International Communications in Heat and Mass Transfer 159 (2024) 108036

Contents lists available at ScienceDirect

International Communications in Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ichmt

Physics-informed neural network simulation of conjugate heat transfer in manifold microchannel heat sinks for high-power IGBT cooling

Xiangzhi Zhang [a], Chaofan Tu [b], Yuying Yan [c,*]

[a] Liangzhu Laboratory, Zhejiang University Medical Center, Hangzhou, China
[b] School of Computer Science, University of Nottingham, Nottingham, UK
[c] Faculty of Engineering, University of Nottingham, UK

ARTICLE INFO

Keywords:
IGBT cooling
Manifold microchannel heat sink
Physics-informed neural network
Numerical simulation

ABSTRACT

This study explores the application of Physics-Informed Neural Networks (PINNs) in modeling fluid flow and heat transfer dynamics within intricate geometric configurations, focusing on manifold microchannel (MMC) heat sinks designed for efficient high-power IGBT cooling. A deep neural network architecture comprising two sub-PINNs, one for flow dynamics and another for thermal behavior, is developed, each initialized with a sine activation function to capture high-order derivatives and address the vanishing gradient problem. Comparisons between PINN and CFD simulations reveal close agreement, with both methods showing an increase in pressure drop and a decrease in temperatures as inlet velocity increases. Discrepancies arise in scenarios with rapid flow pattern or gradient changes, highlighting PINNs' sensitivity to geometric complexity and numerical stability. Overall, this study underscores PINNs' potential as a promising tool for advancing thermal management strategies across various engineering applications.

1. Introduction

Insulated Gate Bipolar Transistors (IGBTs) are crucial semiconductor components in power electronics, used across sectors including wind energy, solar power, rail transportation, and power conversion systems. Functioning as switches in converters and inverters, IGBTs amalgamate the advantageous traits of MOSFETs with the robust current-handling capabilities inherent in bipolar transistors to withstand high levels of current and voltage. The necessity for meticulous cooling arrangements for IGBTs stems from their inherent tendency toward significant thermal dissipation during operation, primarily due to switching and conduction losses. Switching losses occur during dynamic transitions between the IGBT's on and off states, leading to rapid fluctuations in voltage and current, while conduction losses manifest when the IGBT operates in its on-state, resulting in power dissipation as heat. Absence of effective cooling mechanisms leads to a notable increase in the temperature profile of the IGBT, adversely affecting efficiency, reliability, and device longevity. Permissible temperatures for conventional IGBTs typically range below 150–175 °C [1], as exceeding these thresholds reduces system efficiency and increases thermal stress, potentially resulting in system deformation and reduced lifespan.

Commonly used cooling technologies [2] for power electronics include natural air cooling, forced air cooling, forced liquid cooling, and phase change thermal management [3]. While natural air cooling offers a cost-effective solution, its efficacy may be limited, especially in high-temperature environments. Forced air cooling provides enhanced cooling performance but entails increased complexity and potential noise issues. Forced liquid cooling offers superior efficiency and uniform temperature distribution, albeit requiring additional equipment and maintenance. Phase change thermal management provides passive cooling benefits, yet its applicability may be constrained by considerations such as temperature range and material compatibility. Therefore, air cooling systems are favored due to their reliability, simplicity, and cost-effectiveness in managing IGBT thermal conditions [4]. Microchannel heat sinks for forced air cooling, as depicted in Fig. 1a, have emerged as effective solutions due to their compact design and minimal coolant usage, making them particularly suitable for scenarios with spatial constraints and demanding thermal dissipation requirements. In contrast to traditional microchannel heat sinks, manifold microchannel heat sinks [5,6], illustrated in Fig. 1b, feature numerous inlet and outlet channels arranged at periodic intervals along the length of the microchannel. These manifold structures serve as flow dividers, markedly reducing the flow length to a fraction of the total length. Moreover, the enhanced cooling capabilities of microchannel heat sinks facilitate

* Corresponding author.
E-mail addresses: chaofan.tu@nottingham.ac.uk (C. Tu), yuying.yan@nottingham.ac.uk (Y. Yan).

https://doi.org/10.1016/j.icheatmasstransfer.2024.108036

Available online 5 September 2024

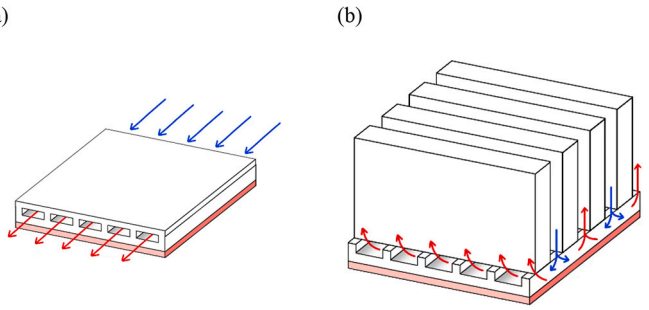


Fig. 1. Schematic views of (a) a traditional microchannel (TMC) heat sink, and (b) a manifold microchannel (MMC) heat sink.

efficient management of discrete components characterized by high heat flux.

Computational fluid dynamics (CFD) simulations are extensively employed in both scientific research and engineering endeavors for thermal management [7–9]. Despite significant advancements in simulating multi-physics problems through numerical discretization of partial differential equations (PDEs), challenges [10] persist in seamlessly integrating noisy data into existing algorithms, navigating complex mesh generation, and addressing high-dimensional problems governed by parameterized PDEs. Furthermore, solving inverse problems [11] involving hidden physics remains prohibitively expensive and necessitates distinct formulations. By combining thermal simulation analysis with machine learning algorithms, Vafai et al. developed a support vector regression (SVR) model which can accurately predict the parameters in the complex internal structure of multi-layer 3D chips [12]. While deep learning shows promise in addressing these issues [13], supervised learning techniques mainly rely on traditional data-driven neural networks, which can be costly and time-consuming to generate data [14]. Moreover, these models may not consistently adhere to fundamental physical laws, leading to extrapolation errors and unreliable outcomes. Recent research has emphasized the promising role of Physics-Informed Neural Networks (PINNs) in various engineering domains [15], which have demonstrated effectiveness in addressing real-world challenges marked by noisy data and incomplete physics. PINNs leverage automatic differentiation [16] to accurately evaluate differential operators without discretization errors, while also employing a multi-task learning approach to fit observed data and ensure compliance with the underlying principles of physics.

