



## Physics-informed graph convolutional neural network for modeling geometry-adaptive steady-state natural convection

Jiang-Zhou Peng<sup>a</sup>, Nadine Aubry<sup>b</sup>, Yu-Bai Li<sup>c</sup>, Mei Mei<sup>d</sup>, Zhi-Hua Chen<sup>a</sup>, Wei-Tao Wu<sup>e,\*</sup>

<sup>a</sup> Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China

<sup>b</sup> Department of Mechanical Engineering, Tufts University, Medford, MA 02155, USA

<sup>c</sup> Key Laboratory of Ocean Energy Utilization and Energy Conservation of Ministry of Education, Dalian University of Technology, Dalian 116023, China

<sup>d</sup> Sino-French Engineer School, Nanjing University of Science and Technology, Nanjing 210094, China

<sup>e</sup> School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China



### ARTICLE INFO

#### Keywords:

Natural convection  
Surrogate model  
Physics-informed neural network  
Graph convolutional neural network

### ABSTRACT

This paper presents a novel deep learning-based surrogate model for steady-state natural convection problem with variable geometry. Traditional deep learning based surrogate models are more or less limited by the requirement for large amounts of training data, loss of accuracy due to pixelization of the original data, the low accurate prediction near the boundaries and low geometry adaptive capability. To overcome the above challenges, the proposed natural convection prediction framework is mainly composed of a physics-informed neural network (PINN) and a graph convolutional neural network (GCN), called natural convection prediction model based on physics-informed graph convolutional network (NCV-PIGN). The GCN serves as the prediction module, inferring and predicting natural convection phenomena by considering the interactions between unstructured nodes and their neighbor; the PINN incorporates the governing equations of natural convection into the loss function of the neural network, allowing the predictions from GCN to satisfy the constraints imposed by the physical laws. The advantages of this framework are twofold: the operation principles of the GCN better align with the development of the temperature field in real situations, and the embedding of physical information strengthens the model's understanding of the flow field, accurately describing the variations of temperature gradients at the boundary positions while reducing the model's reliance on training data. Finally, to validate the superiority of the NCV-PIGN, we analyze its geometric adaptability and accuracy of prediction using single and dual heat source cases. We compare the model's prediction results at different sampling point quantities and contrast them with those of purely data-driven models. The results demonstrate that the excellent geometric adaptability and prediction capability of the proposed model can be achieved with only 20 training data and once the fully trained the model can solve natural convection problems within 3 ms. The max and mean relative errors in predicting the temperature field are less than 2% and 0.4% for both single and dual heat source cases. Compared to the pure data-driven model, the proposed model has reduced the maximum error by 65.5% and the mean error by 72%. These results validate the effectiveness of the developed NCV-PIGN model, enabling better performance of the deep learning-based surrogate models for natural convection problems.

### 1. Introduction

Natural convection is a widely observed heat transfer phenomenon in nature and engineering, with significant implications for optimizing energy utilization, improving engineering design, and addressing environmental issues. The flow behavior in natural convection problems is typically influenced by multiple factors such as temperature gradients, fluid properties, and geometric shapes. In the solution of these problems,

the traditional widely used solvers mainly include Boundary Element Method (BEM) [1], Finite Element Method (FEM) [2], Finite Volume Method (FVM) [3] and other high-accurate Fourier-Galerkin (FG) solution [4], double multi relaxation time Lattice Boltzmann method (Double MRT-LBM) [5] and meshless multiscale element free Galerkin method (VMEFG) [6]. However, when using these solvers, traditional numerical methods often require detailed physical modeling and parameter adjustments for these factors. The fact is that tasks such as

\* Corresponding author.

E-mail address: [weitaowwtw@njust.edu.cn](mailto:weitaowwtw@njust.edu.cn) (W.-T. Wu).

design, parameter estimation, or sensitivity analysis conducted in real applications often require hundreds or thousands of computations, resulting in a substantial consumption of computational resources and simulation time. This limitation restricts the application scope and computational efficiency of numerical methods in practical engineering scenarios. Recently, deep learning techniques have been growing in interest in the fields of fluid mechanics [7] and heat transfer [8]. As a data-driven method, deep learning has the ability to handle complex problems and extract potential from large-scale data. By training on large amounts of data samples, deep neural networks can automatically learn input features and output relationships, extract features and patterns from data, and build a solution model for complex fluid states. This makes deep neural networks more flexible and adaptable in solving physical problems.

Deep learning has fundamentally revolutionized various classical engineering approaches and has provided effective surrogate models for solving natural convection problems [9–11]. Leveraging the potential of deep learning, these models can accelerate computation while maintaining accuracy. Due to the scalability of deep learning technology with large amount of data and hierarchical feature learning, well-trained data-driven models can approximate real temperature fields with high precision. Rajabi et al. [12] utilized an encoder-decoder convolutional neural network architecture to perform end-to-end regression in the domain of natural convection in porous media, using the porous square cavity benchmark. The numerical experiments conducted underscored the robustness and efficiency of the proposed model in handling heterogeneous domains. The authors demonstrated that the developed model yielded satisfactory approximations, effectively addressing the challenge of dimensionality associated with high-dimensional input and output spaces. In the work of Tizakast et al. [13] they explore the application of machine learning models in investigating double-diffusive natural convection in rectangular cavities filled with non-Newtonian fluids. Four dimensionless parameters, including the buoyancy ratio, thermal Rayleigh number, Lewis number, and power-law behavior index, are utilized as input features for the predictive models. The output of the model includes three separate flow characteristics: stream function at the center of the cavity, average Nusselt number, and average Sherwood number, which are used to evaluate the performance of the model. Four machine learning models are employed in the comparative study. The authors evaluated the generalization ability of all models using previously unseen data, and the results confirmed the excellent calibration and high adaptability of these models. Their work validates the promising solution of utilizing machine learning methods for modeling non-Newtonian double-diffusive fluid flow problems. Lu et al. [14] discovered that a neural network with a single hidden layer can accurately approximate any nonlinear and continuous operator. Based on this finding, the authors proposed the DeepONet framework, which learns the operational mechanism of continuous nonlinear operators from data and utilizes fully-trained neural networks to construct novel nonlinear operators, also referred to as deep operators, for fast and complex dynamic predictions. The framework demonstrates excellent generalization performance in learning various linear and nonlinear explicit and implicit operators. In the industrial manufacturing field, Durgam et al. [15] proposed a combination of machine learning algorithms for predicting the temperatures of rectangular silicon heaters with dummy elements. This work explores the performance of combining different machine learning algorithms, specifically comparing the predictive results of Support Vector Machine (SVM) and ensemble Artificial Neural Network (ANN). The results show that there is an agreement of less than 10% between the simulated and predicted temperatures. Additionally, it is observed that SVR outperforms XGBoost and ANN when analyzed individually. This work further illustrates the deep learning method is capable of learning physical problems without knowing the underlying fundamental governing equation. Although data-driven models demonstrate strong predictive capabilities, they have two shortcomings from an application

perspective. Firstly, they require a large amount of data. During the training phase, these models need to be provided with a substantial volume of raw data to learn potential patterns. However, this reliance on a high quantity of training data does not effectively address the computational costs associated with traditional numerical solvers. This creates a loop that fails to overcome the inherent expense of numerical computation [16]. Secondly, data-driven deep learning algorithms lack strong physical interpretability. In many physics and engineering domains, valuable prior knowledge is often embedded within the training data. Unfortunately, this crucial information is not fully captured by purely data-driven deep learning algorithms, limiting their ability to provide meaningful physical insights.

To address these limitations, researchers have started incorporating additional information, such as balance equations and physical laws, into traditional deep learning algorithms. Raissi et al. [17,18] proposed a solution based on Physical Information Neural Network (PINN) to solve partial differential equations by utilizing the balance equation and the residual of boundary conditions to construct a physical information loss function. By minimizing this loss function, the neural network is trained to constrain the prediction results within the physical laws. For instance, Wang et al. [19] made improvements to the DeepONet framework proposed by Lu et al. by incorporating physics-informed neural networks (PINN), resulting in significant enhancements in prediction accuracy, generalization performance, and data efficiency compared to the original framework. Subsequently, based on the Raissi's framework, the PINN approach has been applied to several heat transfer problem such as the work proposed by Cai et al. [20]. They developed a data-driven strategy based on PINNs to infer the temperature field in the prototypical problem of heat convection in the flow past a cylinder. The authors had access to temperature measurements on both the cylinder surface and the wake region, but the thermal boundary conditions on the cylinder surface were completely unknown. Using PINN, the velocity and temperature fields at the sampling points were computed, and then the Navier-Stokes and energy equations were enforced on the state quantities at these sampling points. Through training the PINN, the model not only discovered the previously unknown boundary conditions but also accurately predicted the temperature field throughout the entire calculation domain with a negligible error. In Laubscher's work [21], the effectiveness of PINNs in solving the forward problem of steady-state multi-species convective and diffusive flow, as well as heat transfer, was assessed through a simple case study. The author employed both single-network and segregated-network PINN architectures to predict the momentum, species, and temperature distributions in a two-dimensional rectangular domain for a heat transfer problem. The prediction results of both models were compared against the results obtained from the OpenFOAM solver, and both models demonstrated a high level of accuracy. Notably, the segregated-network PINN method outperformed the single-network PINN, achieving an average reduction in loss of 62%. Bararnia and Esmailipour [22] conducted a study on the application of PINNs for solving boundary layer thermal-fluid problems. Their approach involves formulating a loss function that ensures the satisfaction of both the boundary conditions and the governing equations within the network. The model was evaluated through testing on three benchmark problems, namely Blasius-Pohlhausen, Falkner-Skan, and Natural convection. The results of these tests demonstrated reasonable prediction accuracy. Collectively, these studies highlight the suitability of PINN for systems governed by partial differential equations (PDEs). In summary, PINN offers several advantages over purely data-driven models. By incorporating the partial differential form of physical laws into machine learning, PINN not only learns the data patterns but also captures the underlying physical laws. Furthermore, since PINN is trained to satisfy the governing equations, it can perform well even in regions where no observable data is available. However, PINN models do have limitations. The two primary limitations are the training cost and the scalability of the models [23]. The point-wise formulation of PINN involves a large number of automatic

differentiation operations, leading to the generation of large tensors during the back-propagation process [24]. In the literature [25] and [26], studies have investigated the impact of the number of collocation points in the computational domain on PINN training. The results indicate that increasing the number of collocation points can enhance the model performance. This, in turn, requires more memory on the Graphics Processing Unit (GPU) and increases the training time of PINN [27]. In addition, if slight modifications are made to the inference problem, a well-trained network may struggle to generalize to solving new problems. For instance, in most existing PINN frameworks, fully connected neural networks (FNNs) are commonly used as the model structure [28–30]. Such structures take arbitrary distributed data points as input, which do not explicitly incorporate geometric features that may exist in the data. In other words, they only compute the gradients based on the input point set itself, overlooking the spatial structure formed by the point set. As a result, PINN models can be specifically trained for fixed shapes or geometries. However, when the problem changes, the PINN needs to be retrained to solve the new problem.

To enhance the scalability of the model and improve its sensitivity to changes in geometry, it is necessary to incorporate both the spatial coordinates of a data point and the geometric properties of the computational space into the neural network. This can be effectively achieved by utilizing convolutional neural networks (CNNs) [31], which employ weight-sharing network structures [32] and can directly capture spatial features from the raw data. Several CNN-based frameworks for geometric adaptation have been developed, leveraging these advantages, such as [12,33–35]. However, it should be noted that these approaches are purely data-driven models, and the integration of PINNs for geometric adaptation has not been extensively explored. This is primarily due to the requirement of traditional CNNs to convert input data into regular matrix form Morningstar et al. [36], which can lead to the loss of coordinate information and hinder the direct coupling of differential equation residuals. In order to remedy the above mentioned shortcomings, we propose a novel PINN framework that leverages graph convolutional network (GCN) to solve natural convection (abbreviated as NCV-PIGN) with non-uniform grids. The GCN was proposed by Kipf and Welling [37], considers both vertex features and edge properties in its computation of graph data. Unlike traditional convolutional neural networks, GCN does not require pixelated preprocessing of the data, thus preserving the linking relationships between nodes. Furthermore, the input of GCN includes the spatial coordinate information of the observation points, enabling direct calculation of spatial derivatives when solving PDEs and enhancing the prediction accuracy of the model. Compared to traditional physics-informed neural networks, our framework also incorporates a slight data-driven approach. The training of the NCV-PIGN model not only utilizes physical laws to constrain the parameter updates of the neural network but also takes into account the deviations between network predictions and numerical simulations. This training approach has been validated in the research conducted by Rezaei et al. [38], where they demonstrated that incorporating a mixture of data-driven and physics-informed terms in the loss function accelerated the model training process. Furthermore, the authors confirmed that introducing physics-informed equations into the loss function effectively improved the model's generalization capability. This integration of physics-based constraints enhances the model's ability to capture underlying physical phenomena while harnessing the power of data-driven learning.

To evaluate the effectiveness and accuracy of the proposed Non-uniform Grid-based Physics-Informed Graph Neural Network (NCV-PIGN) model for geometric adaptation, a study on the natural convection problem of heat sources located at various positions in the computational domain was conducted. The performance of the model was analyzed by comparing its prediction results with numerical calculations. Furthermore, an error analysis was performed to compare the proposed model with the purely data-driven GCN model. The rest of this paper is structured as follows: in Section 2, we describe our approach in

detail, including dataset preparation and training algorithms, and in Sections 3 and 4 we present and discuss results on test cases where the trained network model predicts. Finally, we provide a summary of the current work.

## 2. Method

In this section, we provide a detailed exposition of our research, focusing on three key aspects: the selection and generation of training data, the structure of neural network, and the training method. Our proposed model aims to predict thermal flow fields and comprises two main components: a graph convolutional network and a physics-informed neural network. The graph convolutional network is mainly used to identify and analyze geometric structure features, so that the model has the ability to adapt to geometric position changes. Furthermore, the physics-informed learning strategy is formulated by constructing a PDE-based loss function in the model. As training progresses, the loss function encapsulating PDE residuals gradually converges, resulting in significantly improved prediction result that is closely aligned with the underlying physical equations.

