v and w . $M_t(\bullet)$ and $U_t(\bullet)$ represent the message and update functions of the node, respectively. The processes showing the message passing and update are displayed in [Fig. 4](#).

During the message passing process, blue and yellow nodes represent the initial and updated node features, respectively. Nodes update their own features and accumulate the neighboring nodes' features. The update feature representation is obtained by combining these two types of features. Specifically, for node v , the accumulation and updating process can be summarized as follows:

$$\begin{aligned} m_v^{(t+1)} &= \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t, e_{vw}) \\ h_v^{(t+1)} &= U_t(h_v^t, m_v^{(t+1)}) \end{aligned} \quad (5)$$

where $w \in \mathcal{N}(v)$ indicates that the node w is the neighbor for node v . $m_v^{(t+1)}$ is the information, and subscript and superscript are the node v and time step $t + 1$, respectively.

2.3.2. FNN

The FNN is a dense and feedforward neural network, which contains three categories of layers: the input, output, and hidden layers at the 0 layer, the n layer, and the remaining $n-1$ layers, respectively. All nodes in $n-1$ layer are individually connected to all nodes in n layer. The input of each node in n layer is the weighted summation of all nodes in $n-1$ layer. Furthermore, every layer contains several neurons in the network, but there is no message passing between nodes on the same layer during computation, which is different from GCN. [Fig. 5](#) shows the FNN structure in detail.

For structure design, the layer number in FNN is identical with that in the GCN, with 128 nodes per layer. The main purpose of the FNN model during training is constructing the mapping relationship between

Table 1
The hyper-parameters of the GCN model.

Hyper-parameter (GCN)	Structure of GCN
Inputs. (number of cases, nodes per case, features per node)	$X, (N, n, 4)$
Outputs. (number of cases, nodes per case, features per node)	$Z, (N, n, 1)$
Number of layers and neurons per layer. (ℓ, n_ℓ)	(6, 128)
Activation function	Tanh()
Batch size	1
Learning rate	1e-4

geometric features (i.e., input data) and temperature field (i.e., output data), which can be represented by the formula:

$$T = f(\mathbf{X}, \mathbf{w}, b) \quad (6)$$

where T denotes the output, \mathbf{w} is the weight matrix, and b denotes the bias term. \mathbf{X} denotes the input data in FNN model, which has identical data structure with the input GCN, except that the adjacency matrix is removed. The information transmission process of FNN from node i in the previous layer to a node in the next layer can be represented as:

$$\mathbf{x}_i^j = \sigma \left(\sum_{i \in K^{j-1}} w_{ii}^{j-1} \mathbf{x}_i^{j-1} + b_i^j \right) \quad (7)$$

where the superscript j and subscript i are the layer number and the layer neuron in the FNN. The neurons number in each layer is represented by K^{j-1} . Similarly, before passing the information to the next neuron, the data is processed by the activation function $\tanh()$ to give the network the ability to fit non-linear data.

2.3.3. PINN

The PINN incorporates physical information into the network structure. It leverages prior knowledge of the underlying problem physics to formulate a loss function for optimization. PINN's basic network structure and training methodology are depicted by solving a nonlinear PDE as illustration. The PDE can be expressed by the general form as follows.

$$\begin{cases} \mathbf{u}_t + \mathcal{N}_{\mathbf{x}}[\mathbf{u}] = 0, \mathbf{x} \in \mathbb{R}^d, t \in [0, t_T] \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{h}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t), \mathbf{x} \in \mathbb{R}^b, t \in [0, t_T] \end{cases} \quad (8)$$

where $\mathbf{x} = (x, y)$ and $\mathcal{N}_{\mathbf{x}}$ represent the point position coordinates and the general nonlinear differential operator. t and t_T denote the temporal coordinates and terminal time, respectively. \mathbb{R}^d and \mathbb{R}^b represent the computational domain and boundary, respectively. Under the initial condition $\mathbf{h}(\mathbf{x})$ and boundary condition $\mathbf{g}(\mathbf{x}, t)$, $\mathbf{u}(\mathbf{x}, t)$ is the calculated solution for PDEs. Here, the neural network can be either an FNN, a GCN, or other networks. The PINN uses $\mathbf{n}(\mathbf{x}, t; \theta)$ to approximate $\mathbf{u}(\mathbf{x}, t)$. The PINN is a powerful approach for accurately solving problems with limited data by incorporating known physical information, e.g., governing equations and boundary conditions, into the loss function of a conventional neural network structure. In PINN, the inputs are the coordinates (\mathbf{x}, t) , and the network gives outputs the residual of the partial differential equations (PDEs).

$$f(\mathbf{x}, t; \theta) = \frac{\partial}{\partial t} \mathbf{n}(\mathbf{x}, t; \theta) + \mathcal{N}_{\mathbf{x}}[\mathbf{n}(\mathbf{x}, t; \theta)] \quad (9)$$

The required partial derivatives in equations can be gained through automatic differentiation ([Baydin and Pearlmutter, 2014](#)) within the PINN framework. The PINN loss function is defined as follows:

$$\mathcal{L}(\theta) = \mathcal{L}_f(\theta) + \mathcal{L}_b(\theta) + \mathcal{L}_0(\theta) \quad (10)$$

where $\mathcal{L}_f(\theta)$, $\mathcal{L}_b(\theta)$, and $\mathcal{L}_0(\theta)$ represent the losses associated with the governing equations, boundary, and initial conditions, respectively. Notably, all the loss terms are dependent on the network parameters θ , which consist of weights and neuron biases. [Fig. 6](#) illustrates the schematic diagram of proposed PINN.

These parameters, restrained by the governing equation and computed by automatic differentiation, are used to calculate the loss, which is backpropagated to update the weights of the neural nodes in the GCN. Numerical discretization is not involved here. And detailed training process is displayed in following sections.

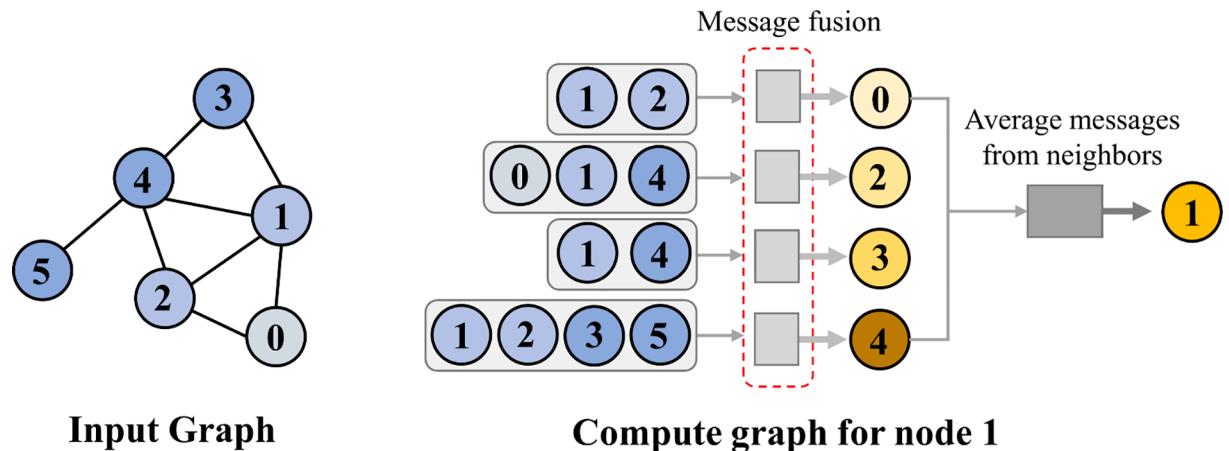


Fig. 4. Processes showing the message passing and updating for the node v_1 and its neighbors applied in graph data.

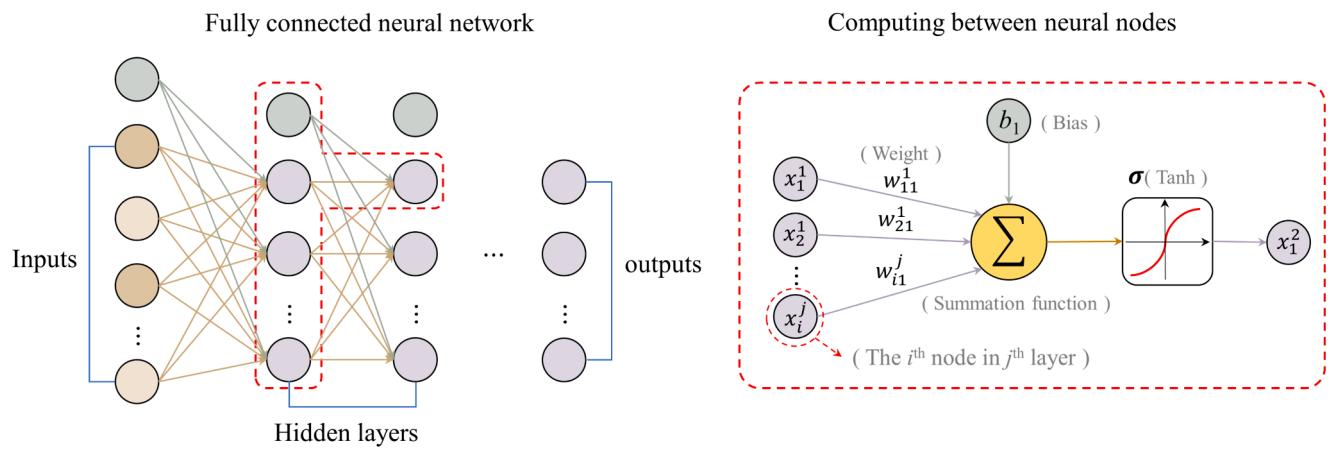


Fig. 5. Schematics of the FNN structure.