Consider the general form of a PDE given by:

$$\begin{cases} \mathscr{N}_i[u](\mathbf{x}) = f_i(\mathbf{x}), \forall i \in \{1, \cdots, N_{\mathscr{N}}\}, \mathbf{x} \in \mathscr{D} \\ \mathscr{C}_j[u](\mathbf{x}) = g_j(\mathbf{x}), \forall j \in \{1, \cdots, N_{\mathscr{C}}\}, \mathbf{x} \in \partial\mathscr{D} \end{cases}, \tag{1}$$

where $\mathscr{N}_i$ is the general differential operator, $\mathbf{x}$ denotes the set of independent variables defined over a bounded continuous domain $\mathscr{D} \subseteq \mathbb{R}^D, D \in \{1, 2, 3, \cdots\}$, and $u(\mathbf{x})$ represents the solution to the PDE; $\mathscr{C}_j$ denotes the constraint operator, encompassing differential, linear, and nonlinear terms, typically covering boundary and initial conditions; $\partial\mathscr{D}$ represents a subset of the domain boundary necessary for defining the constraints. The solution $u(\mathbf{x})$ is approximated by a neural network $u_{NN}(\mathbf{x}; \theta)$. In its simplest form, this neural network comprises fully connected layers and is represented as follows:

$$\begin{cases} u_{NN}(\mathbf{x}; \theta) = \mathbf{W}_n\{\phi_{n-1} \circ \phi_{n-2} \circ \cdots \circ \phi_1 \circ \phi_E\}(\mathbf{x}) + \mathbf{b}_n \\ \phi_i(\mathbf{x}_i) = \sigma(\mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i) \end{cases}, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^{d_0}$ is the input to the network, θ denotes the set of the network's trainable parameters, which are optimized iteratively during the training using variants of the stochastic gradient descent method. Here, n represents the number of layers, $\phi_i \in \mathbb{R}^{d_i}$ is the i^{th} layer of the network, $\mathbf{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$ and $\mathbf{b}_i \in \mathbb{R}^{d_i}$ are the weight and bias of the i^{th} layer, and σ is the activation function. To train the neural network, a loss function is constructed to penalize the derivation of the approximate solution $u_{NN}(\theta)$, with the constraints encoded as penalty terms, defining the following residuals as:

$$\begin{cases} r^{(i)}_{\mathscr{N}}(\mathbf{x}; u_{NN}(\theta)) = \mathscr{N}_i[u_{NN}(\theta)](\mathbf{x}) - f_i(\mathbf{x}) \\ r^{(j)}_{\mathscr{C}}(\mathbf{x}; u_{NN}(\theta)) = \mathscr{C}_j[u_{NN}(\theta)](\mathbf{x}) - g_j(\mathbf{x}) \end{cases}, \tag{3}$$

where $r^{(i)}_{\mathscr{N}}$ and $r^{(j)}_{\mathscr{C}}$ represent the PDE and constraint residuals, respectively. The loss function is then formulated as:

$$\mathscr{L}_{\text{res}}(\theta) = \sum_{i=1}^{N_{\mathscr{N}}} \int_{\mathscr{D}} \lambda^{(i)}_{\mathscr{N}}(\mathbf{x}) \left\| r^{(i)}_{\mathscr{N}}(\mathbf{x}; u_{NN}(\theta)) \right\|_p d\mathbf{x} + \sum_{j=1}^{N_{\mathscr{C}}} \int_{\partial\mathscr{D}} \lambda^{(j)}_{\mathscr{C}}(\mathbf{x}) \left\| r^{(j)}_{\mathscr{C}}(\mathbf{x}; u_{NN}(\theta)) \right\|_p d\mathbf{x}, \tag{4}$$

where $\left\| r^{(i)}_{\mathscr{N}}(\mathbf{x}; u_{NN}(\theta)) \right\|_p$ and $\left\| r^{(j)}_{\mathscr{C}}(\mathbf{x}; u_{NN}(\theta)) \right\|_p$ denote the p-norm of residuals, $\lambda^{(i)}_{\mathscr{N}}$ and $\lambda^{(j)}_{\mathscr{C}}$ serve as weight functions controlling the loss interplay within and across different terms. During each iteration, the integral terms within the loss function $\mathscr{L}_{\text{res}}(\theta)$ are generally approximated using either a regular or Quasi-Monte Carlo method [17], with samples drawn from the independent variables $\mathbf{x}$ as a batch. Fig. 2 illustrates the learning process of a typical PINN. The inputs to the neural network include the spatial coordinates $\mathbf{x} = [x, y, z, t]$ of a point cloud and realizations from the parametric space $(p_1, \cdots, p_n)$, which are mapped to the quantities of interest $u(\mathbf{x}) = [u_1, u_1, \cdots, u_n]^T$ through a multi-layer perceptron (MLP) with nonlinear activation functions. The derivatives are computed and utilized to formulate the residuals of the governing equations in the loss function. This loss function generally comprises multiple terms weighted by different coefficients, including residuals associated with the output, boundary and initial condition information.