### 2.1. Mathematical concept

The cases examined in this paper involve a steady-state two-dimensional natural convection problem. The problem is set within a bounded annular computational domain and involves the presence of a circular heat source that can both move and deform. Both the inner and outer walls of the annular enclosure are maintained at constant temperatures. Additionally, we simplify the governing equations for fluid flow and heat transfer by employing the boussinesq approximation, which considers density variations only in the buoyancy term. Furthermore, we assume that the fluid inside the annular enclosure is an incompressible Newtonian fluid. Hence, the physics law of steady-state natural convection in the computational domain can be expressed by the continuity equations, momentum equations, and the energy equations [39]:

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \\ \left( u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) + \frac{\partial p}{\partial x} - Pr \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) &= 0 \\ \left( u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) + \frac{\partial p}{\partial y} - Pr \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - Ra \Pr T &= 0 \\ u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} - \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) &= 0 \end{aligned} \quad (1)$$

$$\text{Boundary conditions : } T(x_i, y_i) = 300 \text{ }^\circ\text{C}; x_i, y_i \in \phi_{outer},$$

$$T(x_i, y_i) = 310 \text{ }^\circ\text{C}; x_i, y_i \in \phi_{source}$$

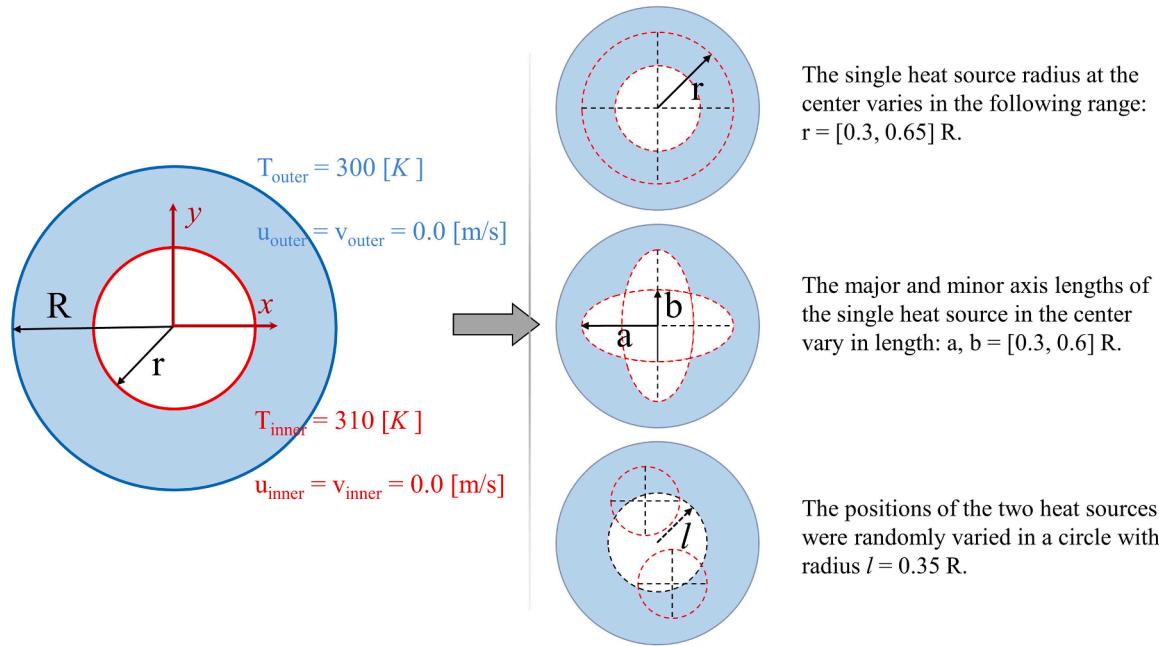
where  $p$  denotes pressure,  $\rho$  is density,  $u$  and  $v$  represent velocity in the  $x$  and  $y$  direction, respectively.  $T(x, y)$  is the solution of the temperature in the computational domain,  $\phi_{outer}$  and  $\phi_{source}$  represent the outer circular and heat source of the computational domain, respectively. Here  $Pr$  is Prandtl number and  $Ra$  is the Rayleigh number, which are defined as.

$$Pr = \frac{v}{\alpha}, \quad Ra = \frac{g\beta(T_{hot} - T_{cold})R^3}{v\alpha} \quad (2)$$

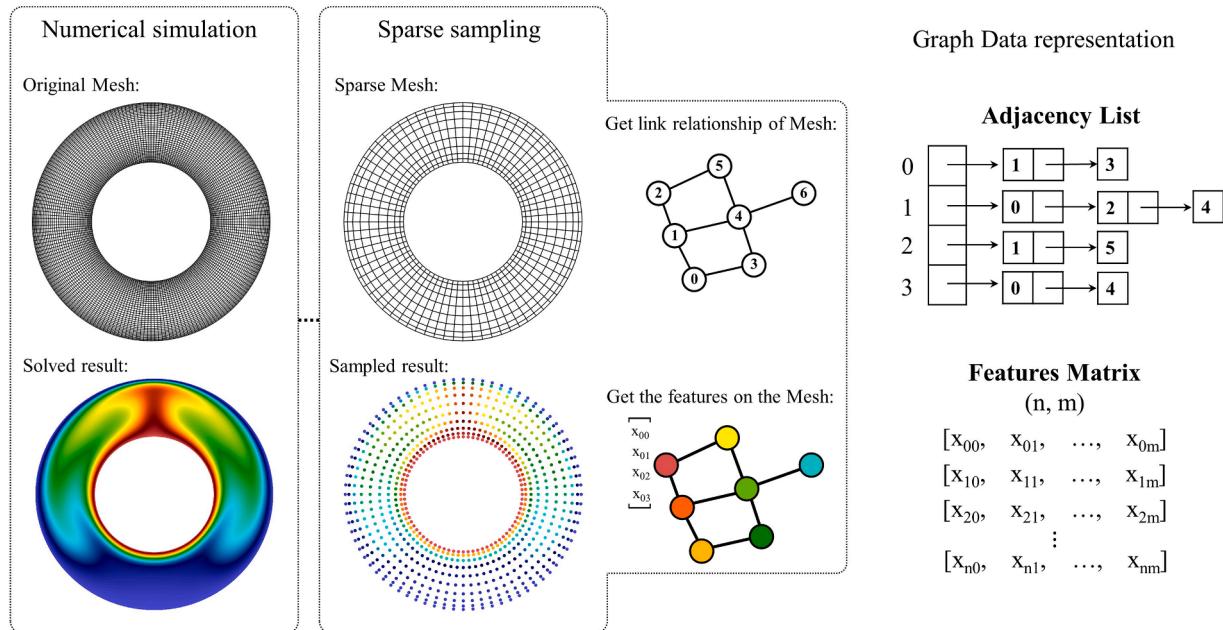
where  $R$  is the characteristic length of the studied object,  $g$  is the gravitational acceleration,  $v$  is kinematic viscosity,  $\alpha = k/\rho c$  is the thermal diffusivity of fluid;  $k, c$  are thermal conductivity and specific heat, and  $\beta$  is the volumetric coefficient. The dimensionless form of the governing equation is made as follows:

$$T^* = \frac{T - T_{cold}}{T_{hot} - T_{cold}}, \quad x^* = \frac{x}{R}, \quad y^* = \frac{y}{R}, \quad u^* = \frac{uR}{\alpha}, \quad v^* = \frac{vR}{\alpha}, \quad p^* = \frac{pR^2}{\rho\alpha^2} \quad (3)$$

### The basic mathematical model



**Fig. 1.** Boundary conditions setting for the Physical field and Training set generation methods.



**Fig. 2.** The generation process from mesh data to graph data, and the graph data composed of adjacency matrix and feature matrix.

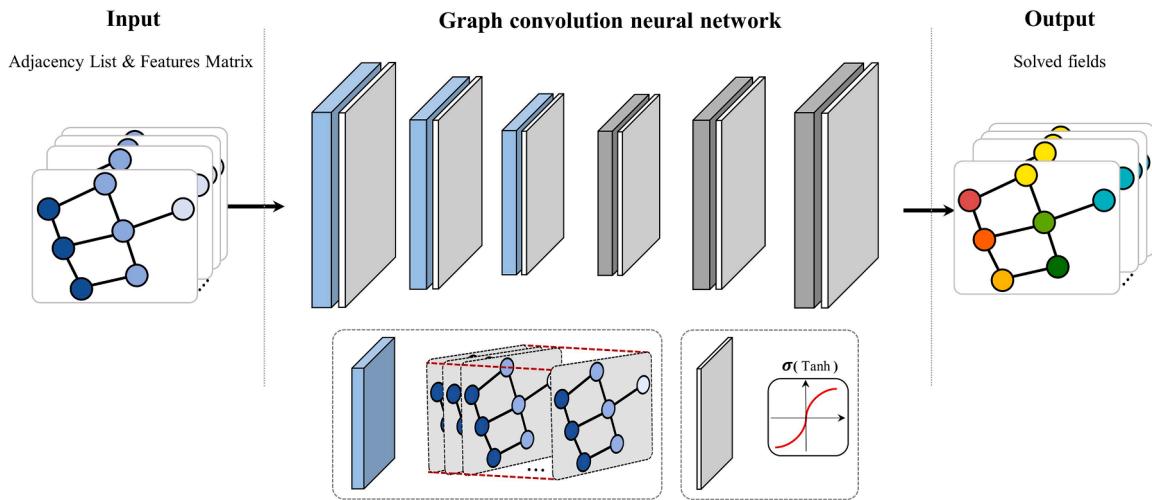
where  $*$  represents the non-dimensional variable, in order to simplify the notation in the subsequent descriptions, we omit the asterisk symbol ( $*$ ). Based on the mentioned above, we have presented the entire physical field in the Fig. 1. Moreover, the figure also illustrates the types of test cases used in this study for training the model and their generation methods.

#### 2.2. Data pre-processing

The graph is represented as  $G = (V, E)$ , where  $V (v \in V, i \in N)$  and  $E ((v_i, v_j) \in E, i, j \in N)$  denote nodes and edges of the graph, respectively. The correspondence between the graph and the grid can be expressed as

follows: the nodes of the mesh are described as vertices  $V$ , the connectivity of the mesh is represented by the edges  $E$ , the neighbors of node  $v_i$  are denoted as the set  $\mathcal{N}(v_i)$ . The process of generating graph data is illustrated in Fig. 2.

From left to right, the figure displays the numerical computation results, the sparse distribution of sampling points used for model training, and the storage format of graph data in the computer. As can be seen from the figure, the graph data indicates a complex relationship between points and points. To efficiently implement the graph model calculation, we selected a sparse distribution of nodes from the dense mesh. Subsequently, the graph data is commonly represented in the form of an adjacency list. In the adjacency list, each node in the graph



**Fig. 3.** The structure of the graph convolutional network.

stores information about all other nodes connected to it. Additionally, taking into account the feature information associated with each node, we establish a feature matrix to store these features. In this study, we preprocess the mesh data by converting the node connections into graph data to better suit the structural characteristics of graph data, which enables us to train the model effectively. We utilize the gmsh library in Python to generate the mesh data. The grids are structured but non-uniformly distributed. We set denser grids near the boundaries to accurately capture boundary layer effects. Subsequently, we introduce probes within the mesh to capture physical information within the flow field, treating these probes as new nodes. We extract the indices of these probe nodes and their adjacent nodes to construct the adjacency list for the graph data. The coordinate information captured by the probes within the computational domain, along with the corresponding physical information, is combined to create the feature matrix and training labels, respectively. The features matrix  $X \in R^{n \times m}$  contains the coordinates of all  $n$  nodes, with each node having  $m$  features. The physical results are used as labels for model training. One important point to emphasize regarding the mesh is that we employ a relatively large number of nodes in the numerical calculations to ensure higher accuracy and provide high-quality labels. Conversely, for the generation of the adjacency list and probes in the same case, we use a smaller number of nodes in order to reduce the training cost of the neural network.

After completing the preprocessing of the raw data following this approach, the data is divided into two parts for evaluating the model's performance during training: the training set and the test set. The training set consists of 20 cases, each containing heat sources with different sizes, shapes, and positions. These cases are used to train the model and enable it to possess the predictive capability of adapting to geometric transformations. The test set, on the other hand, is used to evaluate the model's performance (including prediction accuracy, robustness, generalization, etc.) after the training process. It is an independent dataset separate from the training set.

### 2.3. Network architectures

Our main objective is to predict the natural convection phenomenon of heat sources at arbitrary positions in a two-dimensional annular domain using neural networks. Considering the involvement of geometric shapes and positional transformations in this process, we employ advanced graph convolutional networks to learn the patterns in the data, enabling the model to adapt to geometric transformations. Additionally, the model incorporates physics-informed neural networks by embedding them into the loss function, thereby enhancing its capability to handle natural convection problems. To address this problem, we subsequently

introduce the methods of graph convolutional neural networks and physics-informed neural networks.

#### 2.3.1. Graph convolution neural network

As a type of neural network which directly operates on a graph structure, GCN has the ability to be applied to non-Euclidean data. The operation process of GCN mainly consists of two parts: the propagation of node features between nodes and the propagation of node features between layers. The basic structure of graph convolutional networks is shown in Fig. 3,

In the GCN network, the information of each node in the next layer is obtained through a weighted sum of the information from the current layer itself and the information from its neighboring nodes. This aggregated information is then passed through a nonlinear transformation. The process of information transfer between layers can be described as follows,

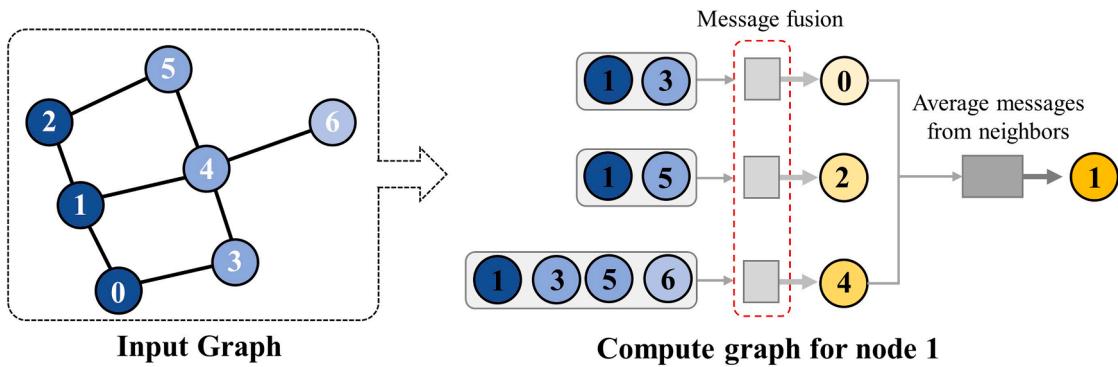
$$\mathbf{H}^{l+1} = \sigma(\mathcal{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathcal{D}^{-\frac{1}{2}}\mathbf{H}^l\mathbf{W}^l) \quad (4)$$

Here,  $\tilde{\mathbf{A}} = \mathcal{D} - \mathbf{A}$  is the Combinatorial Laplacian matrix, which solves the problem of self-transmission (i.e., considers the influence of the node itself on itself).  $\tilde{\mathbf{A}}$  is calculated from the degree matrix  $\mathcal{D}$  and the adjacency matrix  $\mathbf{A}$  (converted by the adjacency list);  $\mathcal{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathcal{D}^{-\frac{1}{2}}$  normalizes the adjacency matrix and balances the influence among nodes;  $\mathbf{H}^l$  represents the hidden feature at the  $l^{\text{th}}$  layer. We set  $\mathbf{H}^0 = \mathbf{X}$  and  $\mathbf{H}^L = \mathbf{Z}$ , where  $L$  denotes the number of layers. Specifically, the input matrix  $\mathbf{X}$  has  $n$  nodes, each with  $m$  input features, and the output matrix  $\mathbf{Z}$  has  $n$  nodes, each with  $c$  output features;  $\mathbf{W}$  is a learnable weight matrix which is continuously updated during the training process.  $\sigma(\cdot)$  denotes an activation function, such as the  $\tanh(\cdot)$ .