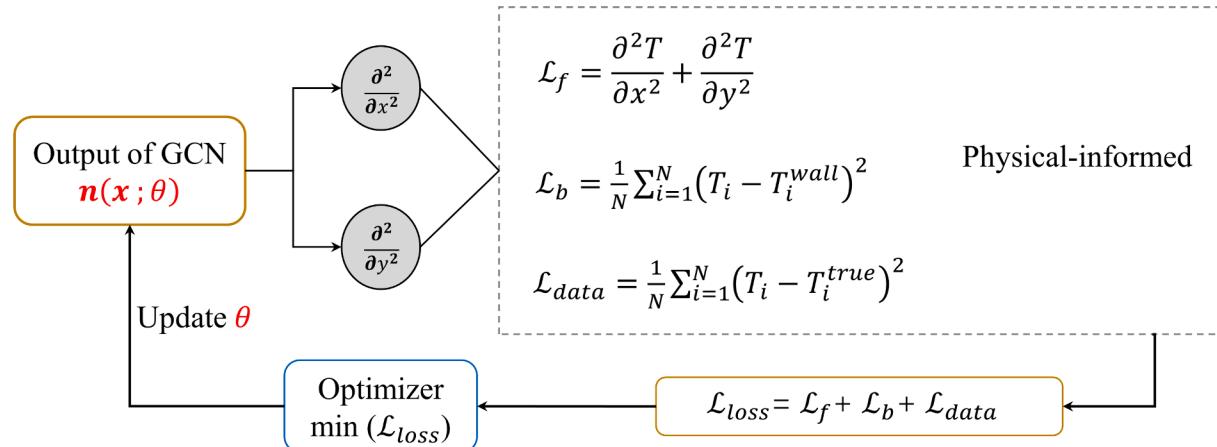


Fig. 6. Schematic diagram showing PINN incorporated with heat conduction equations.

2.4. Training implementation

In this study, the steady-state two-dimensional heat transfer phenomenon is investigated. Our objective is to train the HCP-PIGN model to accurately predict the temperature field for different thermal boundary configurations in computational regions. Since the heat

transfer process is steady-state, the time component or initial conditions are not considered for the loss function. Hence, we define the loss term as follows.

$$\mathcal{L}(\theta) = \mathcal{L}_f(\theta) + \mathcal{L}_b(\theta) + \mathcal{L}_{data}(\theta) \quad (11)$$

where $\mathcal{L}_f(\theta)$, $\mathcal{L}_b(\theta)$, and $\mathcal{L}_{data}(\theta)$ are expressed as follows:

$$\begin{aligned}\mathcal{L}_f(\theta) &= \frac{1}{\mathcal{Z}_1} \sum_{n=1}^{\mathcal{Z}_1} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right)^2, \quad \mathcal{Z}_1 \in \mathbb{R}^d \\ \mathcal{L}_b(\theta) &= \frac{1}{\mathcal{Z}_2} \sum_{n=1}^{\mathcal{Z}_2} (\hat{T} - T)^2, \quad \mathcal{Z}_2 \in \mathbb{R}^b \\ \mathcal{L}_{data}(\theta) &= \frac{1}{\mathcal{Z}_3} \sum_{n=1}^{\mathcal{Z}_3} (\hat{T} - T)^2, \quad \mathcal{Z}_3 \in \mathbb{R}^{label}, \quad \mathbb{R}^{label} \in \mathbb{R}^d\end{aligned}\quad (12)$$

where \mathcal{Z}_1 , \mathcal{Z}_2 and \mathcal{Z}_3 are the node numbers located in domain \mathbb{R}^d , on the boundaries \mathbb{R}^b , and in domain \mathbb{R}^{label} , respectively. The point set in the region \mathbb{R}^d forms \mathbb{R}^{label} , which is utilized as input for the neural network. In our approach, both a purely data-driven and a physics-informed neural networks are trained, as depicted in Fig. 7.

The purely data-driven model needs a large training data size, which is often expensive if achieved by numerous Computational Fluid Dynamics (noted as CFD) simulations. On the other hand, a PINN trained for one specific problem may not generalize well to similar problems. To address this, a method is proposed to incorporate physical equations into the data-driven approach, reducing the required training data while ensuring the model's predictive capability. For training, the following conditions are settled: 800 nodes on both the Dirichlet wall and cylinder boundaries, and approximately 1300 grid nodes in the domain to compute residuals. The label number corresponds to the grid points number in the computational region, denoted as \mathbb{R}^{label} . The dataset used in the paper has been published on GitHub (<https://github.com/JZPeng1/HCP-PINN>). The overall system loss is optimized by the Adam optimizer, and 5,000 iterations is adopted for model training with a learning rate of $1e^{-4}$. Fine-tuning is performed using the L-BFGS optimizer (Bollapragada et al., 2018). Prior to model training, the minimum–maximum normalization is applied to the input coordinates to expedite convergence. The learning models are implemented using PyTorch and trained on a single Nvidia GeForce RTX 2080ti graphics

card.

3. Results

In this section, the capability of HCP-PIGN in predicting one-dimensional and two-dimensional heat transfer problems are investigated. For the one-dimensional problem, we solved Burgers' equation using HCP-PIGN with the data provided in the paper (Raissi et al., 2017). As for the two-dimensional problem, we created our own test data that included examples of heat transfer with different boundary positions without including the training data. Additionally, the performance of four different network structures on the test set are compared for the two-dimensional problem.

By calculating the magnitude of the errors between the model's predicted results and the numerical simulation results, the accuracy of the proposed model's predictions is evaluated. Major metrics include the mean absolute percentage error (E_{MAPE}), the maximum and average value of the E_{MAPE} of all cases, relative error, and coefficient of determination (R^2) of the predicted results. They are defined as follows,

$$\begin{aligned}E_{MAPE} &= \frac{1}{n} \sum_{i=1}^n \left| \frac{T_i - \hat{T}_i}{T_i} \right| \times 100\%; E_{max} = \max(E_{MAPE}); E_{mean} = \frac{1}{N} \sum_{i=1}^N E_{MAPE} \\ E_{relative} &= \frac{(T_i - \hat{T}_i)}{T_i}; R^2 = 1 - \frac{\sum_{i=1}^n (T_i - \hat{T}_i)^2}{\sum_{i=1}^n (T_i - \bar{T})^2} \quad (13)\end{aligned}$$

where n represents the nodes in each calculation case, N denotes the number of cases, T and \hat{T} are the simulated temperature field by OpenFOAM and the predicted temperature field.

3.1. Solving Burgers' equation

The implementation of predicting the one-dimensional problem is to verify the model feasibility. The Burger's equation is represented as,

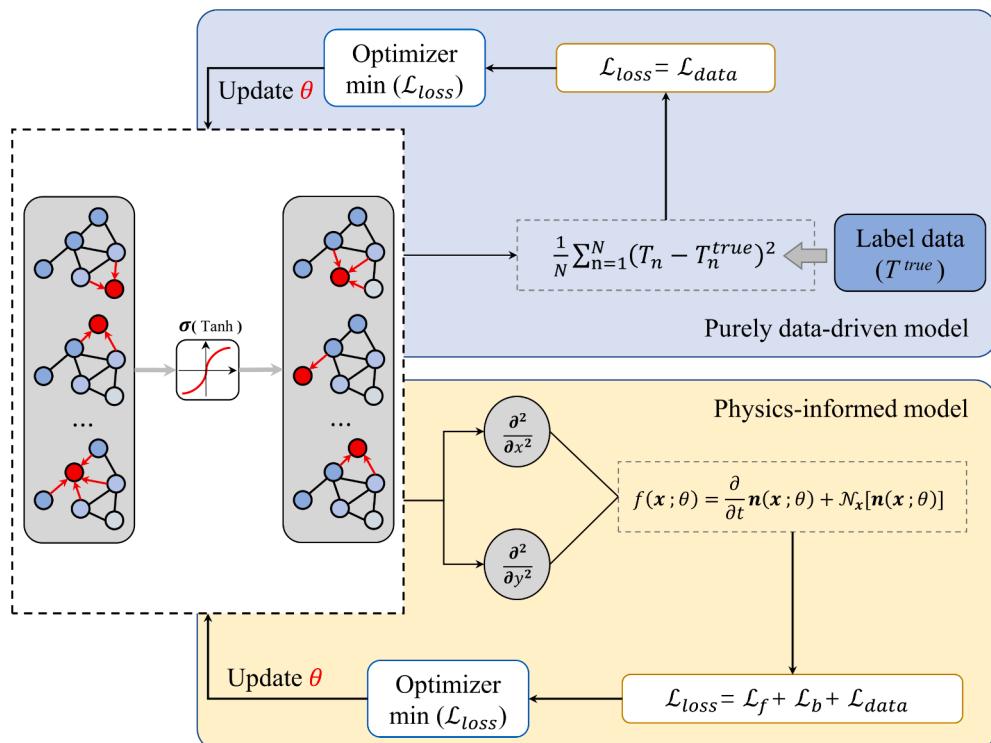


Fig. 7. GCN based purely data-driven prediction model and physics-informed GCN prediction model.