The primary objective of this research is to develop a physics-informed neural network (PINN) model capable of simulating conjugate heat transfer by integrating fundamental physical principles about flow dynamics and heat transfer into the neural network framework. The study aims to investigate the robustness of PINNs in accurately predicting fluid flow behaviors and temperature distributions within complex geometric structures with complicated boundary conditions, thereby contributing to advancements in thermal management strategies for various engineering applications.

2. Methodology

In this study, the numerical simulation is conducted by utilizing the conjugate heat transfer PINN-model to evaluate the cooling performance of MMC for high-power IGBTs. In training PINNs for problems with symmetrical geometry and physical quantities, reducing the computational domain and utilizing symmetry boundaries can accelerate training, reduce memory usage, and enhance accuracy in certain scenarios. This complexity reduction is achieved by simplifying the calculation domain through geometric symmetry considerations, as depicted

in Fig. 3. The geometric model consists of two primary regions: the cooling air and the heat sink, which includes both the base and fin elements. Introducing the cooling air at an initial temperature of 20 °C through the inlet situated at the top left of the geometry, it travels through the system, absorbing heat dissipated by the IGBT module as it traverses the fin structure. Detailed physical properties of both the cooling air and the metal materials constituting the heat sink are outlined in Table 1.

2.1. Governing equations and boundary conditions

Given that the characteristic Reynolds number of the fluid flow is less than 10, the governing equations portray the fundamental principles represent the conservation of mass and momentum for an incompressible, Newtonian fluid in three dimensions as:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0, \tag{5}$$

$$\begin{cases} u\dfrac{\partial u}{\partial x}+v\dfrac{\partial u}{\partial y}+w\dfrac{\partial u}{\partial z}=-\dfrac{1}{\rho_f}\dfrac{\partial p}{\partial x}+\nu\left(\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}+\dfrac{\partial^2 u}{\partial z^2}\right) \\ u\dfrac{\partial v}{\partial x}+v\dfrac{\partial v}{\partial y}+w\dfrac{\partial v}{\partial z}=-\dfrac{1}{\rho_f}\dfrac{\partial p}{\partial y}+\nu\left(\dfrac{\partial^2 v}{\partial x^2}+\dfrac{\partial^2 v}{\partial y^2}+\dfrac{\partial^2 v}{\partial z^2}\right) \\ u\dfrac{\partial w}{\partial x}+v\dfrac{\partial w}{\partial y}+w\dfrac{\partial w}{\partial z}=-\dfrac{1}{\rho_f}\dfrac{\partial p}{\partial z}+\nu\left(\dfrac{\partial^2 w}{\partial x^2}+\dfrac{\partial^2 w}{\partial y^2}+\dfrac{\partial^2 w}{\partial z^2}\right) \end{cases}, \tag{6}$$

where $u, v,$ and w denote the velocity field components along the respective axes; ρ_f and ν are the density and the kinematic viscosity of the fluid; p stands for pressure.

The equations governing energy conservation within the fluid and solid domains are presented as follows:

$$u\frac{\partial T_f}{\partial x}+v\frac{\partial T_f}{\partial y}+w\frac{\partial T_f}{\partial z}-\alpha_f\left(\frac{\partial^2 T_f}{\partial x^2}+\frac{\partial^2 T_f}{\partial y^2}+\frac{\partial^2 T_f}{\partial z^2}\right)=0, \tag{7}$$

$$k_s\left(\frac{\partial^2 T_s}{\partial x^2}+\frac{\partial^2 T_s}{\partial y^2}+\frac{\partial^2 T_s}{\partial z^2}\right)=0, \tag{8}$$

where T_f and T_s denote the temperatures of the fluid and solid, respectively; α_f and k_f represent the thermal diffusivity and the thermal conductivity of the fluid; while k_s signifies the thermal conductivity of the solid.

The inlet velocity is determined by a velocity profile:

$$w_i=\frac{6\overline{w}}{(y_2-y_1)^2}(y-y_1)(y+y_1-2y_2), \tag{9}$$

where $\overline{w}$ is the average velocity, the variable y represents the coordinate along the y -axis, y_1 and y_2 respectively denote the lower and upper limits of the inlet coordinate. The outlet boundary condition is governed by a prescribed pressure $p_o = 0$. For symmetry boundaries, the conditions are $\mathbf{u} \bullet \mathbf{n} = 0$, and $\frac{\partial p}{\partial n} = 0$, where $\mathbf{n}$ is the unit normal vector of the plane. The remaining walls are designated as non-slip walls with $\mathbf{u} = 0$. For the thermal boundary conditions, the temperature at the inlet is specified as $T_i = 293.15$ K. On the bottom wall, a boundary heat source is defined as $Q_b = k_s\frac{\partial T}{\partial n}$, where $\frac{\partial T}{\partial n} = 80$ K/m denotes the normal temperature gradient. For the interface between fluid and solid, both Dirichlet and Neumann boundary conditions are enforced as follows:

$$\begin{cases} T_f = T_s \\ k_f\dfrac{\partial T_f}{\partial n}=k_s\dfrac{\partial T_s}{\partial n} \end{cases}. \tag{10}$$

In addition, the normal temperature gradients at the outlet and all the channel walls are prescribed as $\frac{\partial T}{\partial n} = 0$.

2.2. PINNs framework

The multi-layer perceptron (MLP) represents a foundational neural network architecture characterized by its hierarchical arrangement of layers. Information propagation within an MLP occurs in a forward manner, commencing from the input layer, traversing through one or more hidden layers, and ultimately terminating at the output layer. In this investigation, a MLP neural network comprising 6 layers is deployed, with each layer comprising 512 neurons interconnected by weighted connections. Throughout the training procedure, these connection weights undergo adjustments to minimize the disparity between predicted outputs and actual target values. The utilization of MLPs within PINNs offers a distinct advantage owing to their capacity to effectively capture intricate nonlinear relationships existing between input and output variables, all while incorporating imposed physical constraints or principles [18].