The key distinction between GCN and traditional CNN neural networks lies in the *message passing scheme* used to compute the connections between nodes. GCN models the information flow as a process of passing messages from one node to another directly connected node. The *message passing scheme* consists of two stages: the message passing process and the update process. The message passing process can be defined as follows:

$$\mathbf{h}_v^{t+1} = \mathbf{U}_t \left( \mathbf{h}_v^t, \sum_{w \in \mathcal{N}(v)} M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, e_{vw}) \right) \quad (5)$$

where  $\mathbf{h}_v^t$  is the feature vector of node  $v$  at the  $t^{\text{th}}$  iteration,  $\mathbf{U}_t(\cdot), M_t(\cdot)$  is the update function and message function of the node, respectively.  $e_{vw}$  is the edge feature vector between node  $v$  and node  $w$ . The aggregation



**Fig. 4.** Mechanism of message passing and updating between a node ( $v_1$ ) and its neighbors on local graph data.

process and update process of the message transfer mechanism are shown in Fig. 4,

In this process, the blue nodes represent the node features before updating, while the yellow nodes represent the node features after updating. The information propagation can be divided into two steps: aggregating the features of neighboring nodes and updating the node's own features. All nodes simultaneously aggregate the features of their neighboring nodes to obtain updated feature representations. For a node  $v$ , the aggregation and update processes can be described as follows:

$$\begin{aligned} m_v^{(t+1)} &= \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t, e_{vw}) \\ h_v^{(t+1)} &= U_t(h_v^t, m_v^{(t+1)}) \end{aligned} \quad (6)$$

where  $m_v^{(t+1)}$  is the information received by node  $v$  at time step  $t + 1$ , and  $w \in \mathcal{N}(v)$  means that node  $w$  belongs to the neighbor nodes of node  $v$ .

### 2.3.2. Physics-informed network

PINN is a neural network that incorporates physical information into its structure. The model utilizes prior knowledge of the underlying physics to formulate a loss function for the optimization problem. To illustrate the basic network structure and training method of PINN, let's consider the example of solving nonlinear partial differential equations (PDEs). The general form of a nonlinear PDE can be expressed as follows:

$$\begin{cases} \mathbf{u}_t + \mathcal{N}_x[\mathbf{u}] = 0, & x \in \mathbb{R}^d, t \in [0, t_f] \\ u(x, 0) = h(x), & x \in \mathbb{R}^d \\ u(x, t) = g(x, t), & x \in \mathbb{R}^b, t \in [0, t_f] \end{cases} \quad (7)$$

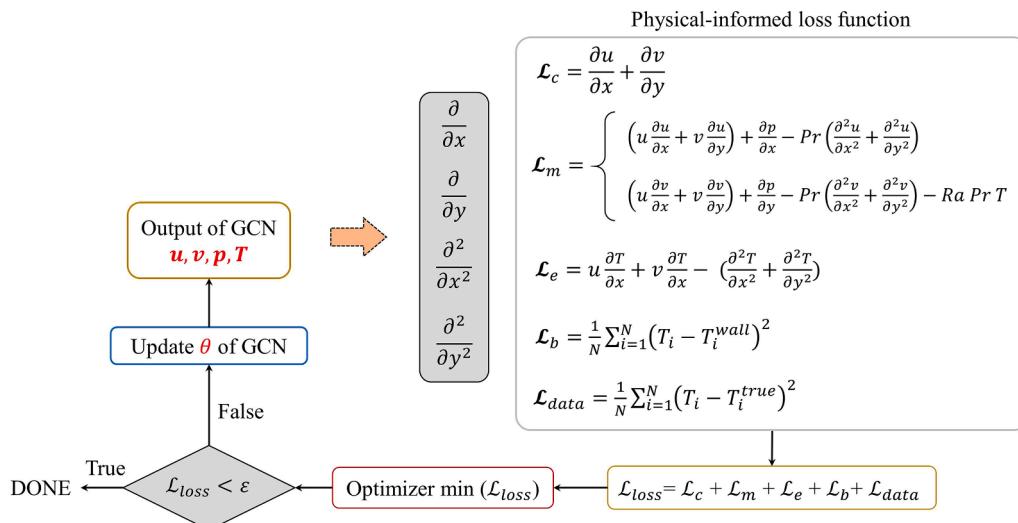
where  $x = (x, y)$  represents the position coordinates of the point,  $\mathcal{N}_x$  represents a general nonlinear differential operator,  $t$  is the temporal coordinate and  $t_f$  is the terminal time,  $\mathbb{R}^d$  and  $\mathbb{R}^b$  denote the computational domain and the boundary, respectively.  $\mathbf{u}(x, t)$  is the solution of the PDEs with initial condition  $h(x)$  and boundary condition  $g(x, t)$ . In PINN, the network uses  $n(x, t; \theta)$  to approximate  $u(x, t)$ . The neural network here can be a fully connected neural network, or a graph neural network, or any other suitable network architecture. PINN integrates the known physical information of the problem (control equations, boundary conditions, etc.) into the loss function. This integration allows PINN to make accurate predictions of the problem's solution even with limited training data. PINN takes the coordinates  $(x, t)$  as inputs and outputs the residual of the PDEs:

$$f(x, t; \theta) = \frac{\partial}{\partial t} n(x, t; \theta) + \mathcal{N}_x[n(x, t; \theta)] \quad (8)$$

The partial derivatives in the aforementioned equation can be computed using the automatic differentiation [24] capability of the neural network. Consequently, the loss function of PINN can be defined as follows:

$$\mathcal{L}(\theta) = \mathcal{L}_f(\theta) + \mathcal{L}_b(\theta) + \mathcal{L}_0(\theta) \quad (9)$$

Here  $\mathcal{L}_f(\theta)$ ,  $\mathcal{L}_b(\theta)$ , and  $\mathcal{L}_0(\theta)$  are denote the loss of governing



**Fig. 5.** Schematic diagram of a physics-informed network embedded with governing equations.

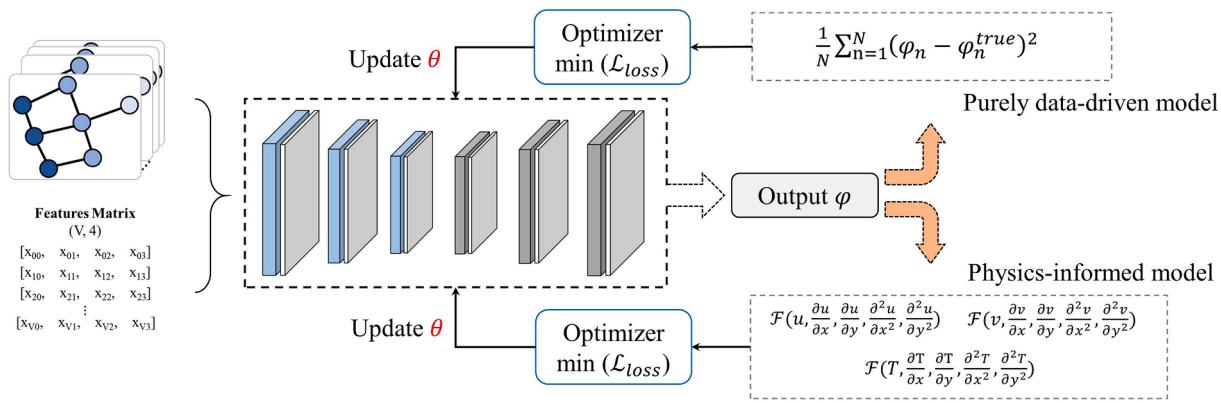


Fig. 6. Graph convolutional network based purely data-driven prediction model and physics-informed prediction model.

equations, the boundary conditions and the initial conditions, respectively. Notice that all loss terms are a function of the network parameters  $\theta$ , which contains the weights and biases of the neurons. A schematic diagram of the proposed physics-informed network is shown in Fig. 5.

Please note that in this paper, we have made certain adjustments to enhance the model's adaptability to geometric variations. To expedite convergence, we have incorporated appropriate label data during the training process. Additionally, since our research focuses on steady-state natural convection, we have omitted the time-related and initial terms from the equation. As a result, the input comprises the state variables ( $u$ ,  $v$ ,  $p$ ,  $T$ ) approximated by the GCN. These variables are constrained by the control equations and undergo automatic differentiation. The obtained loss results are then propagated back to the GCN to update the neural network's weights. It is important to emphasize that this process does not involve any numerical discretization. Further details regarding the training process will be presented in the subsequent subsections.

#### 2.4. Implementation of training

The focus of this paper is on studying a steady-state two-dimensional natural convection process. Moreover, the trained physical neural network is expected to predict the temperature field at various thermal boundary positions within the computational domain. Taking these requirements into account, we have formulated a specific loss term, which can be expressed as follows:

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

where

$$\begin{aligned} \mathcal{L}_f(\theta) &= \frac{1}{\mathcal{Z}_1} \sum_{n=1}^{\mathcal{Z}_1} \left( \frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right)^2, \mathcal{Z}_1 \in \mathbb{R}^d \\ \mathcal{L}_b(\theta) &= \frac{1}{\mathcal{Z}_2} \sum_{n=1}^{\mathcal{Z}_2} (\hat{T} - T)^2, \mathcal{Z}_2 \in \mathbb{R}^b \\ \mathcal{L}_{data}(\theta) &= \frac{1}{\mathcal{Z}_3} \sum_{n=1}^{\mathcal{Z}_3} (\hat{T} - T)^2, \mathcal{Z}_3 \in \mathbb{R}^{label}, \mathbb{R}^{label} \in \mathbb{R}^d \end{aligned} \quad (11)$$

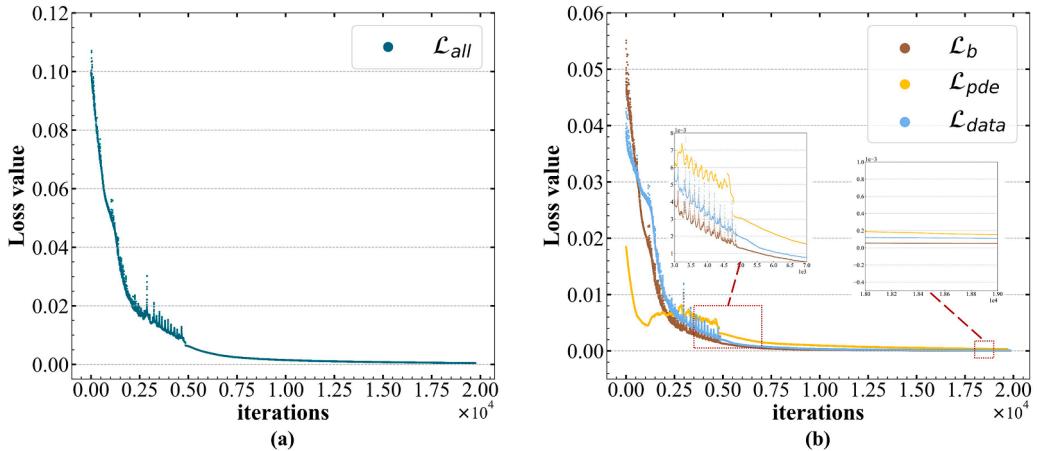
where  $\mathcal{Z}_1$  is the node number in domain  $\mathbb{R}^d$ .  $\mathcal{Z}_2$  represents the number of nodes located on the boundaries  $\mathbb{R}^b$ .  $\mathcal{Z}_3$  represents the number of nodes located in domain  $\mathbb{R}^{label}$ . As the process under consideration is steady-state, the loss function does not take into account the time term or the initial conditions. It is worth mentioning that the approach of adding boundary conditions to the loss equation as used in this study, is commonly referred to as soft-boundaries within the intersectional research field of neural networks and fluid mechanics. Correspondingly, there is a method known as hard-boundaries used to aid the training of

physical neural networks [40]. Hard-boundaries parameterize boundary conditions through pre-training neural networks, and they are often used to represent complex boundary shapes that are difficult to express analytically. However, Considering the regular boundaries used in this paper, excellent predictive performance can be achieved by utilizing only soft-boundaries during training. In addition to training physical neural network, we also trained a purely data-driven neural network for comparison purposes. The training processes for both networks are illustrated in Fig. 6.

For the training data, we placed 120 nodes along the cold wall boundary and the heat source boundary. In the single heat source computational domain, approximately 2000 grid nodes were placed, while 3000 grid nodes were placed in the case of dual heat sources to calculate residuals. The number of labels selected for each case is the number of grid points in the computational domain, and the set is denoted as  $\mathbb{R}^{label}$ . The training data in this paper encompasses various scenarios, including variations in the radius, shape, and relative positions of a single heat source, as well as the same variations for a double heat source. To enable the model to efficiently distinguish between different geometric features during training, each node's features include not only coordinate information but also additional information specific to different scenarios. For single heat source cases, this additional information includes the lengths of the major and minor axes. For double heat source cases, the node features also include the center position information for each heat source. Hence, we divide the training process into two separate models: one primarily trained on data with a single heat source, and the other trained on data with double heat sources. In addition to the different input feature matrices, the working conditions of steady-state convection produced by single heat source and double heat source are also different. Since exploring the adaptation of the model to working conditions is not the focus of this study, we train two separate models to address these variations. It is worth mentioning that, the total loss of the system is optimized with the Adam optimizer, and both models are trained using 5000 iterations with learning rate of 1e-4. Finally, L-BFGS optimizer is used for finetuning [41]. Prior to training the models, we apply min-max normalization to the input coordinates to enhance the convergence speed of the algorithm. Min-max normalization uses the range,  $\max(X) - \min(X)$  in a given dataset  $X$  applied to each training case.

$$\bar{x}_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}, i = 1, 2, \dots, N. \quad (12)$$

Here,  $X$  is the input feature matrix,  $x_i$  denotes the features in each node, and  $\bar{x}$  is the normalized result. After applying the min-max normalization, the training data set values range from 0 to 1. The learning models are constructed using PyTorch, and the network training is conducted on a single Nvidia GeForce RTX 4090 graphics card.