$$\begin{aligned} u_t + uu_x - \left(\frac{0.01}{\pi}\right)u_{xx} &= 0, x \in [-1, 1], t \in [0, 1] \\ u(0, x) &= -\sin(\pi x) \\ u(t, -1) &= u(t, 1) = 0 \end{aligned} \quad (14)$$

where $u(t, x)$ is the hidden solution, predicted by HCP-PIGN model. The loss function after embedding the Burger's equation into the neural network is,

$$\begin{aligned} \mathcal{L}(\theta) &= \mathcal{L}_f(\theta) + \mathcal{L}_b(\theta) + \mathcal{L}_0(\theta) \\ \mathcal{L}_f(\theta) &= \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - \frac{0.01}{\pi} \bullet \frac{\partial^2 u}{\partial x^2} \right)^2 \\ \mathcal{L}_b(\theta) &= u^2 \\ \mathcal{L}_0(\theta) &= (u + \sin(\pi x))^2 \end{aligned} \quad (15)$$

The results are summarized in the Fig. 8. At time points 0.1, 0.5, and 0.8, the predicted results of HCP-PIGN model are consistent with exact solutions, with an average absolute error relative to the exact solutions of only 1.5 %, showing the proposed model is able to understand simple systems and make accurate predictions.

3.2. Solving 2D temperature field under various cases

The accuracy of HCP-PIGN model for the 2D temperature field prediction and generalization model performance are the main focuses. Therefore, three test cases are used to test the predictive performance of HCP-PIGN, and the results are compared with numerical simulations. As shown in Fig. 9. Following iterative training, our model demonstrates remarkable accuracy in predicting the temperature field within the computational region. Notably, all prediction results exhibit excellent agreement with the corresponding numerical solutions. The contour maps depicting the temperature magnitude exhibit consistent and logical patterns as the circular heat boundary undergoes movement within the computational domain. In particular, the HCP-PIGN model adaptively captures the underlying heat transfer principles across varying geometric positions. The figure showcases three distinct test cases where the heat boundaries are located at different positions within the computational domain. With a fixed vertical coordinate of 0.45D, the horizontal coordinates are 0.48D, 0.64D, and 0.73D, respectively. Since the computational domain has a Dirichlet boundary condition and a low-temperature wall, the temperature gradient change near the wall is more significant when the heat boundary is closer to the wall. This conduction property is clearly captured by the HCP-PIGN model.

In Fig. 9, the temperature distribution along a profile passing through the center of the circle and parallel to the x -axis ($y = 0.45D$) is selected as the reference. Different colored lines represent three different test cases. Each line is divided into two parts on either side of the heat boundary, and preserving empty inside the heat boundary due to zero heat flux assumption.

For statistical analysis, the maximum and average prediction errors of all test case are displayed in Fig. 10. The model's prediction error

increases with the circular heat boundary approaches cold walls. And the major errors are concentrated on the left side of heat boundary. Although there are some slight deviations, the maximum prediction errors in all three cases are below 3 %, and the average errors are also small, below 0.5 %, indicating that the accuracy by HCP-PIGN model is identical with that by CFD model.

3.3. Comparison of various structural models

In this work, GCN is introduced to learn geometric feature representations in non-Euclidean domains, and use physics-informed neural networks to train the model with prior physical knowledge. To demonstrate the effectiveness of these network modules, we discuss the advantages of the HCP-PIGN model over single module, and analyze the prediction accuracy, verifying the framework efficiency for heat conduction calculation. In Fig. 11, the temperature fields computed by different models are shown in the first row, including the CFD, proposed HCP-PIGN, PINN-FNN, data-driven GCN, and data-driven FNN models.

The selected test case is the heat boundary with a center coordinate of (0.47, 0.75) D. For the computational domain in this case, the heat boundary is near the upper boundary, there is thus a sharp temperature change near the cold boundary. Among the predicted results of various models, only the HCP-PIGN model predicts results that approximate numerical calculation. Other models fail to predict a reasonable temperature field near the cold boundary. In conclusion, models with physical information embedding or graph convolutional neural networks are more efficient in learning at the boundaries. The former is caused by the direct application of boundary conditions as constraints during training, while the latter benefits from the message passing mechanism of GCN, where feature transfer between adjacent nodes is more in line with the thermal conduction mechanism, leading to better performance than FNN models. One thing worth noting is that pure data-driven models can also capture the underlying mechanisms of heat conduction when there is sufficient training data available. However, the training dataset used in this paper only includes 20 cases, and under these situations, the GCN with physics information embedding demonstrated the best performance. Furthermore, we conducted quantitative analysis on the error results to further explain the impact of different module combinations on prediction accuracy. Fig. 12 shows the related results.

As seen from Fig. 12, the HCP-PIGN model predicts results with both the maximum error and average error smaller than the other three models. Through the evaluation of the model's coefficient of determination (R^2), it was found that the R^2 for HCP-PIGN is 0.9919, while the R^2 for PINN-FNN is 0.9815. This indicates that the HCP-PIGN model exhibits a higher level of fit compared to the PINN-FNN model. Overall, both the PINN-FNN model (b) and the data-driven FNN model (d) have relatively large average errors. The boxplot results indicate that the relative deviation of the HCP-PIGN model's predicted temperature field is concentrated around 0, while the prediction errors of the FNN-based models are mostly far from 0. Subsequently, we evaluated the prediction results of each model using 40 test cases to weaken the impact of

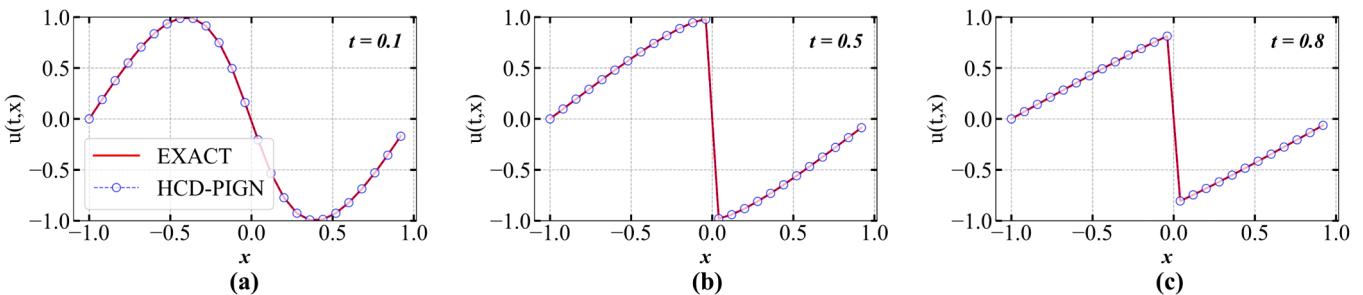


Fig. 8. Comparison between the exact solutions and HCP-PIGN predictions under three moments. (a) $t = 0.1$. (b) $t = 0.5$. (c) $t = 0.8$.