In this numerical investigation, by assuming the fluid properties remain constant with temperature variations, the case is treated as a one-way coupling between heat transfer and fluid flow equations. To

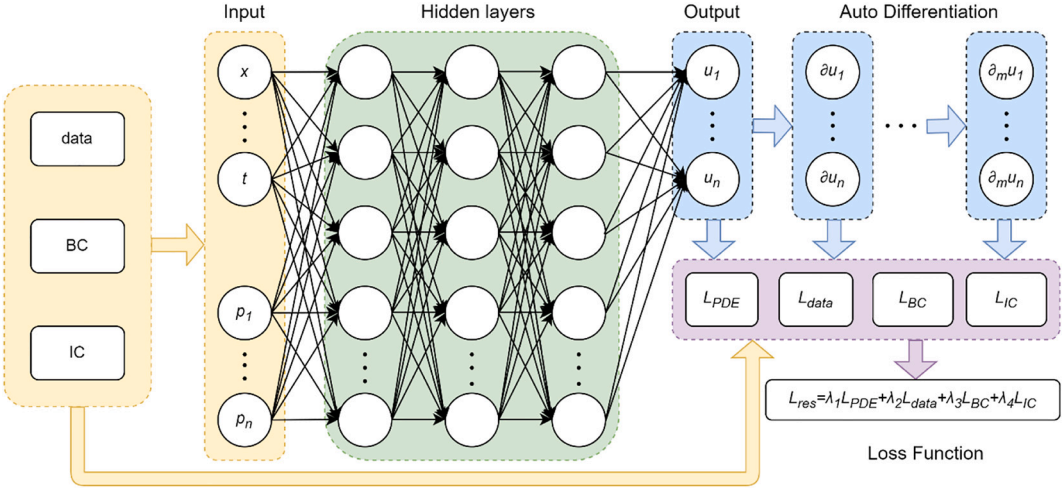


Fig. 2. Schematic of the structure of a physics-informed neural network (PINN) solver. The inputs to the neural network include the spatial coordinates $\mathbf{x} = [x, y, z, t]$ of a point cloud and realizations from the parametric space $(p_1, \cdots, p_n)$, which are mapped to the quantities of interest $u(\mathbf{x}) = [u_1, u_1, \cdots, u_n]^T$ through a multi-layer perceptron (MLP) with nonlinear activation functions. Automatic differentiation (AD) is employed to compute the derivatives, which are then utilized to formulate the residuals in the loss function.

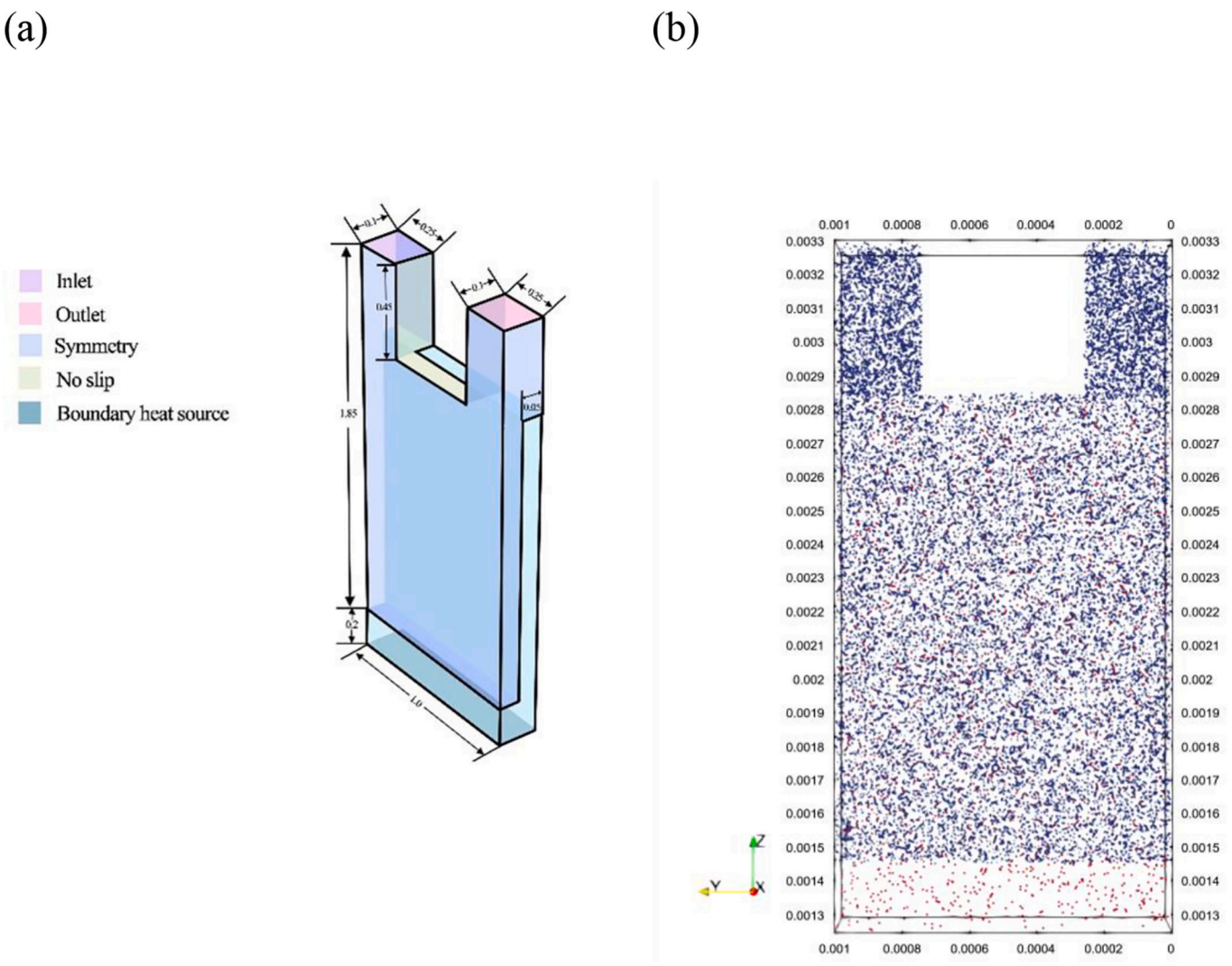


Fig. 3. Figures of (a) the geometric model of calculation regions and the applied boundary conditions and (b) the distribution of sampling points from the main view. Blue points denote the points within the fluid domain $\mathscr{D}_f$, while red points represent the points of the solid domain $\mathscr{D}_s$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