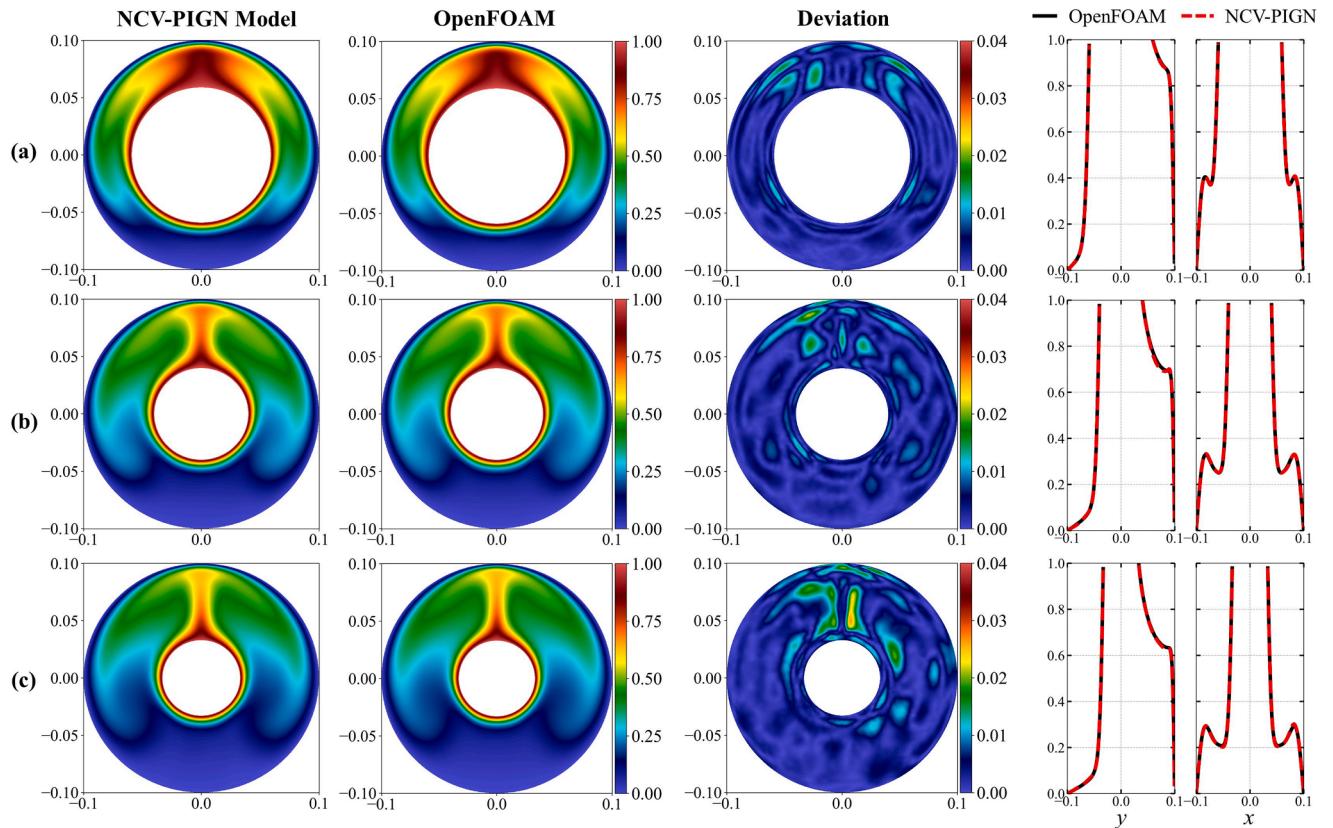
**Fig. 7.** The training convergence process of the NCV-PIGN model,  $\mathcal{L}_b$  represents the loss of the boundary conditions,  $\mathcal{L}_{pde}$  represents the loss of the governing equations (including the continuity equation, momentum equation, and energy equation),  $\mathcal{L}_{data}$  represents the loss of the labeled data, and  $\mathcal{L}_{all}$  is the sum of all loss terms. (a) The convergence history of total loss value; (b) The convergence history of different loss terms.

### 3. Results

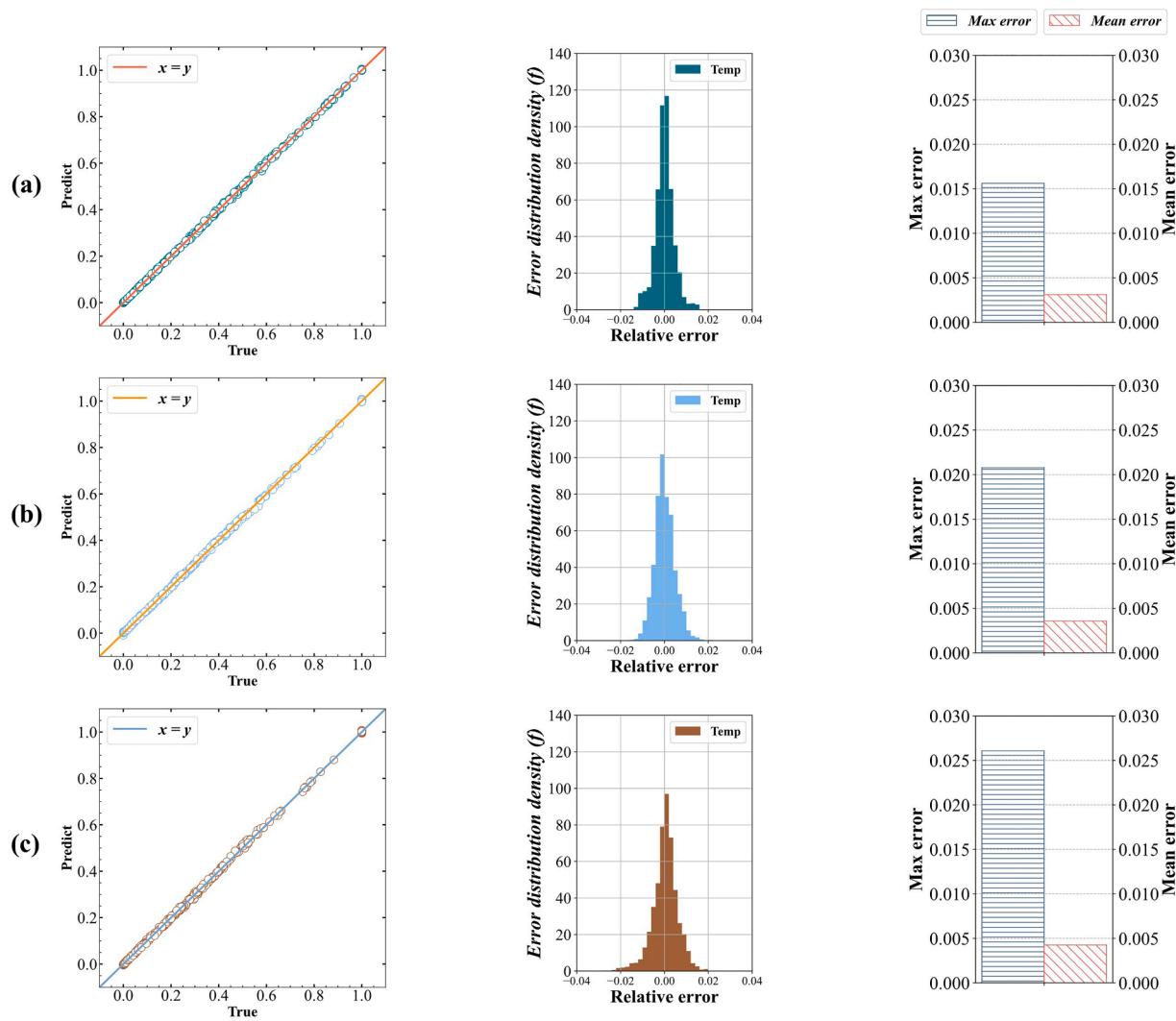
In this section, we investigate the geometric adaptive prediction performance of the model under variations in heat source size, shape, and quantity. The predicted results of the model includes velocity field (with two components: flow rate and vertical flow rate), pressure field, and temperature field, which are all required for the PINN to construct the physical constrain of the governing equations. As this article focuses on heat transfer, for preventing redundant results, in the following we specifically focus on two physical variables: the distribution of temperature in field and the distribution of Nusselt numbers at the outer

circular ring. We compare and analyze the differences between the model's predicted results and the results obtained from numerical computations. Furthermore, we examine the relationship between different spatial node resolutions and the accuracy of the model's predictions. It is important to note that the results presented in this section are obtained from independent test cases, separate from the training set, and their purpose is primarily to validate the model's robustness in geometric adaptation.

Before discussing the model's prediction results, we first present the convergence process of the loss values during the model training. As shown in Fig. 7, we display the convergence process of the loss values for



**Fig. 8.** Comparison of model prediction and numerical calculation results of natural convection temperature field of circle-heat source. (a) Heat source radius is 0.59 R; (b) Heat source radius is 0.4 R; (c) Heat source radius is 0.33 R.



**Fig. 9.** Error analysis of model prediction results of natural convection temperature field of circle-heat source. (a) Heat source radius is 0.59 R; (b) Heat source radius is 0.4 R; (c) Heat source radius is 0.33 R.

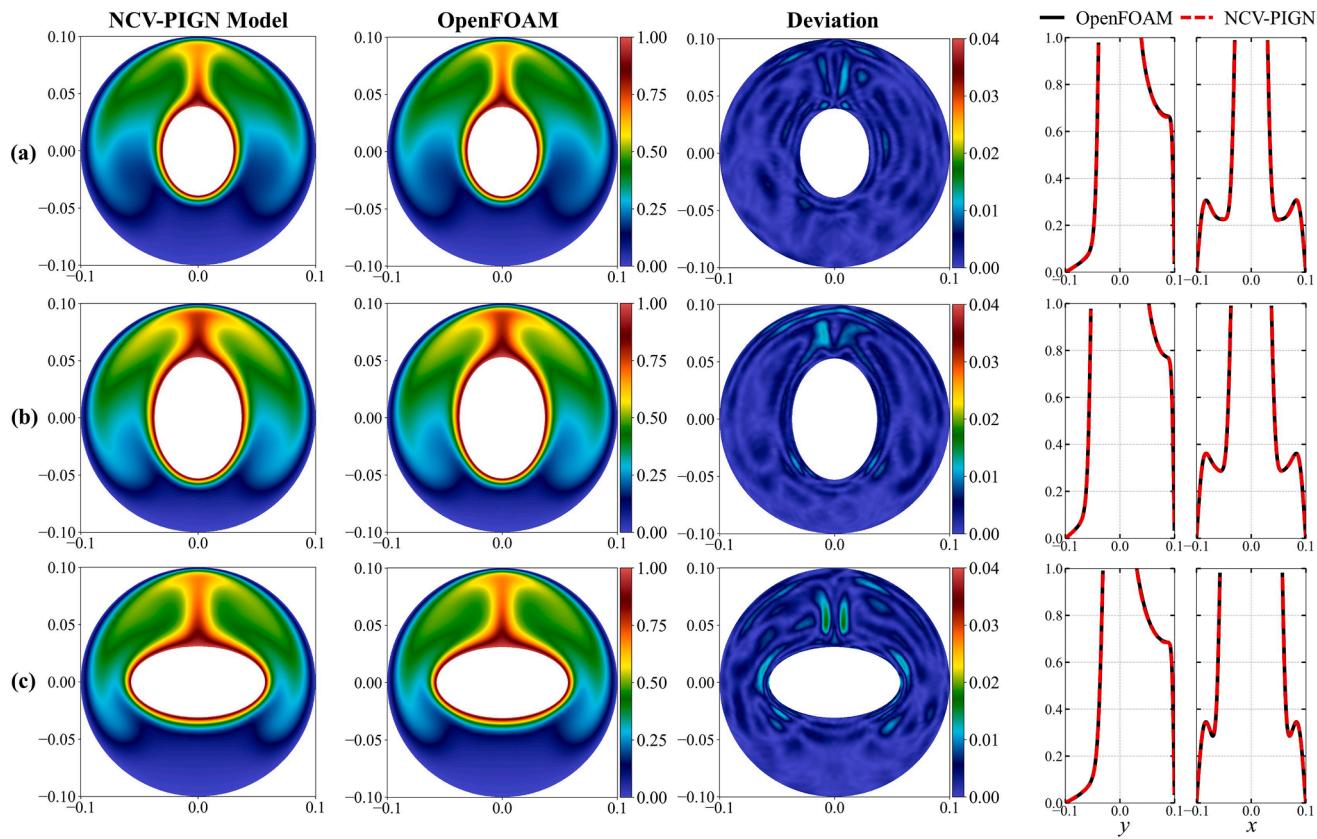
the case of double-heat source geometry, which exhibits relatively complex geometric configuration compared to other cases. It is worth noting that the convergence curves of the loss values remain largely consistent across all cases.

It is important to note that the model training is performed with the Adam optimizer until 5000 steps, and then fine-tuned globally using the L-BFGS optimizer. From the figure, it can be observed that before 5000 steps, there are fluctuations in the loss curves due to changes in the position of the thermal boundary. After 5000 steps, the training enters the global optimization phase, and the loss curves smoothly decrease. Specifically, the loss of the boundary conditions converges quickly because the values on the boundaries are not affected by the thermal boundary in the physical field and do not exhibit complex variations (fixed values). Differently, the loss of the governing equations starts at a relatively low level in the early stages of training but oscillates significantly during the Adam training phase. This phenomenon occurs because the model attempts to find solutions that satisfy different cases and adjusts the parameters of the neural network accordingly. Once the initial node parameters are obtained, the loss rapidly decreases in the L-BFGS training phase as the optimizer searches for the globally optimal node parameters. Towards the end of the training, the loss curves tend to stabilize, indicating that the predicted results satisfy the constraints of the governing equations. The loss of the labeled data starts with higher values in the early stages of training but rapidly decreases as the training

iterations increase. Similar to the governing equations' loss, the loss value decreases steadily when entering the L-BFGS training phase, followed by a stable trend. At the end of the training, the values of all three loss terms were below 2e-4 and nearly converged to similar values. This indicates that the model evenly distributed the contributions of each equation to the total loss value during the training process.

### 3.1. Solving single-heat source field with NCV-PIGN

We first investigate the performance of the proposed framework on predicting the temperature field with single heat source. An interesting research is that we have tried to use the NCV-PIGN model to solve the corresponding physical field for different sizes of heat sources in the computational domain when  $\text{Ra} = 70,000$  and  $\text{Pr} = 0.707$ . The training duration for the model in the single heat source cases is approximately 1 h and 30 min. In addition, comparisons were made with numerical simulation for the purpose of cross-validating the accuracy and validity of the model prediction results. The sizes of the heat sources in the three test cases are [0.59, 0.4, 0.33] R, respectively. The results are shown in Fig. 8. The detailed temperature distributions in the computational domain can be accurately predicted by the model, and the predicted results closely match those obtained from numerical simulations. Our model accurately captures the following phenomena: as the size of the heat source changes, the temperature field exhibits completely different



**Fig. 10.** Comparison of model prediction and numerical calculation results of natural convection temperature field of ellipse-heat source. (a) The major axis is 0.3 R and the minor axis is 0.39 R; (b) The major axis is 0.45 R and the minor axis is 0.58 R; (c) The major axis is 0.57 R and the minor axis is 0.31 R.

distribution patterns; influenced by the Dirichlet boundary conditions at the outer ring, the temperature field shows significant gradient changes near the wall surface.

For clarity, we sampled the temperature values in the computational domain at two locations: the vertical cutline at  $x = 0$  and the vertical cutline at  $y = 0$ . We sampled the temperature values at these locations and compared them with the results obtained from numerical simulations. The corresponding results can be seen in the last column of Fig. 8. In the vertical cutline along the  $x$ -axis, approaching the heat source from below, the temperature slowly increases at first and then rapidly rises. Moving away from the heat source from above, the temperature gradually decreases and then sharply drops near the cold wall surface; In the vertical cutline along the  $y$ -axis, the temperature exhibits a symmetric distribution. The temperature distribution along the cutline closely matches the numerical results, further confirming the accuracy of our model predictions.