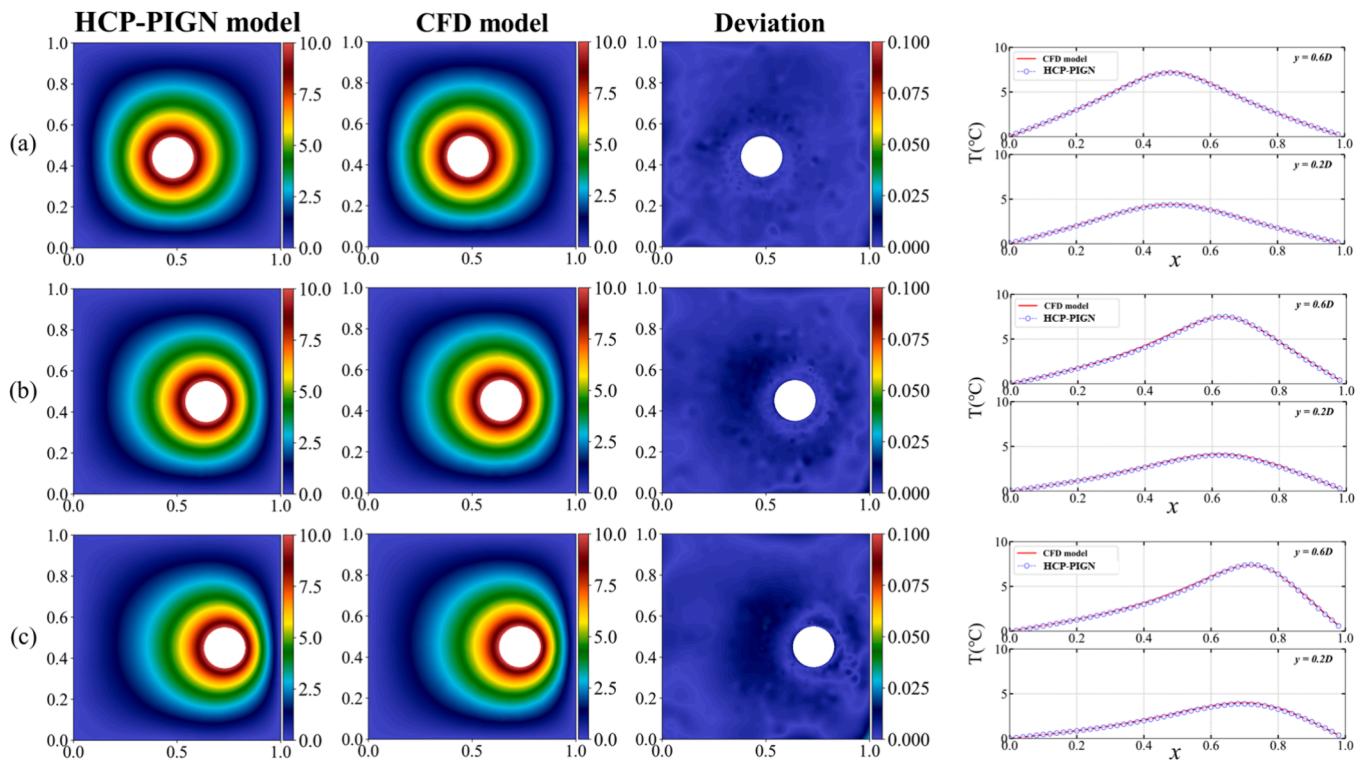


Fig. 9. Comparison of temperature field obtained by the proposed model and the numerical model, and the corresponding deviations between two methods, under three thermal boundary conditions with changing of hot temperature wall: The centroid of inner circle boundary are (a) [0.48, 0.45] D, (b) [0.65, 0.45] D, and (c) [0.75, 0.45] D. The fourth column is the temperature profile, distributed along the straight line at $y = 0.2$ and $y = 0.6$.

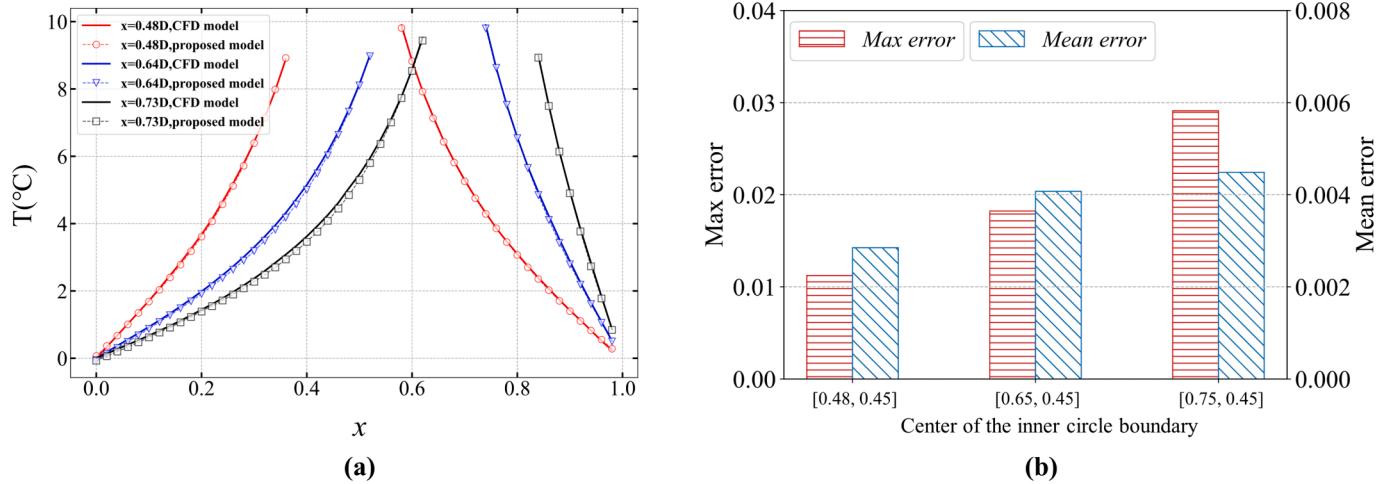


Fig. 10. (a) The temperature profile passing through the circle center (parallel to the y-axis). (b) The maximum and mean prediction errors for all test cases.

occasional errors, as shown in Fig. 13.

In addition, the average relative error of 40 test cases is calculated, which is presented in the form of a blue whisker plots. We found that in the case of predicting 2-D heat conduction phenomena with 20 training data, the HCP-PIGN has the smallest average error (less than 1.3 %), with an improvement of 34.6 %, 28.1 % and 46.5 % compared to the PINN-FNN, the pure data-driven GCN, and the pure data-driven FNN, respectively. This further indicates that the learning approach of FNN tends to exhibit large bias in prediction results due to the lack of consideration of the mutual influence between nodes when data is limited. In contrast, the model with GCN has smaller prediction error for the temperature field. Therefore, it can be inferred that the introduction of GCN has stronger adaptability for predicting 2D temperature field

heat conduction.

3.4. Comparison of the model scalability

The most interesting aspect of surrogate models is their ability to predict problems of the same category, such as the heat conduction cases with different thermal boundary positions used in this paper. After training, the model gives reasonable prediction of the temperature distribution in the computational domain with position variations of the thermal boundary. Furthermore, for the generalization ability of surrogate models, we designed a heat conduction case with an elliptical thermal boundary in this section. This is a challenging prediction task for the models trained only with circular thermal boundaries, as the

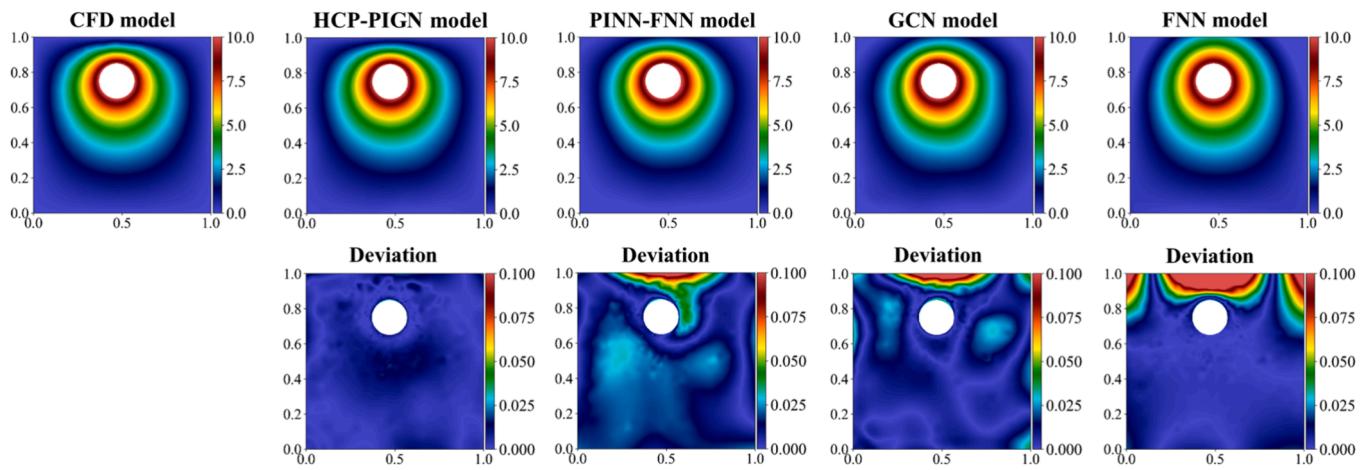


Fig. 11. Comparison of temperature field predicted by different types of models, i.e., CFD, HCP-PIGN, PINN-FNN, GCN, and FNN model.