alleviate complexity, the PINN is partitioned into two sub-PINNs, each dedicated to solving the flow field and the temperature fields of the fluid and solid. This partitioning facilitates the segmentation of the loss function, expediting training while preserving efficiency. Fig. 4 illustrates the schematic of the sub-PINNs and the communication channels for computing losses. The pressure (p) and the velocity field components (u, v, and w) are approximated with the flow model trained to convergence. Their derivatives are then calculated employing automatic differentiation (AD), which offers a hybrid method [13] for derivative calculation, maintaining track of derivative values of expressions and enabling combination through the chain rule to derive the original expression's derivative. The neural network is trained by minimizing the mean square error (MSE) of the residual terms about the governing equations and boundary conditions as $L_{total,flow} = L_{PDE,flow} + L_{BC,flow}$, where the residual terms of PDEs are considered as unsupervised learning, while the boundary conditions are considered as supervised learning.

Table 1
Thermal physical properties of materials in fluid and solid regions.

Property	Air	Aluminum
Density / $\left(\frac{kg}{m^3}\right)$	1.1173	
Kinematic viscosity / $(\frac{m^2}{s})$	1.725×10^{-5}	
Thermal diffusivity / $(\frac{m^2}{s})$	2.4534×10^{-5}	8.3×10^{-5}
Thermal conductivity / $(\frac{W}{m \cdot K})$	0.0276	201

The loss function of the flow sub-PINN concerning the PDEs is derived from the residuals corresponding to Eqs. (5) and (6) as:

$$L_{PDE,flow} = \frac{1}{N_{NS}} \sum_{k=1}^{N_{NS}} \sum_{j=1}^{4} \left| f_{NS,j}(u_i, \partial u_i, \partial^2 u_i) \right|^2 = L_{mass} + L_{x-momentum} + L_{y-momentum} + L_{z-momentum}, \quad (11)$$

where N_{NS} represents the number of residual points corresponding to the Navier-Stokes equations, including the mass conservation equation and the momentum conservation equations; $f_{NS,j}$ denote the functions corresponding to the residual terms of the governing equations; $u_i = [u, v, w, p]^T$ indicates the velocity components on x-axis, y-axis, z-axis, and the pressure; $L_{x-momentum}$, $L_{y-momentum}$, $L_{z-momentum}$, and L_{mass} denote the values of loss associated with the governing equations.

The loss associated with the boundary conditions of the flow sub-PINN is given by:

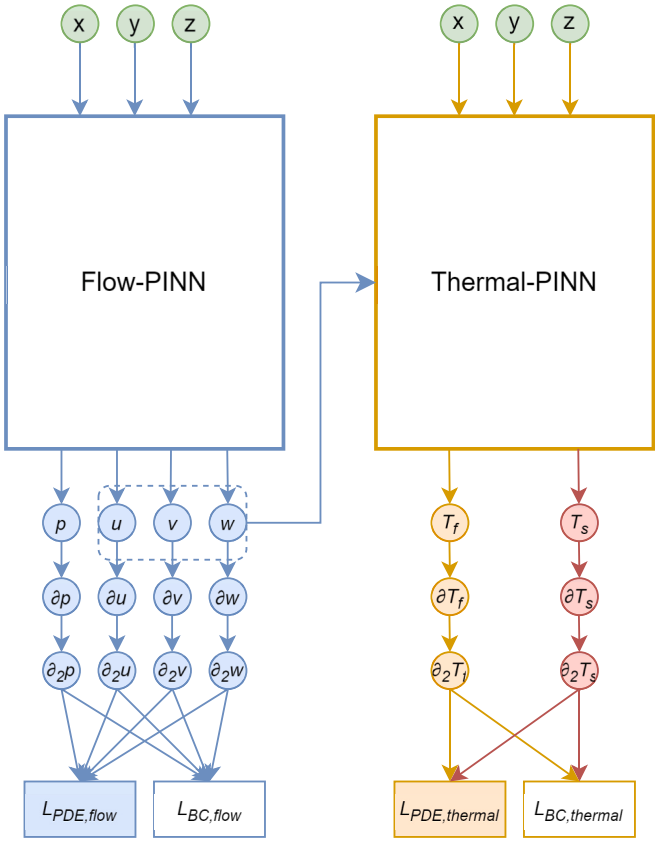


Fig. 4. Schematic of sub-PINNs and communication channels for computing losses. p denotes pressure; u, v, and w represent velocity field components along the respective axes; T_f and T_s depict temperature distributions in fluid and solid domains; $L_{PDE,flow}$ and $L_{PDE,thermal}$ denote the loss functions of the flow-PINN and thermal-PINN concerning the governing equations; $L_{BC,flow}$ and $L_{BC,thermal}$ are the loss functions of the flow-PINN and thermal-PINN concerning the boundary conditions.

$$L_{BC,flow} = \frac{1}{N_{BC,flow}} \sum_{k=1}^{N_{BC,flow}} \sum_{i=1}^{4} \left(\left| u_{NN,i} - u_i \right|^2 + \left| \partial u_{NN,i} - \partial u_i \right|^2 \right) = L_{BC,u} + L_{BC,v} + L_{BC,w} + L_{BC,p} + L_{BC,n}, \tag{12}$$

where $N_{BC,flow}$ is the number of sampling points chosen randomly on the boundary; u_{NN} indicates the output values of the neural network; $L_{BC,u}$, $L_{BC,v}$, $L_{BC,w}$, and $L_{BC,p}$ denote the boundary loss related to u, v, w, and p; while $L_{BC,n}$ indicates the loss term related to the normal gradient of values on the boundary.