To further assess the accuracy of the results, we conducted a quantitative analysis of the error. The local relative error for each case is defined as

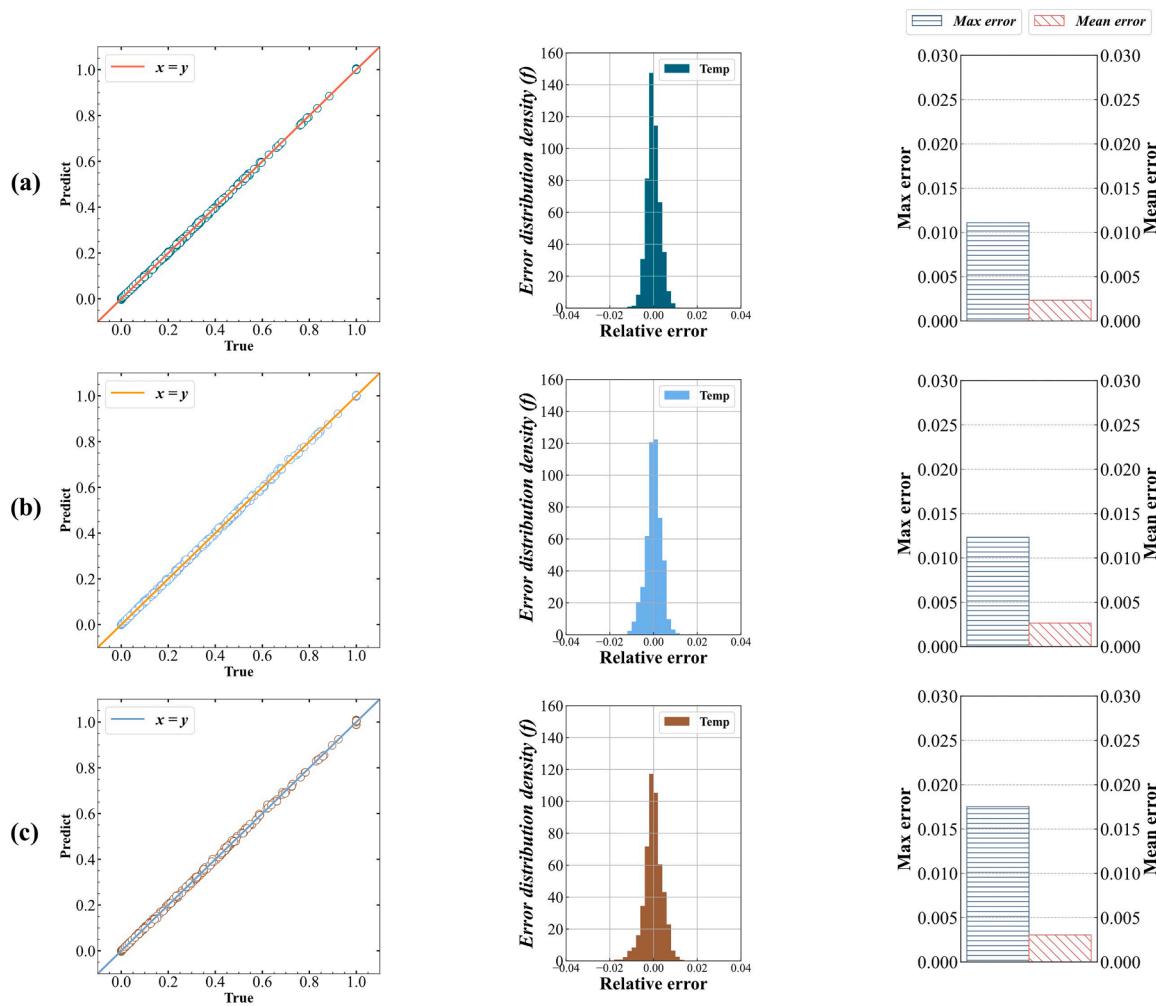
$$E_i = \frac{\varphi_i - \hat{\varphi}_i}{\varphi_h}, i \in N \quad (13)$$

where  $\varphi$  is the physical field simulated by OpenFOAM, and  $\hat{\varphi}$  is the predicted physical field.  $\varphi_h$  represents the temperature at the surface of the heat source. As shown in Fig. 9, we performed a statistical analysis on the predicted temperature distribution at each node, as well as the relative error of the temperature field and the maximum and mean errors in temperature prediction.

We randomly selected 200 prediction points from the flow field as a scatter plot set for examination. The leftmost column in the figure represents their distribution relative to the numerical calculation results.

The solid line represents the numerical results, while the empty circles represent the prediction results. Different cases are distinguished by different colors. The prediction results of the three cases are distributed near the numerical results without significant fluctuations. This indicates that the model accurately predicts the temperature at different locations in the computational domain. The middle column of the figure shows the frequency distribution of relative errors in temperature prediction. The model predicts the temperature field for all three cases with a majority of errors concentrated around zero. However, as the radius of the heat source decreases, the model's performance in temperature prediction relatively deteriorates, as indicated by a decrease in the frequency near zero. This observation is consistent with the phenomenon of larger prediction errors at the top of the annulus shown in Fig. 8. We attribute this to the fact that as the heat source decreases in size, the temperature variations become more concentrated, leading to rapid fluctuations in temperature at a smaller scale at the top of the annulus, increasing the difficulty of the model's prediction. This can also be observed from the maximum and average errors in the far-right column of the figure. As the radius of the heat source decreases, the maximum error in the model's temperature predictions increases from 1.5% to 2.5%. However, the average errors for all three cases are relatively small (about 0.4%), indicating that the model's predictions of the temperature field in the computational domain have overall low errors. Despite the fluctuation in errors observed in the predictions for different heat source sizes, the results remain within a reasonable range. These findings demonstrate that the proposed model exhibits strong robustness to changes in the radius of the heat source in the task of predicting steady-state natural convection with a single heat source.

On the other hand, we also investigated the model's adaptability to changes in the shape of the heat source. In the test cases, the heat source had an elliptical shape with varying major and minor axes. We selected three independent test cases that were not included in the training set.



**Fig. 11.** Error analysis of model prediction results of natural convection temperature field of ellipse-heat source. (a) The major axis is 0.3 R and the minor axis is 0.39 R; (b) The major axis is 0.45 R and the minor axis is 0.58 R; (c) The major axis is 0.57 R and the minor axis is 0.31 R.

The major and minor axes of the elliptical heat sources were  $[[0.3, 0.39], [0.45, 0.58], [0.57, 0.31]]$  R, while keeping other physical conditions consistent. The results are shown in Fig. 10.

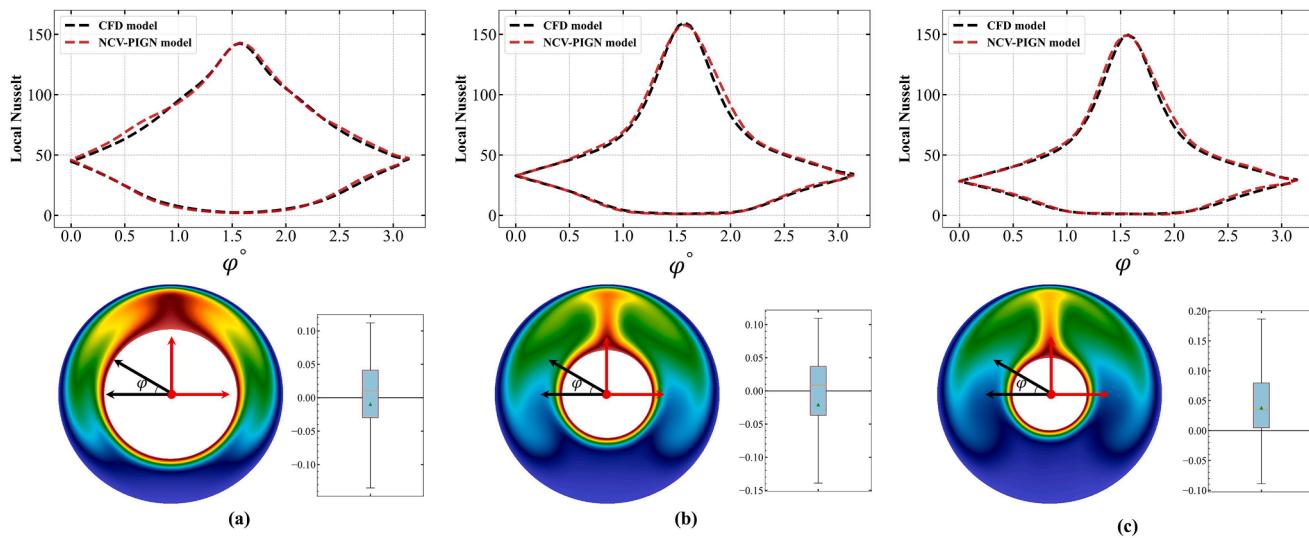
The comparison with the numerical calculations demonstrates that the model accurately predicts the temperature fields of the three test cases. This indicates that the model has learned the underlying patterns of the impact of changes in the major and minor axes on natural convection during training, thus adapting to the variations in the shape of the heat source. Similarly, we sampled the temperature field results at the cutline  $x = 0$  and  $y = 0$  and compared them with the numerical simulation results. The observed patterns of temperature distribution along the cutline are consistent with those of circular heat sources and closely match the numerical calculations. This confirms the powerful fitting capability of the neural network in solving natural convection problems. Subsequently, we analyzed the predicted errors using Eq. (13), and the results are shown in Fig. 11. From left to right in the figure, we have the distribution of predicted temperature fields relative to the numerical calculations, the relative errors of the temperature fields, and the statistical results of the maximum and mean errors of the predicted temperature.

The results indicate that the model performs well in predicting temperature distributions even with changes in the shape of the heat source. The max errors for the three test cases are concentrated below 2%, and the larger errors are mainly located at the top of the heat source. When the major axis is longer than the minor axis, the model tends to have slightly larger maximum errors. However, these larger errors only

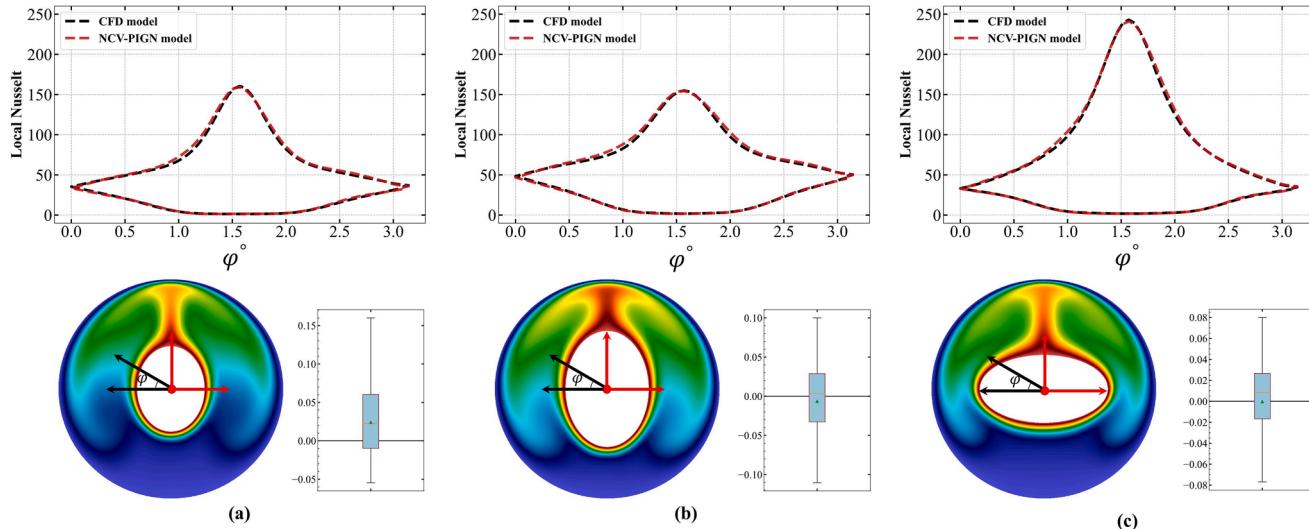
account for a small portion of the overall predictions, and the average error, which is less than 0.3%, supports this conclusion. In summary, through the analysis of heat source examples with different sizes and shapes, we have demonstrated the effectiveness of the coupled model that incorporates physical information embedding and graph convolutional neural network. With only 20 training data, the model is capable of accurately predicting the influence of different heat source shapes on the circulation patterns of natural convection. A fully trained model can precisely establish the underlying connection between geometric variations and steady-state physical fields, thereby constructing a comprehensive understanding of the evolution from geometric shapes to the corresponding steady-state physical phenomena.

At the end of this section, we have validated the idea that the introduction of GCN can directly learn features of non-Euclidean data. Directly learning features from non-Euclidean raw data can ensure that data accuracy is not lost in the preprocessing process. Especially for the annular structure, the accuracy of the prediction results on the curved boundary (wall surface) is an important basis for verifying this idea. Therefore, we sampled the temperature results on the outer ring surface and analyzed the model's prediction accuracy on the physical boundary by calculating the Nusselt number on the outer ring. The Nusselt number is a dimensionless physical quantity describing the rate of energy transfer from the surface. We define the local Nusselt number as,

$$Nu = -\frac{\partial T}{\partial n} \quad (14)$$



**Fig. 12.** Comparison of predicted and numerical simulated results of Nu numbers for natural convection of circle-heat source. (a) Heat source radius is 0.59 R; (b) Heat source radius is 0.4 R; (c) Heat source radius is 0.33 R.