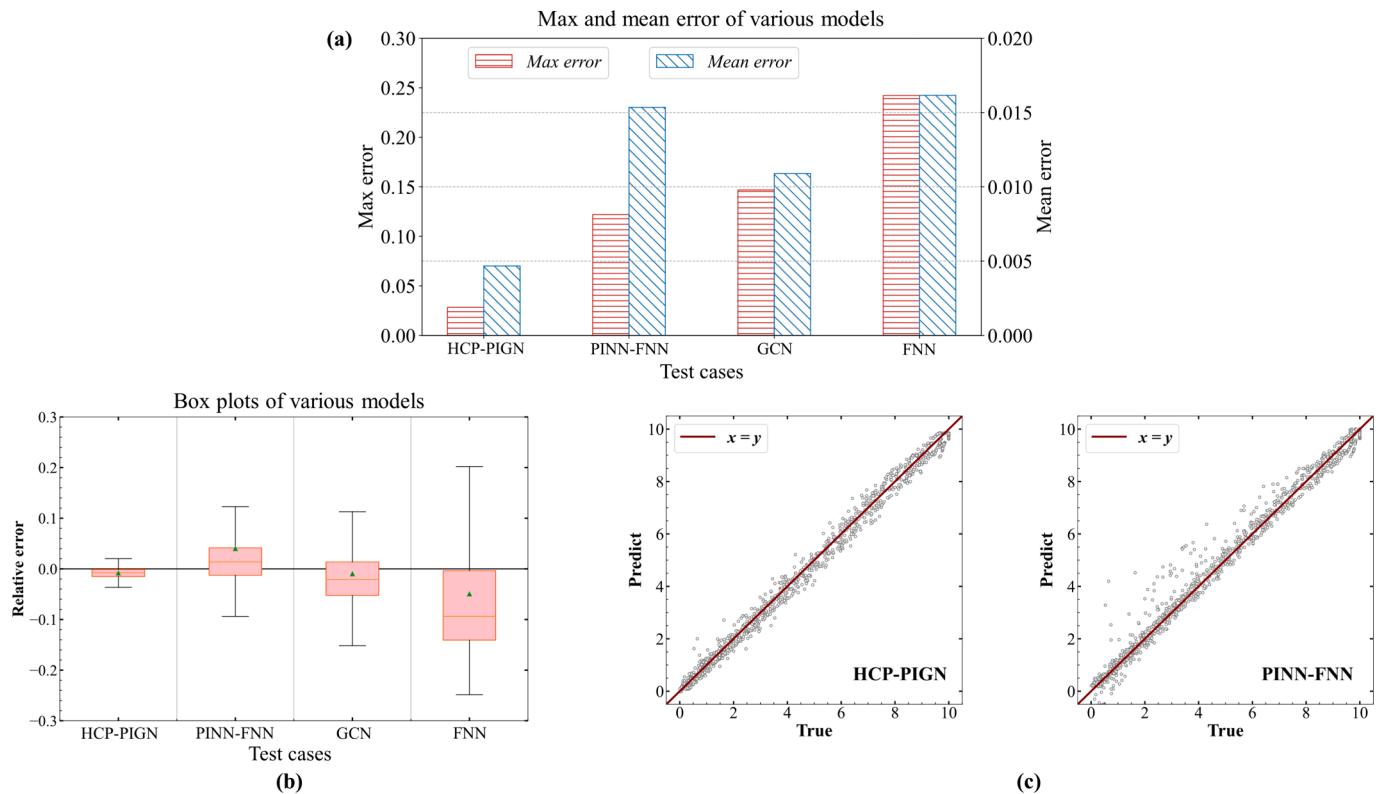


Fig. 12. (a) Comparison of the maximum and mean prediction errors by different models. (b) The box plots show the relative errors predicted by different models. (c) The error distribution of the test results of the HCP-PIGN and PINN-FNN models relative to the numerical results, respectively.

current models have not learned the shape features of an ellipse. Fig. 14 presents the comparison of different models' prediction performance.

The selected test case is an elliptical thermal boundary with a center coordinate of $(0.58, 0.58) D$ and a major axis and a minor axis of $0.24 D$ and $0.18 D$, respectively. Since the elliptical shape is a geometry that has never appeared in the training set, being able to reasonably predict the temperature distribution under the influence of the elliptical heat boundary can further demonstrate the model's scalability. The model embedded with physical information shows relatively low prediction errors. Due to the significant change in the shape of the heat boundary, the errors mainly concentrate at the boundary. In addition, the HCP-PIGN model performs better than the PINN-FNN model. It can be inferred that the introduction of GCN makes the model more sensitive to geometric changes, enhancing its adaptive ability to geometry. The

quantitative statistical results of all test cases are shown in Fig. 15.

In Fig. 15, the predicted maximum and mean errors for each case were first calculated, and then the error distribution density for each case was visualized. The error distribution density shows that the relative error of the HCP-PIGN model for most sample points is lower than 5 %. Although the relative error of the GCN model's predicted results is mostly distributed in the range of $\pm 5 \%$, its maximum error is larger compared to the HCP-PIGN model. In addition, the value of R^2 for the prediction results based on the GCN framework is 0.9946, while the R^2 based on the FNN framework is only 0.9859, indicating that the GCN framework has higher accuracy in predicting the temperature distribution of the elliptical thermal boundary and has better scalability than FNN.

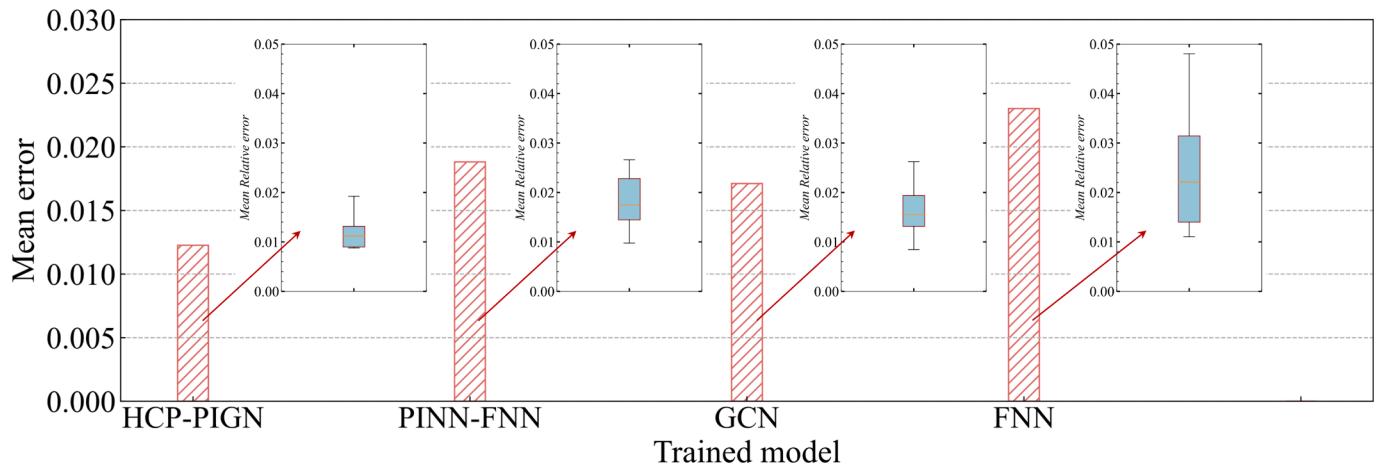


Fig. 13. Statistical results of relative errors of temperature field predicted by different types of models on 40 examples.

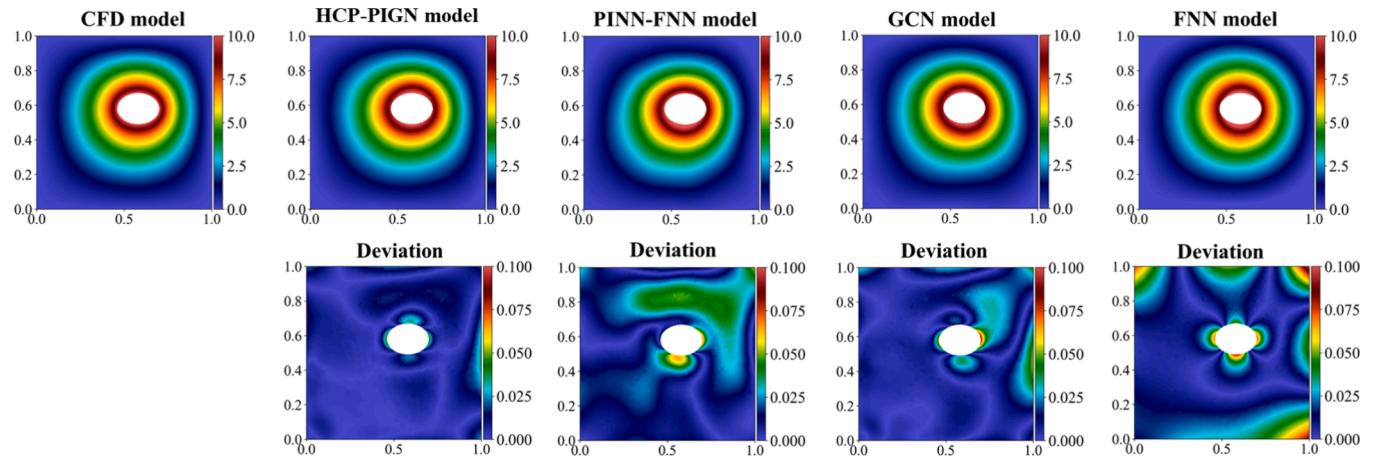


Fig. 14. Comparison of the results of different types of models predicting the temperature field of an elliptical thermal boundary.