Subsequently, the training of the thermal neural network is initiated using the pre-trained flow model, enabling simultaneous resolution of temperature distributions in fluid ($\mathscr{D}_f$) and solid ($\mathscr{D}_s$) domains. The loss function of thermal-PINN is defined as:

$$L_{total,thermal} = L_{PDE,thermal} + L_{BC,thermal}, \tag{13}$$

where the loss function concerning the PDEs is established as:

$$L_{PDE,thermal} = \frac{1}{N_{hf}} \sum_{k=1}^{N_{hf}} \left| f_{hf}(u_i, T_f) \right|^2 + \frac{1}{N_{hs}} \sum_{k=1}^{N_{hs}} \left| f_{hs}(T_s) \right|^2 = L_{hf} + L_{hs}, \tag{14}$$

where N_{hf} and N_{hs} represent the node numbers of heat transfer equations of the fluid and solid domain, f_{hf} and f_{hs} denote the loss functions corresponding to the residual terms in the governing equations, L_{hf} and L_{hs} denote the loss functions concerning the convective heat transfer equation of the fluid domain and the heat conduction equation within the solid domain, respectively. The boundary loss function concerning the thermal field is

$$\begin{aligned} L_{BC,thermal} = {} & \frac{1}{N_{BC,tf}} \sum_{k=1}^{N_{BC,tf}} \left(\left| T_{f,NN} - T_f \right|^2 + \left| \partial T_{f,NN} - \partial T_f \right|^2 \right) \\ & + \frac{1}{N_{BC,ts}} \sum_{k=1}^{N_{BC,ts}} \left(\left| T_{s,NN} - T_s \right|^2 + \left| \partial T_{s,NN} - \partial T_s \right|^2 \right) \\ = {} & L_{BC,T_f} + L_{BC,n,T_f} + L_{BC,n,T_s} + L_{interface,Dirichlet} + L_{interface,Neumann}, \end{aligned} \tag{15}$$

where $N_{BC,hf}$ and $N_{BC,hs}$ represent the node numbers on the boundary, L_{BC,T_f} is the boundary loss terms concerning T_f, L_{BC,n,T_f} and L_{BC,n,T_s} are the boundary loss about the normal gradient of T_f and T_s, and $L_{interface,Dirichlet}$ and $L_{interface,Neumann}$ denote the loss on the fluid-solid interface con-

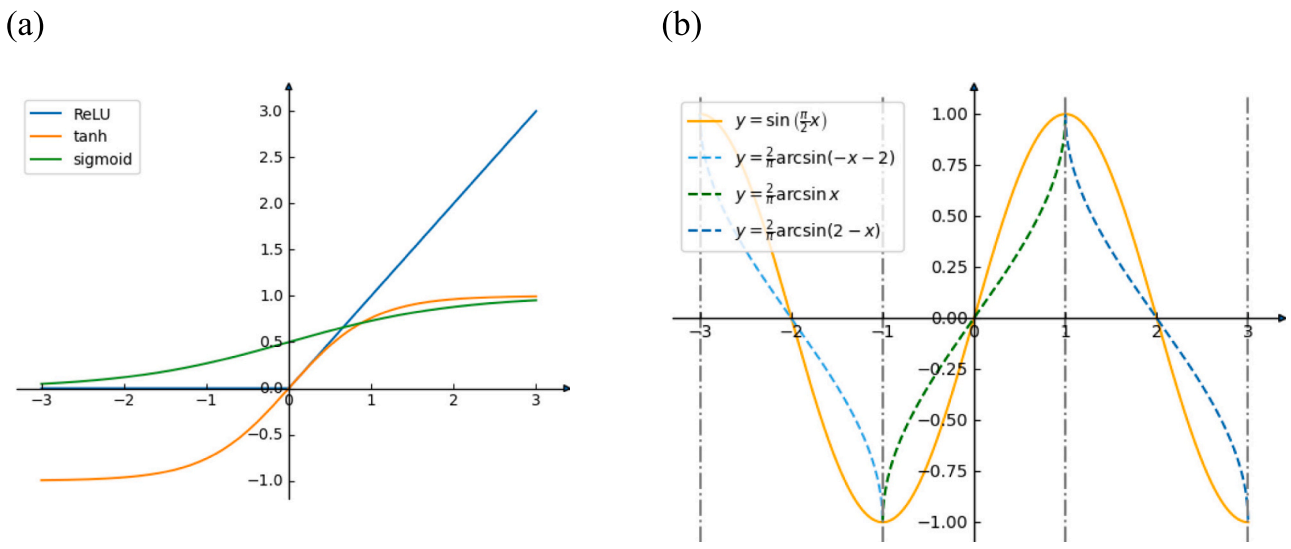


Fig. 5. Graphs of activation functions: (a) ReLU, tanh, and sigmoid; and (b) $y = sin\left(\frac{\pi}{2}x\right)$, and its three reciprocal bijections $y = \frac{2}{\pi}arcsin(-x-2)$, $y = \frac{2}{\pi}arcsinx$, and $y = \frac{2}{\pi}arcsin(2-x)$.