**Fig. 13.** Comparison of predicted and numerical simulated results of Nu numbers for natural convection of ellipse-heat source. (a) The major axis is 0.3 R and the minor axis is 0.39 R; (b) The major axis is 0.45 R and the minor axis is 0.58 R; (c) The major axis is 0.57 R and the minor axis is 0.31 R.

where  $n$  is unit outward normal to the boundary surface,  $Nu$  is calculated Nusselt number. The percentage error of the Nusselt numbers that is averaged over all points on the contour line:

$$\mathcal{E}_{Nu} = \frac{1}{K} \sum_{i=1}^K \frac{|Nu_i - \widehat{Nu}_i|}{Nu_i} \quad (15)$$

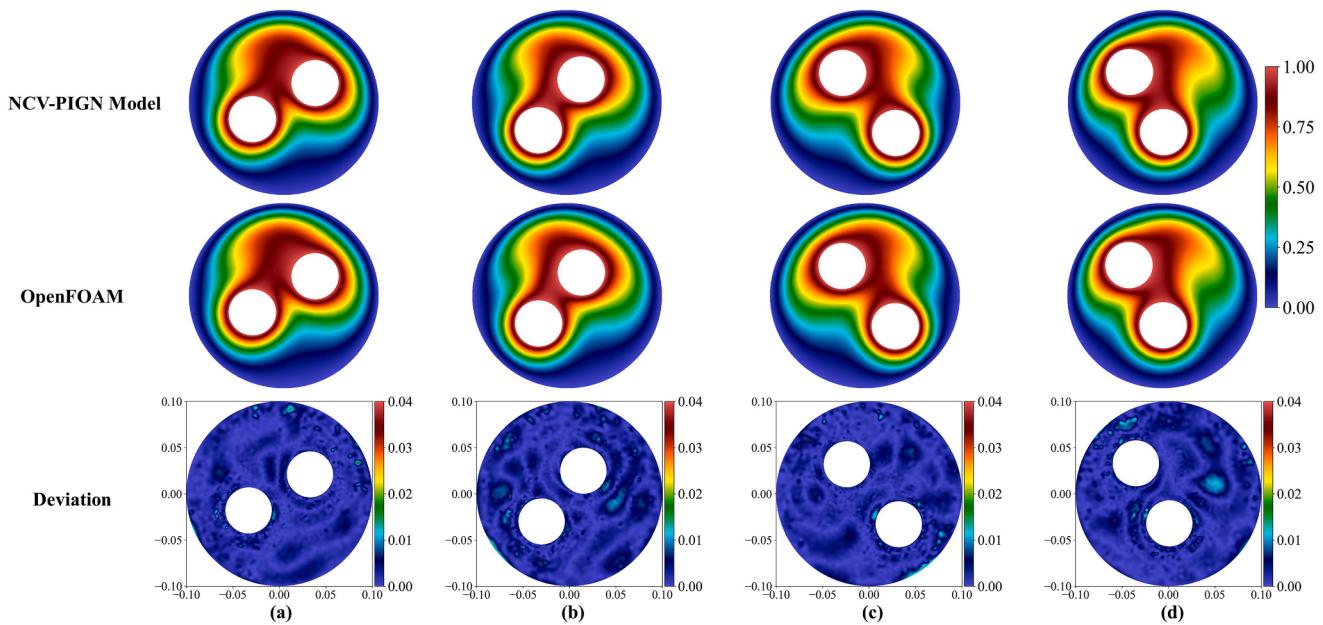
where  $K$  is the number of points on the contour line, and  $\widehat{Nu}$  is a Nusselt number inferred by the proposed model. Figures 12 and 13 shows the prediction and comparison results of circle-heat source and ellipse-heat source cases, respectively.

The solid curve represents the simulated Nu, which was calculated using the temperature results from numerical simulation based on Eq. (14). The dashed red line represents the predicted Nu, which was also calculated using the predicted temperature results based on Eq. (14). By comparing the three sets of curves, we can observe that the predicted and simulated Nusselt numbers are in close agreement. We conducted a statistical analysis of the errors using the definition provided in Eq. (15).

For the three cases with circular heat sources, the maximum predicted errors compared to the numerical simulation results were 3.27%, 5.39%, and 5.17% respectively. For the three cases with elliptical heat sources, the maximum predicted errors compared to the numerical simulation results were 3.5%, 3.61%, and 3.08% respectively. The peak values of Nu for all cases were observed at the top of the heat source, and the magnitude of the peak was influenced by the shape of the heat source. According to the local Nu definition given in Eq. (14), its value is directly proportional to the gradient of the temperature distribution. These results confirm the high accuracy of the proposed prediction model at the boundaries, which is attributed to the utilization of graph convolutional networks for direct feature learning from the raw data, preserving the precision of the data.

### 3.2. Solving double-heat source field with NCV-PIGN

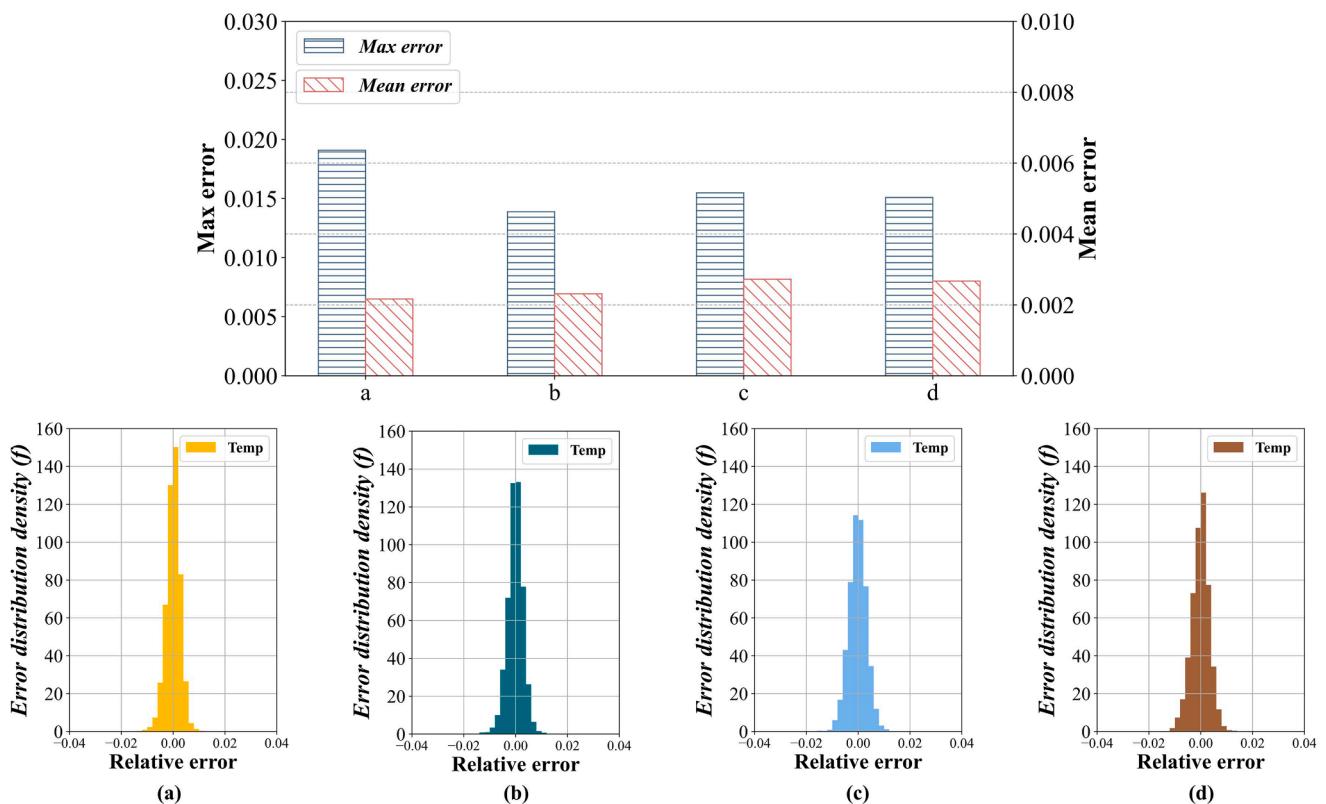
In this subsection, we investigate a scenario where a circular enclosure contains two identical circular heat sources with the same shape and temperature. The model for the double heat sources was trained



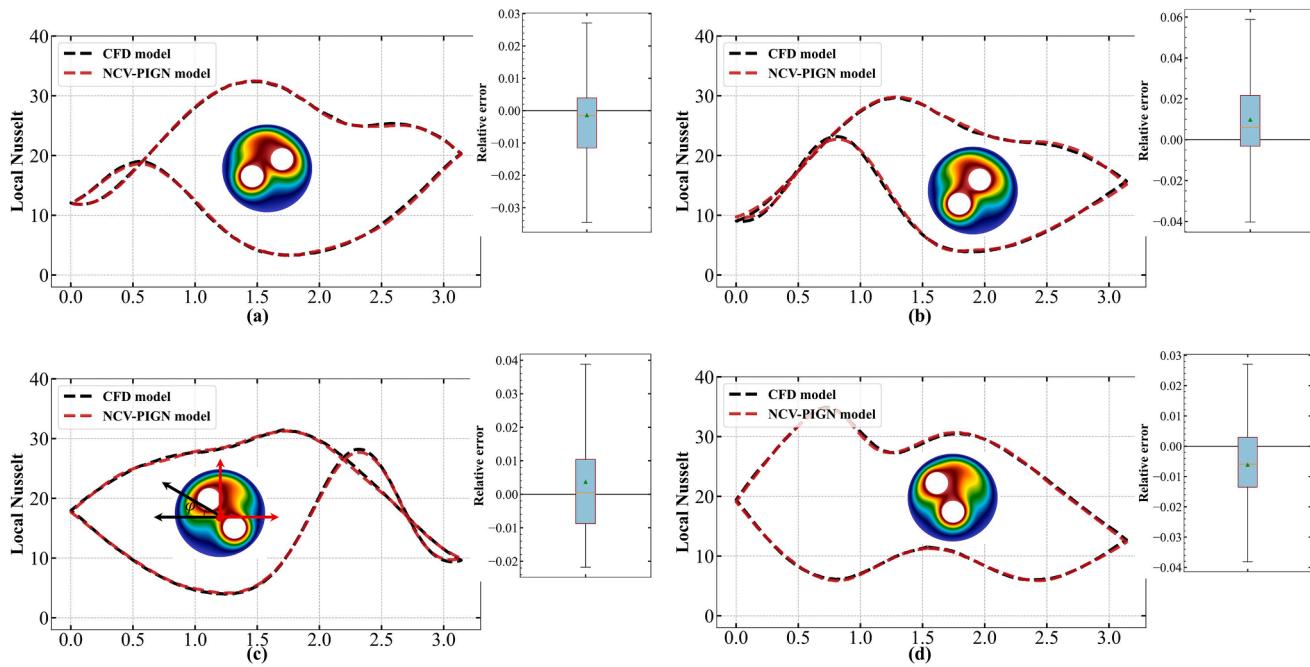
**Fig. 14.** Comparison of model prediction and numerical calculation results of natural convection temperature field of double-heat source. The two heat sources for each case have different distributions.

using a separate set of networks. In order to reach a steady state for the natural convection of the double heat sources, the operating conditions of the physical field were modified to  $\text{Ra} = 7000$  and  $\text{Pr} = 7.07$ . The training duration for the model in the dual heat source cases is approximately 2 h and 7 min. This longer training time can be attributed to the increased number of grid points in each case, which leads to a higher computational cost compared to the single heat source cases. It is worth mentioning that the network architecture used in this case is the

same as the one used for the single heat source. Considering the mutual influence between the heat sources, the complexity of this case is higher than that of the single heat source, which contributes to further verification of the model's performance. The test cases are shown in Fig. 14, where each case contains circular heat sources located at different positions. Due to the varying relative positions of the heat sources, significant differences can be observed in the distribution of the temperature field.



**Fig. 15.** Error analysis of model prediction results of natural convection temperature field of double-heat source.



**Fig. 16.** Comparison of predicted and numerical simulated results of Nu numbers for natural convection of double-heat source.

Similarly, we conducted a comprehensive evaluation of the proposed model's performance in predicting the temperature fields, comparing it to the numerical solver. The comparison between the predicted and CFD values for the temperature fields is presented in Fig. 14. To assess the accuracy, we constructed an error distribution by calculating the absolute differences between the predicted and simulated values and is normalized by  $\varphi_h$ . The four test cases in the computational domain exhibit different distributions of the double heat sources, and their relative positions are also not consistent. From the results, it is evident that there is a mutual interaction between the double heat sources. The temperature contours indicate that when the two heat sources are closer to each other, the intersection of the contours occurs earlier. However, when the relative distance between them is greater (as in case c), the intersection of the temperature contours appears at the next gradient. Based on this phenomenon, we have demonstrated that the model possesses the capability to solve the steady-state natural convection problem with double heat sources. Furthermore, due to the lower Prandtl number in the double heat source cases compared to the single heat source cases, heat diffusion is slower, resulting in smaller temperature gradients. This improvement in the model's predictive performance is evident. The maximum temperature error does not exceed 2% according to the error results, and there is no general pattern observed in its distribution. Moreover, we quantitatively analyze the errors using the same approach, specifically by performing statistical analysis on the predicted results using Eq. (13). Figure 15 sequentially displays the maximum and average errors, as well as the frequency distribution density of the relative errors for these four test cases.

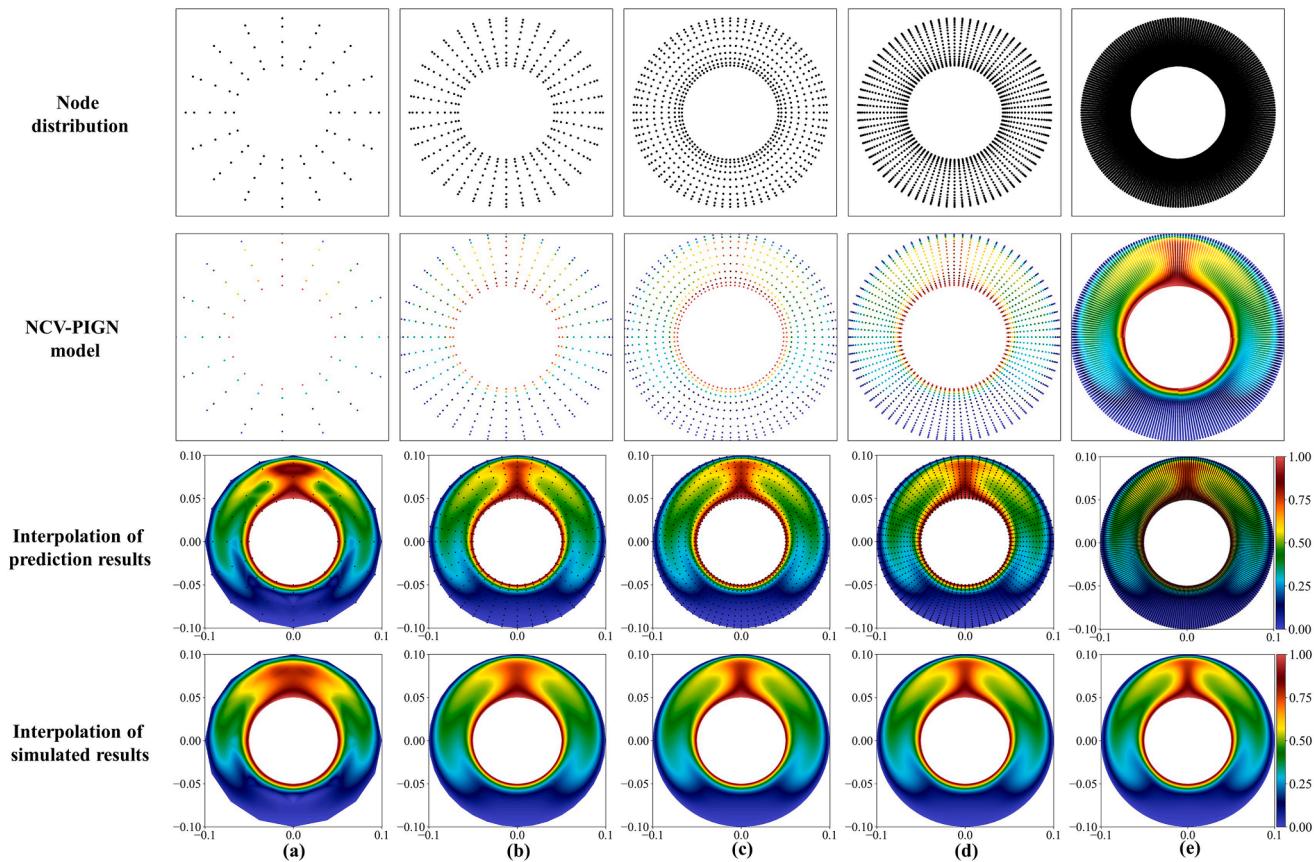
The maximum and average errors for all four cases are small, with values below 2% and 0.3%, respectively. Additionally, the maximum relative error in the predictions is also less than  $\pm 2\%$ . A significant number of predicted results are distributed near zero. Among the four test cases, case (c) exhibits relatively fewer instances of predicted results near zero. Based on the analysis of the positions of the two cylinders in case (c), we believe that the temperature between the two cylinders undergoes rapid changes from high to low and then back to high, resulting in a typical temperature field distribution that leads to an increase in prediction error. Nevertheless, considering that the model was trained using only 20 data, it demonstrates excellent performance in the adaptive prediction task involving multiple heat sources.