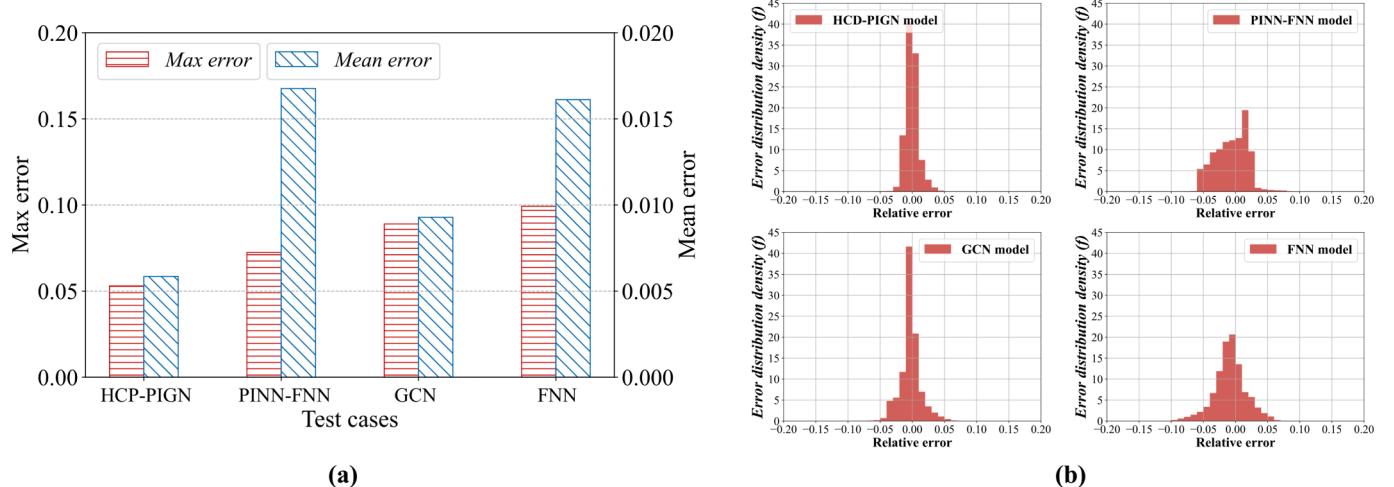


Fig. 15. (a) Comparison of maximum and mean errors for different test cases. (b) The frequency distribution of relative errors for different models in predicting temperature field under elliptical thermal boundary.

3.5. Effects of training data size on the model

The previous validation has shown that the HCP-PIGN model can effectively learn the potential mechanisms of the temperature field changes caused by the variations in the heat boundary position with

only 20 training cases. Undoubtedly, the purely data-driven framework will also have certain performance improvement with sufficient data for training. To compare training data sizes required by different frameworks, this section selects 20, 50, and 100 cases from the prepared database as the training sets for both physics-informed and purely data-

driven GCN models. Among them, the pure data-driven model is trained by the training set with 100 data. All models use the same hyperparameters (including the number of layers, nodes, learning rate, and iteration times, etc.) to make comparative analysis. Finally, the prediction results of these trained models for the heat boundary with the center coordinates of $(0.5D, 0.8D)$ is compared in Fig. 16.

Due to the superior performance of GCN in the current work, only the physics-informed framework and pure data-driven framework based on GCN were selected for testing the models. The training data size affects both model performance, and larger training data sizes lead to superior predictive performance. It is found that the pure data-driven model is more sensitive to the training data sizes. For the temperature field distribution, the predicted results of the training models with 10 or 20 training data have a large deviation near the cold wall, and the training model with 10 data cannot even learn the influence of the cold wall on the temperature distribution. Only when training data size exceeds 50, the pure data-driven model shows strong predictive performance. In contrast, the HCP-PIGN model is able to the predicted temperature field identical to numerical results under only 10 training data. To further quantitatively compare the differences among test cases, the corresponding statistical analysis results are displayed in Fig. 17.

Less than 50 data points are used to train HCP-PIGN model, so there was no corresponding data on 100 nodes. This is because the HCP-PIGN model's maximum and mean prediction errors for the temperature field do not improve significantly when training data size exceeds 20. For the predicted relative errors of temperature fields, the HCP-PIGN model's predicted errors are concentrated near 0. The maximum relative error decreases with the increase of the training data sizes. On the other hand, the GCN model shows larger prediction errors under the scarce data. However, with the increase of training data sizes, both the maximum and average errors drop sharply, and the prediction performance surpasses that by HCP-PIGN model after reaching a data size of 100. This result shows that the pure data-driven model can achieve excellent generalization performance when the training data size is sufficient, and that the GCN with physical information constraints reduces its reliance on data size during training.

4. Discussion

In this section, we will discuss why GCN can efficiently solve heat conduction problems. Starting from the phenomenon of heat conduction and its mathematical model, we will combine the computational principles of GCN to discuss the adaptability of the HCP-PIGN model to heat transfer physical problems. This aims to demonstrate the strong feature learning ability of GCN for node information with high spatial correlation.

Taking the one-dimensional heat conduction model as an example. Suppose a homogeneous one-dimensional chain object, where each unit

on the chain has a different temperature, and the temperature spreads between adjacent units. For the i^{th} node, it is only adjacent to the $(i-1)^{\text{th}}$ and $(i+1)^{\text{th}}$ nodes, and it receives heat from them. Let T_i denote the temperature of the i^{th} node at time t , then we have the following equation:

$$\frac{dT_i}{dt} = k(T_{i+1} - T_i) - k(T_i - T_{i-1}) \quad (16)$$

the right part of the equation denotes the second-order difference, which can be extended from discrete space to continuous space as a derivative. The one-dimensional heat conduction equation is expressed as follows:

$$\frac{dT}{dt} - k \frac{\partial^2 T}{\partial x^2} = 0 \quad (17)$$

where k represents the material thermal diffusivity in high-dimensional Euclidean spaces. The first-order and the second-order derivatives are generalized to the gradient, and the Laplace operator, respectively.

$$\frac{dT}{dt} - k \Delta T = 0 \quad (18)$$

The symbol Δ represents the summation of second-order derivatives for each coordinate.

We analyze the one-dimensional heat conduction model using graph theory, which extends heat conduction to topological spaces for analysis. Each node on the graph only exchanges heat with adjacent (connected) nodes, as shown in Fig. 2. For example, Node 1 exchanges heat only with Nodes 0, 2, 3, and 4, while heat transfer to Node 5 must go through Node 4. We assume that the rate of heat flow still follows Newton's law of cooling. The temperature change at time t for the node i can be described by the following equation:

$$\frac{dT_i}{dt} = k \sum_j A_{ij} (T_i - T_j) \quad (19)$$

The equation includes the adjacency matrix A mentioned earlier. It is worth noting that for each element A_{ij} in this matrix, if node i and j are adjacent, then $A_{ij} = 1$, otherwise $A_{ij} = 0$. This equation precisely represents that only adjacent edges can produce heat exchange with node i and the heat input is proportional to the temperature difference. Further derivation of the equation yields:

$$\frac{dT_i}{dt} = k [T_i \sum_j A_{ij} - \sum_j A_{ij} T_j] = k [\deg(i) T_i - \sum_j A_{ij} T_j] \quad (20)$$

The notation $\deg(\bullet)$ is the node degree, defined by the edges number connected to the node. The expression $\sum_j A_{ij} T_j$ can be understood as the inner product between the i^{th} row of the adjacency matrix and a vector composed of the temperatures of all nodes. When considering all nodes,