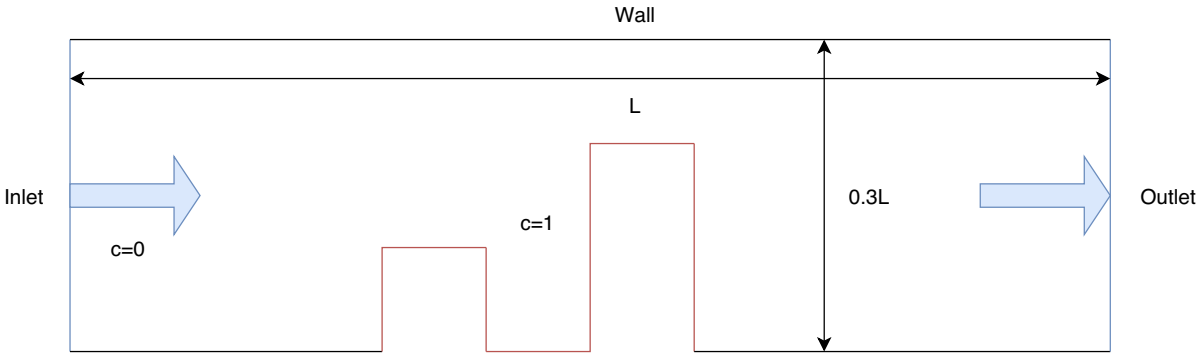


Fig. 6. The geometric structure of the 2D computational domain employed for model validation and numerical simulation. The domain consists of a rectangular flow channel where fluid enters from the inlet on the left side with a prescribed constant velocity and an initial dimensionless temperature of $c = 0$. As the fluid progresses through the channel, it navigates around two embedded objects, each maintained at a dimensionless temperature of $c = 1$. The flow continues toward the outlet located on the right side of the channel.

strained by the Dirichlet and Neumann boundary conditions.

2.3. Activation function

Training an MLP involves forwarding the inputs through the network, computing each neuron's output by applying an activation function to the weighted sum of its inputs as depicted in Eq. (2). As shown in Fig. 5a, common activation functions include sigmoid, tanh, and Rectified Linear Unit (ReLU). While each activation function has its own characteristics and suitability for different tasks, ReLU is often preferred in PINNs due to its computational efficiency and ability to mitigate the vanishing gradient problem, especially in deep networks. However, ReLU-based MLPs, being piecewise linear, have zero second derivatives everywhere, thereby limiting their capability to model information in higher-order derivatives. Although alternative activations like tanh can represent higher-order derivatives, their derivatives often exhibit poor behavior and struggle to capture fine details. Sitzmann et al. [19] proposed a neural network architecture for implicit neural representations that uses the sine as the periodic activation function of the first layer as:

$$\phi_i(x_i) = sin(W_i x_i + b_i). \tag{16}$$

Initialization scheme plays a pivotal role in maintaining inter-layer distributions. Utilizing a sine nonlinearity as input generates a reciprocal sine distribution (shown in Fig. 5b), which preserves the distribution of activations allowing deep architectures to be constructed and trained effectively. This distribution ensures that inputs activated by sine functions adhere to a normal distribution, promoting effective network convergence. Additionally, scaling the first layer of the network by a factor of $\omega_0 = 30$ enables it to span multiple periods of the sine function, further promoting rapid and robust convergence. Previous studies [20–23] indicate that the activation function performs effectively in solving boundary value problems. This process is further aided by the Adam optimizer [24], which updates the weights and biases to ensure a decrease in the total loss of the PINN. The weight matrices of the network are drawn from a uniform distribution as $W_i \sim U\left(-\sqrt{6/n}, \sqrt{6/n}\right)$, ensuring that the input to each sine activation is