In finally, we analyze the Nusselt numbers in the natural convection field with double heat sources. We have still chosen the local Nusselt number distribution along the normal direction of the surface of the cold cylinder, as the results shown in Fig. 16.

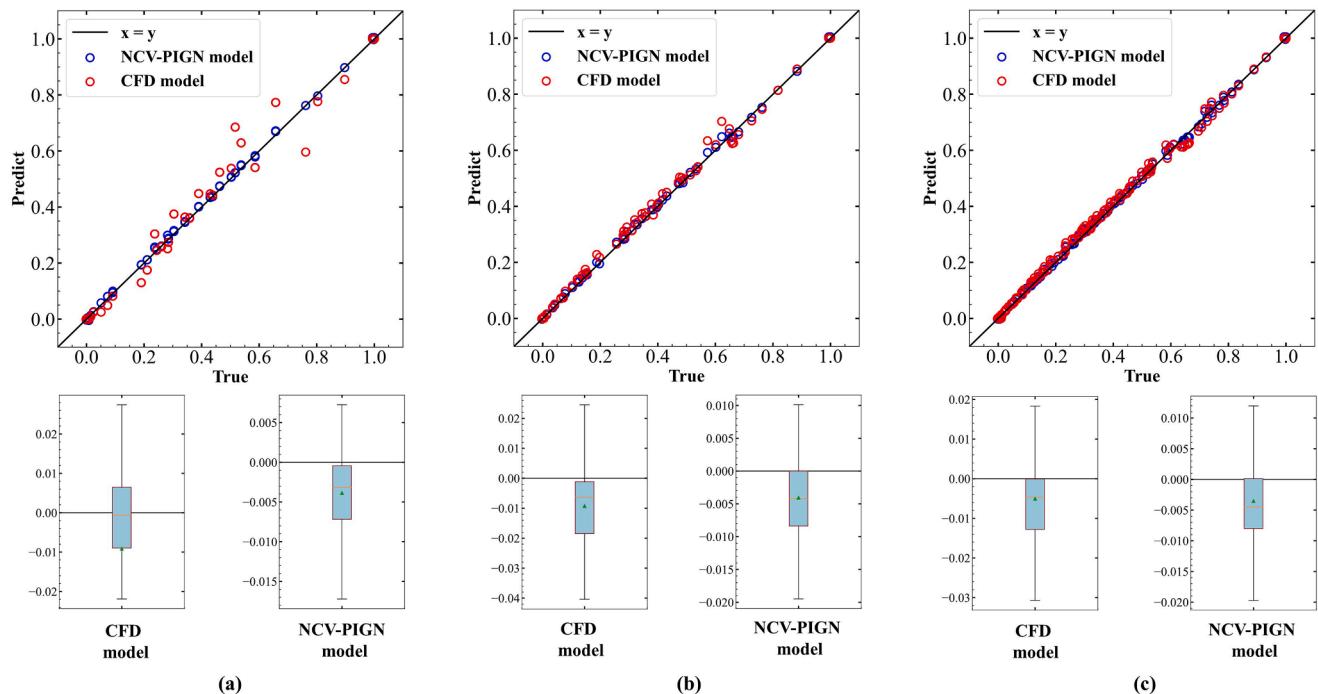
Using Eq. (14), the Nusselt value was calculated from the results of predicted temperature field. We can see from Fig. 16, compared to the single heat source cases, the Nu in the presence of double heat sources exhibit more complex shapes on the cold wall. These curves can intersect themselves, and the peak values of Nu as well as their locations vary significantly. In general, when the heat sources are positioned closer to the cold wall surface, it leads to larger temperature gradients, resulting in higher Nu values. Additionally, we observed only minimal deviations between the predicted and simulated Nu values. Using the same method as for the single heat source cases, the average prediction errors for Nu values in the four test cases are 1.52%, 2.34%, 1.75%, and 1.55% respectively. From the boxplot of the relative errors in Nu values, we can observe that the errors are mainly concentrated around zero, demonstrating the stability of the model's predictions. Compared to the prediction of single heat source cases, there is a significant improvement in the accuracy of the model's predictions for the double heat source cases. This improvement can be attributed to the lower heat diffusion rate in the double heat source cases, resulting in smaller temperature gradients within the circular enclosure and reducing the difficulty of prediction for the model. These results indicate that the proposed model exhibits good adaptability to the quantity and positioning of heat sources in the prediction of steady-state natural convection, achieving high accuracy in predicting temperature fields and Nu values.

### 3.3. The impact of mesh node quantity on prediction results

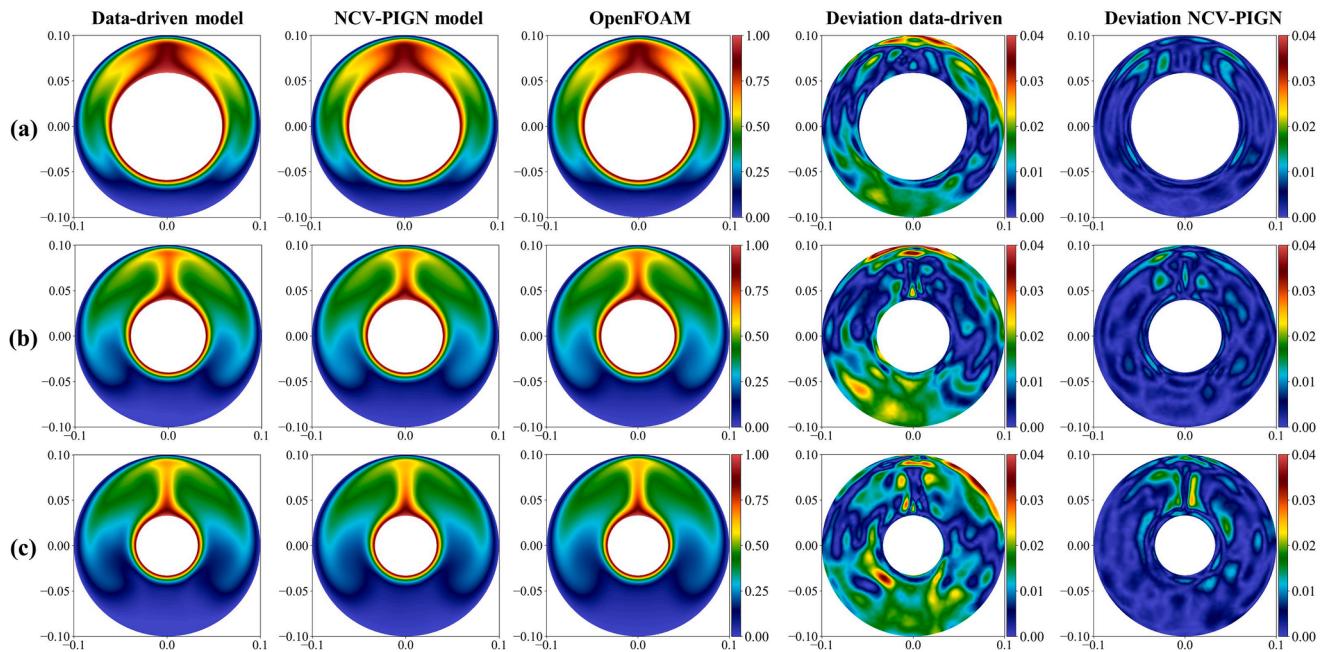
Through the aforementioned results, we conclude that the trained model possesses the capability to solve physical problems. Specifically, in the context of the circular enclosure steady-state natural convection problem addressed in this paper, the proposed model can be regarded as a novel solver based on neural networks. Subsequently, a research is performed in this section to investigate the capabilities of this neural network-based solver. Various configurations of scattered mesh nodes within the domain were explored to solve the governing equation. We prepared four sets of test data with varying numbers of mesh nodes: 80,



**Fig. 17.** Comparison of model prediction and numerical calculation results of natural convection temperature field of circle-heat source with different number of mesh nodes. The number of mesh nodes from (a) to (e) is 80, 360, 760, 1520, and 14,160.



**Fig. 18.** The error distribution of model prediction and numerical calculation relative to the convergence results with different numbers of mesh nodes as input. (a) the number of nodes is 80; (b) the number of nodes is 360; (c) the number of nodes is 760.



**Fig. 19.** Comparison of prediction performance between pure data-driven model and NCV-PIGN model in the natural convection temperature field of circle-heat sources. (a) Heat source radius is  $0.59 R$ ; (b) Heat source radius is  $0.4 R$ ; (c) Heat source radius is  $0.33 R$ .

360, 760, 1520, and 14,160, respectively. The corresponding numerical simulation results were obtained by sampling from the steady-state natural convection field of a single heat source. In the first row of Fig. 17, the distribution of nodes is displayed, with an increasing number of nodes from left to right. In each case, more sampling points were placed at the boundaries to improve the solver's accuracy at the boundaries. The second row of Fig. 17 shows the solver's results for cases with different numbers of mesh nodes. Since scatter plots cannot visually depict the reconstructed temperature distribution, we reconstructed the temperature field using interpolation in the third row. The last row of Fig. 17 presents the numerical simulation results corresponding to the respective numbers of mesh nodes, with the temperature field also reconstructed using the same interpolation method.

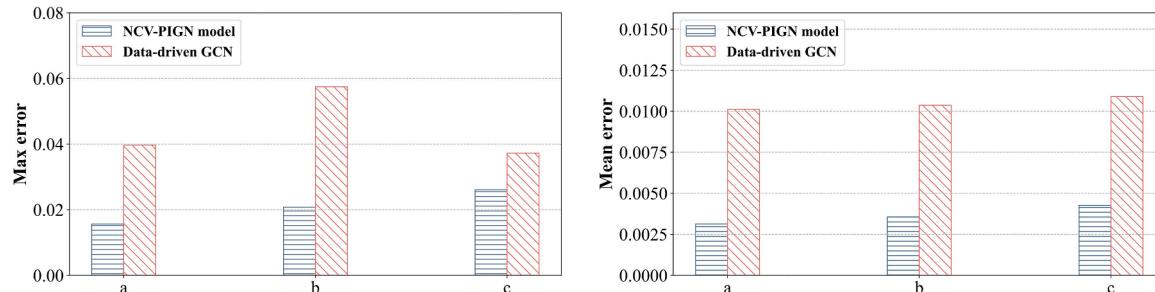
Considering the issue of inaccurate numerical calculations with a low number of mesh nodes, we have used the numerical results obtained with a larger number of nodes and already converged as the benchmark for comparison. Here, we have selected case (e) with 14,160 mesh nodes as the benchmark for comparison. It seems that 360 points are enough to obtain acceptable results while the case of 80 points fails. This result seems to indicate that the inferred temperature field cannot be accurately obtained when there are insufficient measurement points. To further verify this, we subsequently analyzed all prediction results by plotting scatter diagrams. Specifically, we selected cases (a), (b), and (c) for analysis. For each case, we obtained the temperature values at the

corresponding positions from the numerically converged results as the reference for comparison with the predicted results.

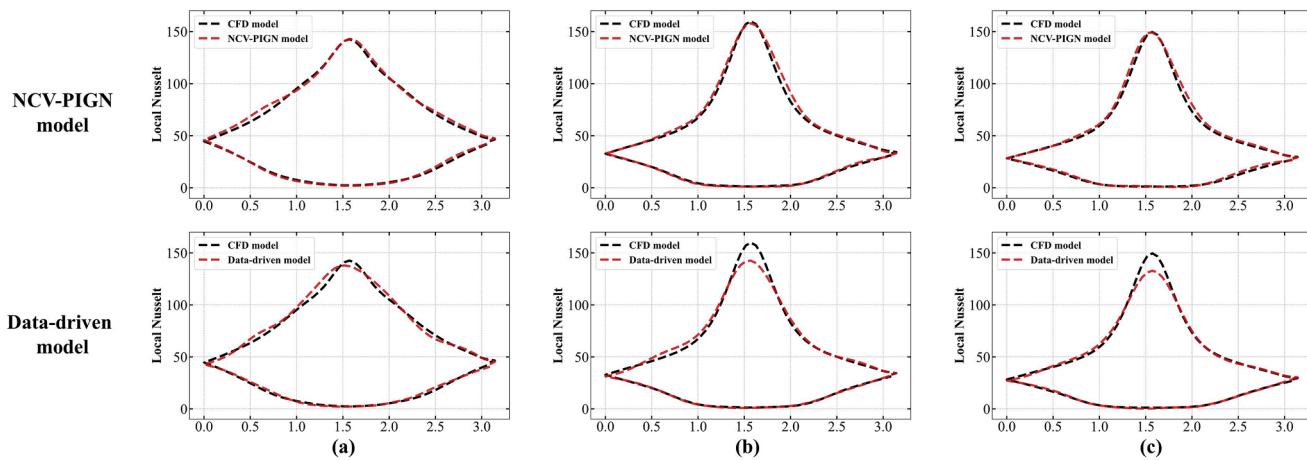
The first row in Fig. 18 displays the predicted results, while the scattered plots represent the deviations of the numerical results from the converged computational results. The second row shows the relative errors of these scattered plots compared to the converged computational results. From the results in the figure, it can be observed that the numerical computation model exhibits larger deviations and higher fluctuations when the number of grid nodes is small. As the number of nodes increases to 760, the computational deviations decrease, and overall fluctuations decrease as well (the convergence of traditional numerical calculations is influenced by the number of grid nodes). However, interestingly, the predicted results of the model do not vary with the change in the number of grid nodes. The scattered plots of the three cases consistently fall near the converged computational results. The box plots of the relative errors also maintain similar distributions. These results indicate that the well-trained model can adapt to changes in the grid at the node level, and its predictive accuracy is no longer affected by the number of nodes.

#### 3.4. Comparing with purely data-driven models

The purpose of this section is to investigate the improvement in model performance with the introduction of control equations. To do



**Fig. 20.** Error analysis of the prediction results of the pure data-driven model and the NCV-PIGN model. On the left is the maximum error and on the right is the average error.



**Fig. 21.** Comparison of Nu numbers results predicted by the pure data-driven model and the NCV-PIGN model.

this, we compared and analyzed the prediction results of physics-based and purely data-driven models. Both models were kept consistent in terms of structure, training data, and selection of training hyperparameters to ensure the scientific validity of the comparison analysis. The training time for the pure data-driven model was 22 min, which is 1 h and 8 min faster than the physics-informed model. This is because the pure data-driven model does not require differentiation of the results during the training process, which saves a significant amount of training time. We selected steady-state natural convection under single heat source conditions as the training data and tested the trained models on three cases with varying radius of a single heat source. In Fig. 19, each row shows the results of the purely data-driven model, physics-embedded model, and numerical calculation, as well as the deviation of the prediction results of the purely data-driven and physics-embedded models from the numerical calculation.