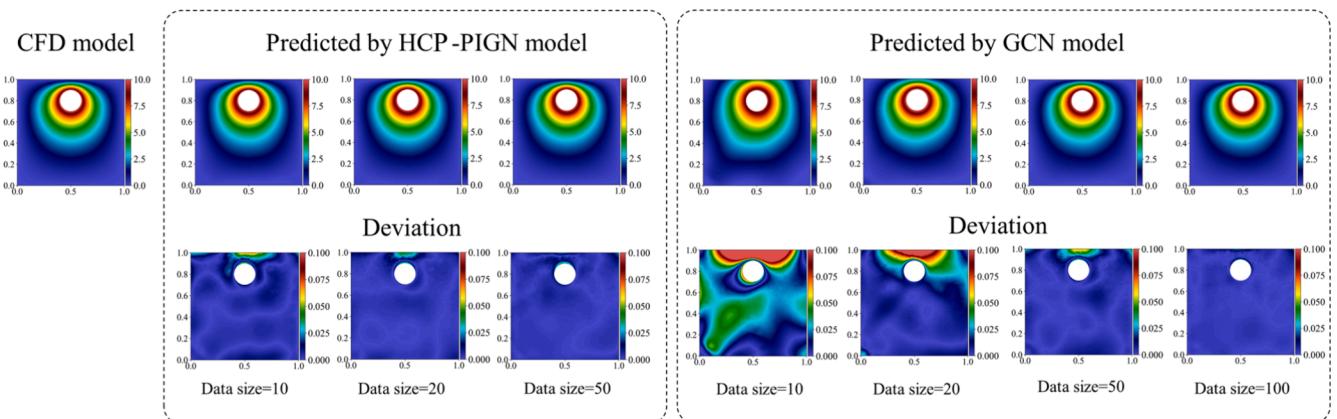


Fig. 16. The prediction performance of HCP-PIGN and GCN, and their comparison with CFD results, under different training data sizes.

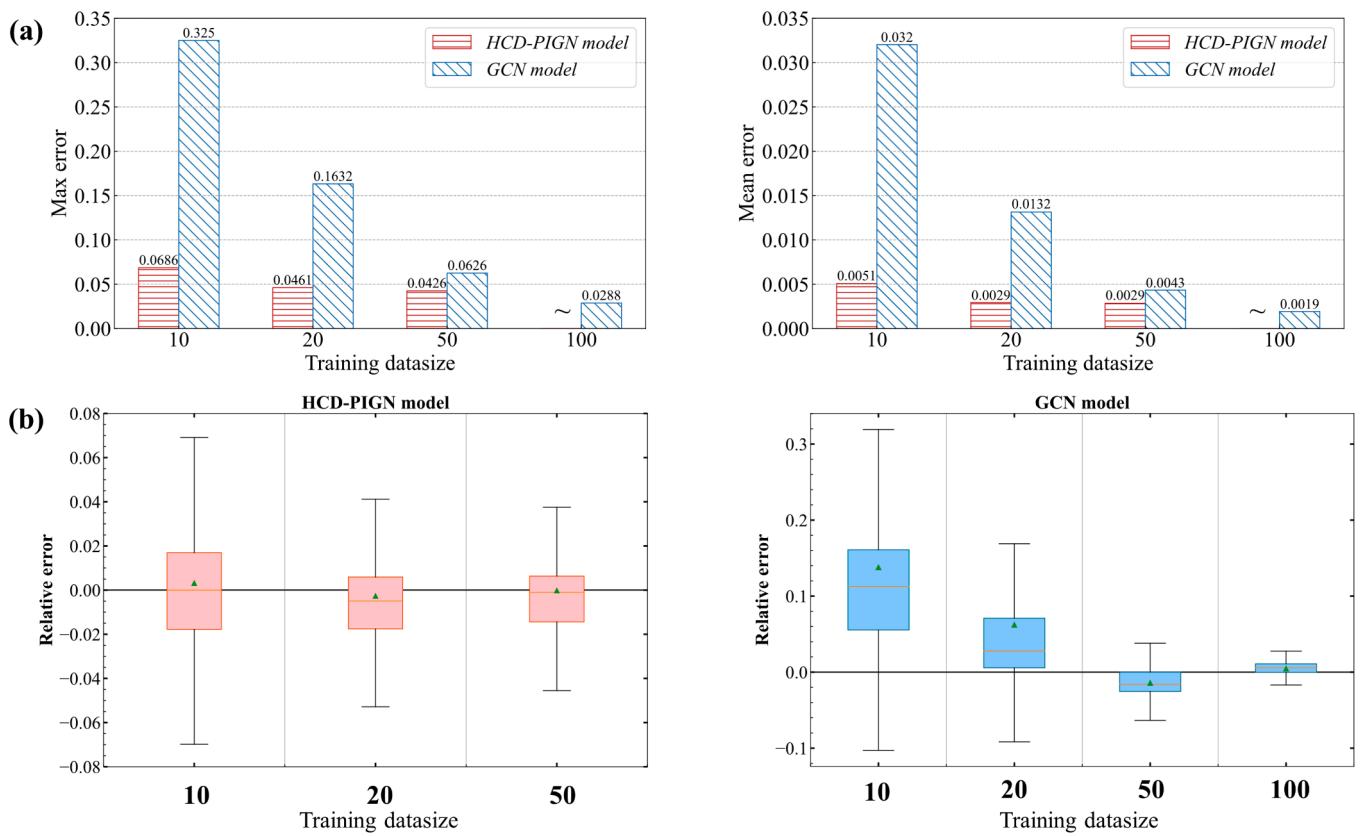


Fig. 17. Comparison of the prediction error for the HCP-PIGN and GCN models, under different training data sizes. (a) Comparison of maximum error and average error. (b) Comparison of relative error.

we use D to represent the degree matrix composed of all nodes, and define the vector $\mathbf{T} = [T_1, T_2, \dots, T_n]$. Thus, we have:

$$\frac{dT}{dt} = kDT - kAT = k(D - A)\mathbf{T} \quad (21)$$

where $(D - A)$ can be denoted by \tilde{A} , which is the Combinatorial Laplacian matrix, substituting \tilde{A} into the above equation, we can observe that the differential equation in graph theory has the same form as that of continuous Euclidean space.

The difference is that graph theory transforms the problem from continuous distribution in Euclidean space to finite nodes in topology space. If we regard the heat flow between each node in the graph as features, and also consider the heat flow features that are transmitted to the nodes in multiple directions, then the problem can be extended from one to two dimensions and solved by graph convolutional networks. The essence of GCN is to describe the propagation and flow of features and messages in a graph. Through the transformation of the problem into a graph's topological structure, the obtained graph data preserves the inherent mesh characteristics, including the refinement and sparsity regions of the original grid. As a result, the accuracy of the original data is maintained.

Based on the computational characteristics of graph neural networks and the constraints of physical information, the framework proposed in this paper offers a robust new paradigm for solving problems related to partial differential equations, and achieves excellent prediction performance, both for solving the Burger's equation in one dimension and heat conduction problem in two dimensions. Given the scalability of the model, and from the perspective of computational mechanics, it will provide effective references for solving physical problems such as incompressible or compressible Navier-Stokes equations, inviscid Euler equations, structural responses, and constitutive relations.

5. Conclusion

This paper aims to propose and investigate the temperature field prediction by the graph neural networks embedded with physical information under geometric position variations. The one-dimensional Burgers' equation and two-dimensional steady-state heat conduction problem is investigated. The model was primarily constructed using graph convolution operations and a loss function informed by heat transfer equations. The traditional physics-informed convolution network used pixilation techniques to preprocess data, but graph-based physics models performed direct learning of the inherent fluid dynamic properties from unstructured grids, which enabled to include the nodes connections and spatial properties.

For the HCP-PIGN prediction performance, two types of models are constructed, i.e., graph neural networks and fully connected neural networks, which are trained by pure data-driven and physics-informed methods, respectively. During the tests, independent data groups and the same hyperparameters of the models were selected to assess the model performance. The scalability and feasibility of HCP-PIGN for temperature distribution prediction is verified over non-uniform fields adaptive to geometric changes. Furthermore, restrained by the governing equation, the HCP-PIGN model is able to calculate the PDE even under limited training data size. The computational time costs are thus greatly saved for dataset generation and pre-processing, which allows the HCP-PIGN model to significantly improve the processing efficiency of repetitive tasks in large-scale computational tasks.

Author contributions

Jiang-Zhou Peng performed the numerical simulation, analyzed results and completed the initial manuscript draft. Mei Mei and Yue Hua supported the work, developed and improved the paper. Nadine Aubry,

Yu-Bai Li, and Zhi-Hua Chen supervised the work and revised the manuscript. All authors contributed to the manuscript preparation. All authors have read and agreed to the published version of the manuscript.

CRediT authorship contribution statement

Jiang-Zhou Peng: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation. **Nadine Aubry:** Writing – review & editing, Methodology, Investigation, Formal analysis. **Yu-Bai Li:** Writing – review & editing, Investigation, Formal analysis. **Zhi-Hua Chen:** Writing – review & editing, Methodology, Formal analysis. **Mei Mei:** Writing – review & editing, Validation, Supervision, Methodology, Investigation. **Yue Hua:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work is supported by Remarkable Distinguished Postdoctoral Program of Jiangsu, Key Laboratory of Thermal Management and Energy Utilization of Aircraft, Ministry of Industry and Information Technology (Grant No. CEPE2022016), State Key Laboratory of Mechanics and Control for Aerospace Structures (Nanjing University of Aeronautics and Astronautics) (Grant No. MCMS-E-0323Y01).

References

- Bararnia, H., Esmaeilpour, M., 2022. On the application of physics informed neural networks (PINN) to solve boundary layer thermal-fluid problems. *Int. Commun. Heat Mass Transf.* 132, 105890 <https://doi.org/10.1016/j.icheatmasstransfer.2022.105890>.
- A. G. Baydin and B. A. Pearlmutter, “Automatic differentiation of algorithms for machine learning,” *arXiv Prepr. arXiv1404.7456*, 2014.
- Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2018. Automatic differentiation in machine learning: a survey. *J. Machine Learn. Res.* 18, 1–43.
- Bodapati, J.D., Veeranjaneyulu, N., 2019. Feature extraction and classification using deep convolutional neural networks. *J. Cyber Secur. Mobil.* 261–276.
- Bollapragada, R., Nocedal, J., Mudigere, D., Shi, H.-J., Tang, P.T.P., 2018. A progressive batching L-BFGS method for machine learning. In: International Conference on Machine Learning, pp. 620–629.
- Bruch Jr, J.C., Zyyoloski, G., 1974. Transient two-dimensional heat conduction problems solved by the finite element method. *Int. J. Numer. Methods Eng.* 8 (3), 481–494.
- Cai, S., Wang, Z., Wang, S., Perdikaris, P., Karniadakis, G.E., 2021. Physics-informed neural networks for heat transfer problems. *J. Heat Transfer* 143 (6).
- Du, Y.-L., et al., 2023. Advances in intellectualization of transportation infrastructures. *Engineering* 24, 239–252.
- Edalatifar, M., Tavakoli, M.B., Ghalambaz, M., Setoudeh, F., 2021. Using deep learning to learn physics of conduction heat transfer. *J. Therm. Anal. Calorim.* 146 (3), 1435–1452. <https://doi.org/10.1007/s10973-020-09875-6>.
- Fic, A., Bialecki, R.A., Kassab, A.J., 2005. Solving transient nonlinear heat conduction problems by proper orthogonal decomposition and the finite-element method. *Numer. Heat Transf. Part B Fundam.* 48 (2), 103–124.
- Gao, X.-W., et al., 2017. Element differential method for solving general heat conduction problems. *Int. J. Heat Mass Transf.* 115, 882–894.
- Gao, H., Sun, L., Wang, J.-X., 2021. PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *J. Comput. Phys.* 428, 110079.
- Gao, T., Zhang, W.H., Zhu, J.H., Xu, Y.J., Bassir, D.H., 2008. Topology optimization of heat conduction problem involving design-dependent heat load effect. *Finite Elem. Anal. Des.* 44 (14), 805–813.
- Han, H., Ingham, D.B., Yuan, Y., 1995. The boundary element method for the solution of the backward heat conduction equation. *J. Comput. Phys.* 116 (2), 292–299.
- He, Z., Ni, F., Wang, W., Zhang, J., 2021. A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials. *Mater. Today Commun.* 28 (June), 102719 <https://doi.org/10.1016/j.mtcomm.2021.102719>.
- Hua, Y., Yu, C.-H., Zhao, Q., Li, M.-G., Wu, W.-T., Wu, P., 2023. Surrogate modeling of heat transfers of nanofluids in absorbent tubes with fins based on deep convolutional neural network. *Int. J. Heat Mass Transf.* 202, 123736.
- Hua, Y., Wang, Z.-Q., Yuan, X.-Y., Li, Y.B., Wu, W.-T., Aubry, N., 2023. Estimation of steady-state temperature field in Multichip Modules using deep convolutional neural network. *Therm. Sci. Eng. Prog.*, 101755.
- Jagtap, A.D., Kharazmi, E., Karniadakis, G.E., 2020. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Comput. Methods Appl. Mech. Eng.* 365, 113028.
- Jin, X., Cai, S., Li, H., Karniadakis, G.E., 2021. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.* 426, 109951.
- Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. *Nat. Rev. Phys.* 3 (6), 422–440.
- Kim, P., Kim, P., 2017. Convolutional neural network. *MATLAB Deep Learn. Mach. Learn. Neural Networks Artif. Intell.* 121–147.
- T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv Prepr. arXiv1609.02907*, 2016.
- Li, Y., Liu, T., Xie, Y., 2022. Thermal fluid fields reconstruction for nanofluids convection based on physics - informed deep learning. *Sci. Rep.* 1–23. <https://doi.org/10.1038/s41598-022-16463-1>.
- Li, K., Shan, S., Zhang, Q., Cai, X., Zhou, Z., 2022. a Computational method to solve for the heat conduction temperature field based on data-driven approach. *Therm. Sci.* 26 (1), 233–246. <https://doi.org/10.2298/TSCI200822165L>.
- Li, W., Yu, B., Wang, X., Wang, P., Sun, S., 2012. A finite volume method for cylindrical heat conduction problems based on local analytical solution. *Int. J. Heat Mass Transf.* 55 (21–22), 5570–5582.
- Lin, Q., Hong, J., Liu, Z., Li, B., Wang, J., 2018. Investigation into the topology optimization for conductive heat transfer based on deep learning approach. *Int. Commun. Heat Mass Transf.* 97, 103–109.
- Liu, Y., Dinh, N., Sato, Y., Niceno, B., 2018. Data-driven modeling for boiling heat transfer: Using deep neural networks and high-fidelity simulation results. *Appl. Therm. Eng.* 144 (May), 305–320. <https://doi.org/10.1016/j.applthermaleng.2018.08.041>.
- Liu, Y., Chen, L., Li, Z., Du, J., 2022. On the global optimum for heat conduction. *Int. J. Heat Mass Transf.* 198, 123381.
- Lohan, D.J., Dede, E.M., Allison, J.T., 2020. A study on practical objectives and constraints for heat conduction topology optimization. *Struct. Multidiscip. Optim.* 61 (2), 475–489.
- H. Ma, X. Hu, Y. Zhang, N. Thuerey, and O. J. Haidn, “A combined data-driven and physics-driven method for steady heat conduction prediction using deep convolutional neural networks,” *arXiv Prepr. arXiv2005.08119*, 2020.
- Mishra, S.C., Mondal, B., Kush, T., Krishna, B.S.R., 2009. Solving transient heat conduction problems on uniform and non-uniform lattices using the lattice Boltzmann method. *Int. Commun. Heat Mass Transf.* 36 (4), 322–328.
- Morningstar, W.R., et al., 2019. Data-driven reconstruction of gravitationally lensed galaxies using recurrent inference machines. *Astrophys. J.* 883 (1), 14.
- Peng, J.Z., Liu, X., Aubry, N., Chen, Z., Wu, W.T., 2021. Data-driven modeling of geometry-adaptive steady heat conduction based on convolutional neural networks. *Case Stud. Therm. Eng.* 28 (November), 101651 <https://doi.org/10.1016/j.csite.2021.101651>.
- Peng, J.Z., Aubry, N., Li, Y.B., Mei, M., Chen, Z.H., Wu, W.T., 2023. Physics-informed graph convolutional neural network for modeling geometry-adaptive steady-state natural convection. *Int. J. Heat Mass Transf.* 216 (July), 124593 <https://doi.org/10.1016/j.ijheatmasstransfer.2023.124593>.
- Peng, J.Z., Hua, Y., Li, Y.B., Chen, Z.H., Wu, W.T., Aubry, N., 2023. Physics-informed graph convolutional neural network for modeling fluid flow and heat convection. *Phys. Fluids* 35 (8). <https://doi.org/10.1063/5.0161114>.
- M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations,” *arXiv Prepr. arXiv1711.10561*, 2017.
- M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arXiv e-prints, p.,” *arXiv Prepr. arXiv1711.10566*, 2017.
- Rajabi, M.M., et al., 2022. Analyzing the efficiency and robustness of deep convolutional neural networks for modeling natural convection in heterogeneous porous media. *Int. J. Heat Mass Transf.* 183, 122131.
- Tadeparti, S., Nandigana, V.V.R., 2022. Convolutional neural networks for heat conduction. *Case Stud. Therm. Eng.* 38, 102089.
- Zeng, J., Jia, X., 2024. Systems Theory-driven framework for AI integration into the holistic material basis research of traditional Chinese medicine. *Engineering*.
- Zobeiry, N., Humfeld, K.D., 2021. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Eng. Appl. Artif. Intell.* 101 (September 2020), 104232. <https://doi.org/10.1016/j.engappai.2021.104232>.
- Zobeiry, N., Humfeld, K.D., 2021. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Eng. Appl. Artif. Intell.* 101 (January), 104232 <https://doi.org/10.1016/j.engappai.2021.104232>.