Based on the distribution of temperature deviations in the deviation plots, it can be observed that the majority of deviations in the prediction results of the purely data-driven model occur at the bottom of the annulus, and larger deviations are mainly concentrated at the boundary positions. This preliminary result indicates that the purely data-driven model has lower accuracy in predicting the temperature field distribution compared to the proposed model. Subsequently, we conducted a statistical analysis on the prediction errors of both models, comparing the maximum and average values of the prediction errors.

The bar chart in Fig. 20 shows that the predicted results of the proposed model in different heat source radius test cases, whether it is the maximum error or the average relative error, are lower than those of the purely data-driven model. Furthermore, the average error reflects the stability of the model's prediction results. The purely data-driven model, which only uses the deviation between the network output and the simulated values as the convergence criterion during training, exhibits more fluctuations in predicting test cases outside the training set. This indicates that the purely data-driven model has lower generality compared to the prediction model constrained by the governing equations. Finally, we compared the prediction performance of the two models at the boundaries by analyzing the distribution of the Nu on the outer ring. Figure 21 shows the Nusselt number distributions of the purely data-driven and physically-driven models in three test cases.

From the results, it can be observed that the purely data-trained model has low accuracy in predicting the Nusselt number at the top of the circular ring, exhibiting significant deviations compared to the numerical calculation results. The maximum prediction errors for the Nusselt numbers in the three test cases are 9.8%, 16.5%, and 17.18%, respectively. These deviations are much larger than those of the physically-driven model. We speculate that this is due to the lack of physical constraints and sufficient training data during the training of

the purely data-driven model, resulting in inferior prediction performance compared to the proposed physically-driven model. These results are sufficient to demonstrate that the introduction of physical information has a positive impact on the model training, effectively enhancing the robustness of the model's predictions. In the steady-state two-dimensional natural convection task presented in this study, the proposed model can improve the prediction performance of the temperature field distribution even with a small amount of training data.

#### 4. Conclusion

This paper presents a physics-informed graph convolutional neural networks (NCV-PIGN) for modeling geometry-adaptive steady-state natural convection problems. The graph neural network in the model is primarily used to identify the geometric information in the flow field. In contrast to the classic convolutional network, which employs pixilation techniques for data preprocessing, the graph-based model directly learns the inherent features of the fluid dynamics system from the unstructured grid. This approach allows the model to take into account the connections between nodes and their spatial properties. On the other hand, the physics-informed neural network imposes constraints from the governing equations during the model training process. Compared to purely data-driven models, it exhibits better performance with less training data and enhances the physical interpretability of the model.

The proposed model focuses on the adaptive capability of graph neural networks with physical information embedding for temperature field prediction under the change of geometric shape, number, and position. To evaluate the performance of the model, we conducted tests using circular objects with built-in heat sources and compared the performance of the models trained using purely data-driven and physics-informed methods. The results demonstrated the feasibility and scalability of the proposed model for predicting the distribution of temperature over a non-uniform field while adapting to changes of problem geometry. Since the NCV-PIGN model is constrained by governing equations, it can learn to solve partial differential equations even with limited sampling data. Furthermore, incorporating governing equations also enhances the focus of the neural network on the boundary conditions, which results in a high prediction accuracy of the Nu number at the boundary.

It is important to note that the scope of this work is limited to two-dimensional steady-state natural convection. While the proposed model demonstrates excellent performance within this context, its applicability to other flow regimes or higher-dimensional problems may require further investigation. Expanding the problem to three dimensions and incorporating transient behavior would pose greater challenges for the model, as it would involve dealing with data in two

**Table 1**

Parameters of the neural network models.

Hyper-parameter (GCN)	Structure of GCN
Inputs. (number of cases, nodes per case, features per node)	X, (N, n, m)
Number of layers and neurons per layer. (l, n <sub>l</sub> )	(6, 128)
Activation function	Tanh ()
Batch size	2
Learning rate	1e-4
Number of epochs	(Adam: 5000; L-BFGS: 15,000)

more dimensions. The current model is not directly capable of handling such scenarios, as the exponential increase in data would impose additional training burdens. To address this, we would need to redesign the network architecture to accommodate the three-dimensional structure of the data or modify the structure of the training data to align the learnable features with the problem we are investigating. This will be an area of further research in future.

### 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.

### Acknowledgment

This work is supported by Natural Science Foundation of Jiangsu Province BK20201302.

### Appendix A

The hyper-parameters of the NCV-PIGN model are shown in Table 1. In the table, N is training cases, n denotes the number of nodes per case, m represents the features per node. During the training process, we performed 5000 steps of preliminary training using the Adam optimizer. Subsequently, we utilized the L-BFGS optimizer to conduct an additional 15,000 steps of training.

### Appendix B

In this work, the hyper-parameters are determined by trial-and-error. The following is the process of tuning key hyper-parameters.

**Learning rate:** The learning rate determines the step size of parameter updates during training iteration. Based on our experience in parameter tuning, a learning rate that is too large can cause the model's loss to converge poorly, while a learning rate that is too small may lead to the network getting trapped in a local optimum and resulting in incorrect convergence. During the parameter tuning process, we experimented with a learning rate range of [5e-5, 5e-3]. Ultimately, a learning rate of 1e-4 for the model was selected based on its performance.

**The structure of GCN:** The number of layers and neurons in a graph convolutional network (GCN) significantly impact the model's performance. In principle, having more layers and neurons can help the model learn more complex features and relationships, thereby enhancing its expressive power. However, there are three potential issues to consider. Firstly, it can lead to overfitting. Secondly, training deep models may encounter problems such as vanishing or exploding gradients, making convergence difficult. Lastly, having a large number of model

parameters requires significant GPU resources. In the process of tuning these two hyperparameters, it is important to consider the number of input nodes and the complexity of the problem.

**Activation function:** The choice of activation function has a significant impact on the performance of a neural network. Activation functions introduce nonlinearity, allowing the network to learn and represent complex nonlinear relationships. In order to effectively express partial differential equations, the selected activation function should have non-zero higher-order derivatives. During the parameter tuning and testing process, we experimented with different types of activation functions, including ReLU, Tanh, and Sigmoid.

**Batch size:** The training batch size refers to the number of samples taken from the training set during each parameter update. A larger batch size can speed up training but also requires more computational resources. Additionally, a larger batch size can provide more accurate gradient estimates and make parameter updates more stable. On the other hand, smaller batch sizes can improve the model's generalization ability by introducing more randomness and diversity during the training process, making it easier for the model to avoid local optima. However, this can also lead to oscillations or instability during training. In this study, we used a batch size of 2 during the Adam training phase and a full batch (i.e., using all available samples) during the L-BFGS phase. The primary reason for this allocation is to ensure that the model has sufficient generalization performance while satisfying the governing equation constraints for each case.

**Number of epochs:** Determining the number of epochs requires considering the model's convergence and the risk of overfitting. We adjust the training epochs based on the performance of the model on the validation set. The value of the Adam training epochs is between [1000, 10,000], while L-BFGS is between [5000, 30,000]. Having too few epochs may result in the model not achieving accurate prediction performance. Conversely, having too many epochs can lead to overfitting on the training dataset, causing a loss in generalization performance.

### References

- [1] H. Han, D.B. Ingham, Y. Yuan, The boundary element method for the solution of the backward heat conduction equation, *J. Comput. Phys.* 116 (2) (1995) 292–299.
- [2] J.C. Bruch Jr, G. Zvoloski, Transient two-dimensional heat conduction problems solved by the finite element method, *Int. J. Numer. Methods Eng.* 8 (3) (1974) 481–494.
- [3] W. Li, B. Yu, X. Wang, P. Wang, S. Sun, A finite volume method for cylindrical heat conduction problems based on local analytical solution, *Int. J. Heat Mass Transf.* 55 (21–22) (2012) 5570–5582.
- [4] M. Fahs, A. Younes, A high-accurate Fourier-Galerkin solution for buoyancy-driven flow in a square cavity, *Numer. Heat Transf. Part B Fundam.* 65 (6) (2014) 495–517.
- [5] H. Sajjadi, A.A. Delouei, M. Izadi, R. Mohebbi, Investigation of MHD natural convection in a porous media by double MRT lattice Boltzmann method utilizing MWCNT–Fe3O4/water hybrid nanofluid, *Int. J. Heat Mass Transf.* 132 (2019) 1087–1104.
- [6] P. Zhang, X. Zhang, J. Deng, L. Song, A numerical study of natural convection in an inclined square enclosure with an elliptic cylinder using variational multiscale element free Galerkin method, *Int. J. Heat Mass Transf.* 99 (2016) 721–737.
- [7] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* 52 (2020) 477–508.
- [8] B. Kwon, F. Ejaz, L.K. Hwang, Machine learning for heat transfer correlations, *Int. Commun. Heat Mass Transf.* 116 (2020), 104694.
- [9] M.M. Rajabi, et al., Analyzing the efficiency and robustness of deep convolutional neural networks for modeling natural convection in heterogeneous porous media, *Int. J. Heat Mass Transf.* 183 (2022), 122131, <https://doi.org/10.1016/j.ijheatmasstransfer.2021.122131>.
- [10] X. Xu, F. Waschkowski, A.S.H. Ooi, R.D. Sandberg, Towards robust and accurate Reynolds-averaged closures for natural convection via multi-objective CFD-driven machine learning, *Int. J. Heat Mass Transf.* 187 (2022), 122557.
- [11] X.-J. He, C.-H. Yu, Q. Zhao, J.-Z. Peng, Z.-H. Chen, Y. Hua, Reduced order modelling of natural convection of nanofluids in horizontal annular pipes based on deep learning, *Int. Commun. Heat Mass Transf.* 138 (2022), 106361.
- [12] M.M. Rajabi, et al., Analyzing the efficiency and robustness of deep convolutional neural networks for modeling natural convection in heterogeneous porous media, *Int. J. Heat Mass Transf.* 183 (2022), 122131.
- [13] Y. Tizkast, M. Kaddiri, M. Lamsaadi, T. Makayssi, Machine Learning based algorithms for modeling natural convection fluid flow and heat and mass transfer in rectangular cavities filled with non-Newtonian fluids, *Eng. Appl. Artif. Intell.* 119 (2023), 105750, <https://doi.org/10.1016/j.engappai.2022.105750>.

- [14] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* 3 (3) (2021) 218–229, <https://doi.org/10.1038/s42256-021-00302-5>.
- [15] S. Durgam, A. Bhosale, V. Bhosale, R. Deshpande, P. Sutar, S. Kamble, Temperature prediction of heat sources using machine learning techniques, *Heat Transf.* 50 (8) (2021) 7817–7838, <https://doi.org/10.1002/htj.22255>.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed DeepONets, *Sci. Adv.* 7 (40) (2021) eabi8605.
- [20] S. Cai, Z. Wang, C. Chryssostomidis, G.E. Karniadakis, Heat transfer prediction with unknown thermal boundary conditions using physics-informed neural networks, in: *Fluids Engineering Division Summer Meeting, American Society of Mechanical Engineers, 2020. V003T05A054*.
- [21] R. Laubscher, Simulation of multi-species flow and heat transfer using physics-informed neural networks, *Phys. Fluids* 33 (8) (2021) 1–14, <https://doi.org/10.1063/5.0058529>.
- [22] H. Bararnia, M. Esmaeilpour, On the application of physics informed neural networks (PINN) to solve boundary layer thermal-fluid problems, *Int. Commun. Heat Mass Transf.* 132 (2022), 105890, <https://doi.org/10.1016/j.icheatmasstransfer.2022.105890>.
- [23] H. Gao, L. Sun, J.-X. Wang, PhyGeoNet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain, *J. Comput. Phys.* 428 (2021), 110079.
- [24] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, *J. Machine Learn. Res.* 18 (2018) 1–43.
- [25] X. Jin, S. Cai, H. Li, G.E. Karniadakis, NSFnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 426 (2021), 109951.
- [26] A.D. Jagtap, E. Kharazmi, G.E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems, *Comput. Methods Appl. Mech. Eng.* 365 (2020), 113028.
- [27] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.* 3 (6) (2021) 422–440.
- [28] Y. Li, T. Liu, Y. Xie, Thermal fluid fields reconstruction for nanofluids convection based on physics - informed deep learning, *Sci. Rep.* (2022) 1–23, <https://doi.org/10.1038/s41598-022-16463-1>.
- [29] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks for heat transfer problems, *J. Heat Transf.* 143 (6) (2021).
- [30] N. Zobeiry, K.D. Humfeld, A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications, *Eng. Appl. Artif. Intell.* 101 (2021), 104232, <https://doi.org/10.1016/j.engappai.2021.104232>. September 2020.
- [31] P. Kim, P. Kim, Convolutional neural network, in: *MATLAB Deep Learn. with Mach. Learn. Neural Networks Artif. Intell.*, 2017, pp. 121–147.
- [32] J.D. Bodapati, N. Veeranjaneyulu, Feature extraction and classification using deep convolutional neural networks, *J. Cyber Secur. Mobil.* (2019) 261–276.
- [33] J.Z. Peng, X. Liu, N. Aubry, Z. Chen, W.T. Wu, Data-driven modeling of geometry-adaptive steady heat conduction based on convolutional neural networks, *Case Stud. Therm. Eng.* 28 (November) (2021), 101651, <https://doi.org/10.1016/j.csite.2021.101651>.
- [34] Q. Lin, J. Hong, Z. Liu, B. Li, J. Wang, Investigation into the topology optimization for conductive heat transfer based on deep learning approach, *Int. Commun. Heat Mass Transf.* 97 (2018) 103–109.
- [35] S. Tadeparti, V.V.R. Nandigana, Convolutional neural networks for heat conduction, *Case Stud. Therm. Eng.* 38 (2022), 102089.
- [36] W.R. Morningstar, et al., Data-driven reconstruction of gravitationally lensed galaxies using recurrent inference machines, *Astrophys. J.* 883 (1) (2019) 14.
- [37] T.N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv Prepr. arXiv1609.02907*, 2016.
- [38] S. Rezaei, A. Harandi, A. Moeineddin, B.-X. Xu, S. Reese, A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: comparison with finite element method, *Comput. Methods Appl. Mech. Eng.* 401 (2022), 115616.
- [39] D. Das, M. Roy, T. Basak, Studies on natural convection within enclosures of various (non-square) shapes – A review, *Int. J. Heat Mass Transf.* 106 (2017) 356–406, <https://doi.org/10.1016/j.ijheatmasstransfer.2016.08.034>.
- [40] L. Sun, H. Gao, S. Pan, J.X. Wang, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, *Comput. Methods Appl. Mech. Eng.* 361 (2020), 112732, <https://doi.org/10.1016/j.cma.2019.112732>.
- [41] R. Bollapragada, J. Nocedal, D. Mudigere, H.-J. Shi, P.T.P. Tang, A progressive batching L-BFGS method for machine learning, in: *International Conference on Machine Learning, PMLR, 2018*, pp. 620–629.