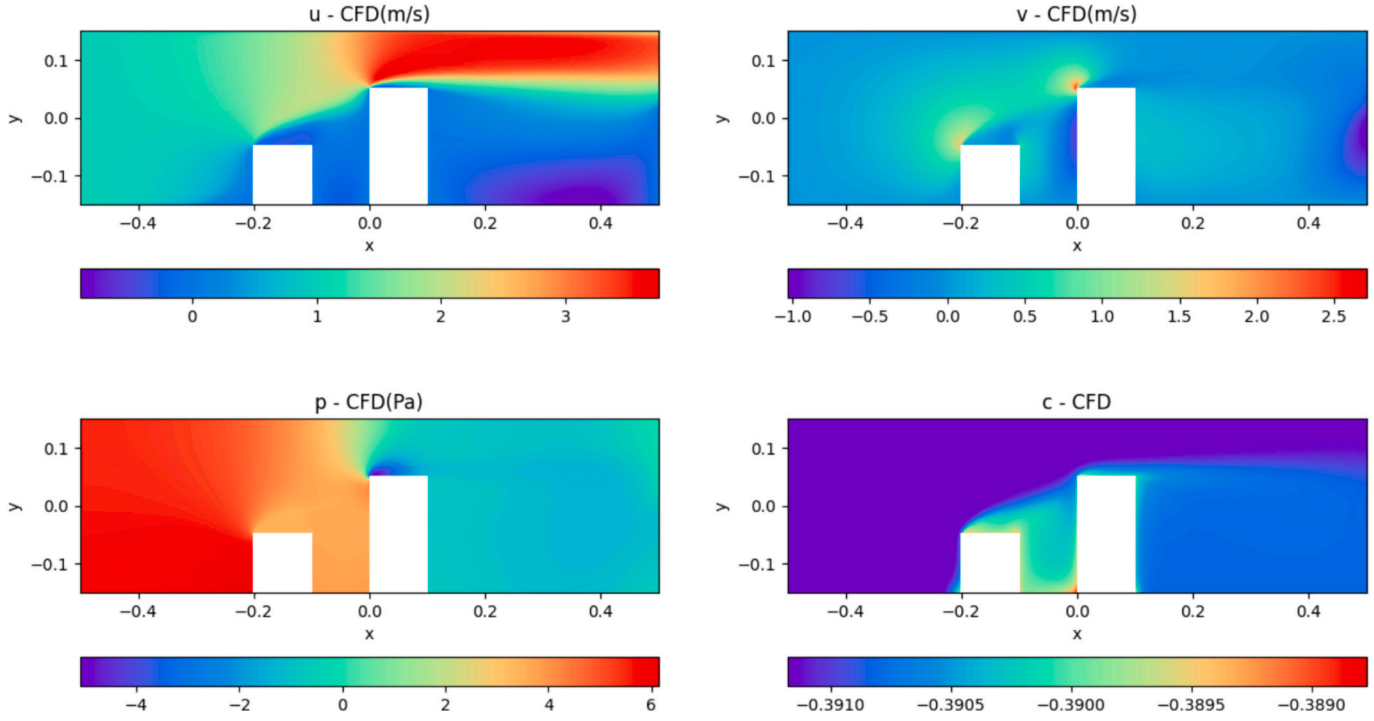


Fig. 7. Velocity components (u and v), pressure (p) and normalized temperature (c) obtained by CFD.

normally distributed with a standard deviation of 1.

3. Model validation

In this chapter, the efficacy of the heat transfer model is evaluated by applying it to a 2D steady-state forced convection problem within an enclosure. The setup, as depicted in Fig. 6, involves a channel where the fluid is introduced with a constant axial velocity of $u = 1m/s$ and an initial scaled temperature of $c_i = 0$. The scaled temperature c is defined as $c = \frac{T}{T_i} - 1$, where $T_i = 293.15K$ is the temperature of the fluid at the inlet. Thus, the scalar transport equation of the dimensionless

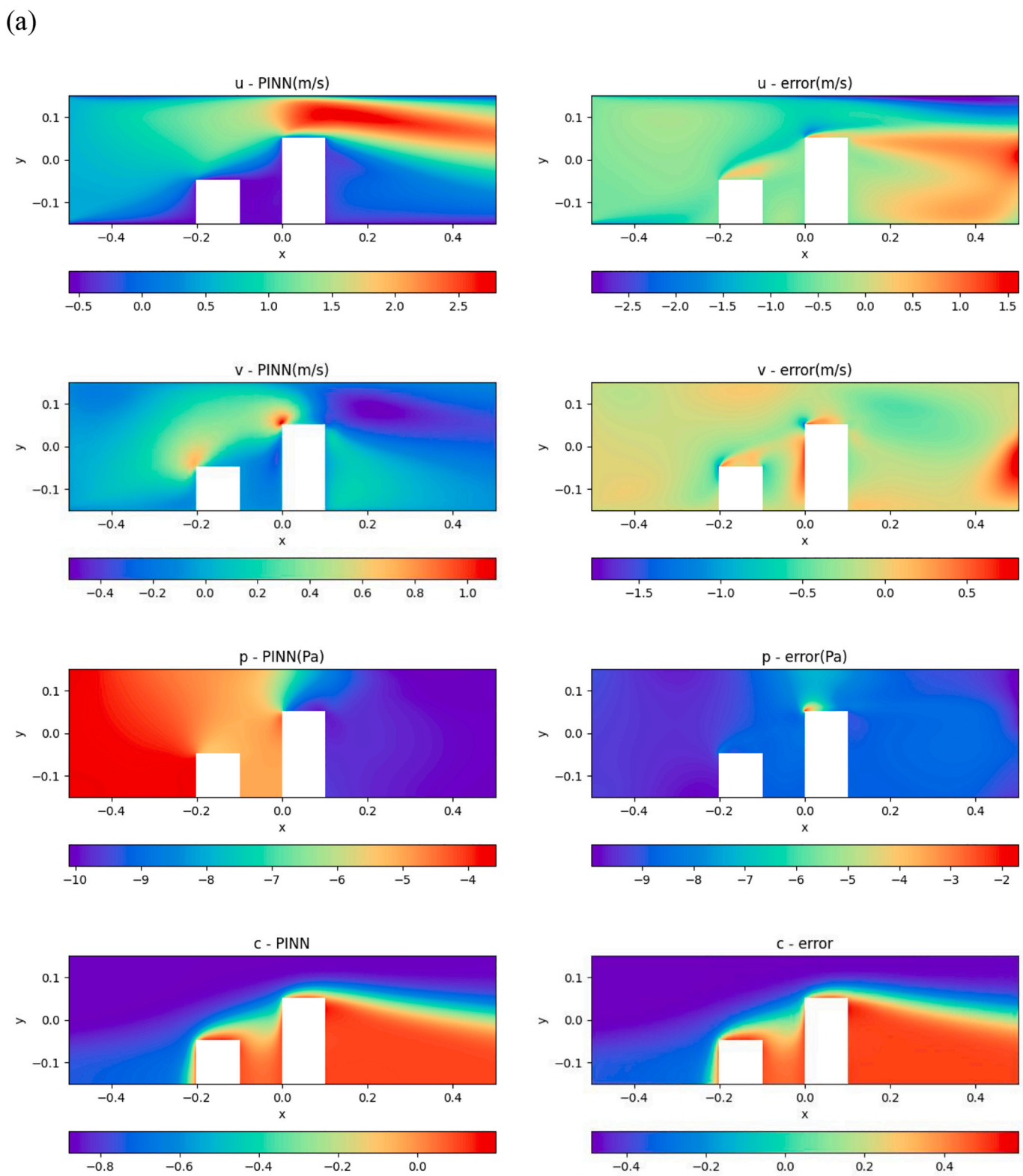


Fig. 8. Distributions of results and errors obtained from PINNs for different point cloud densities: (a) $n = 1200$, (b) $n = 2400$, (c) $n = 3600$, (d) $n = 4800$, and (e) $n = 6000$.

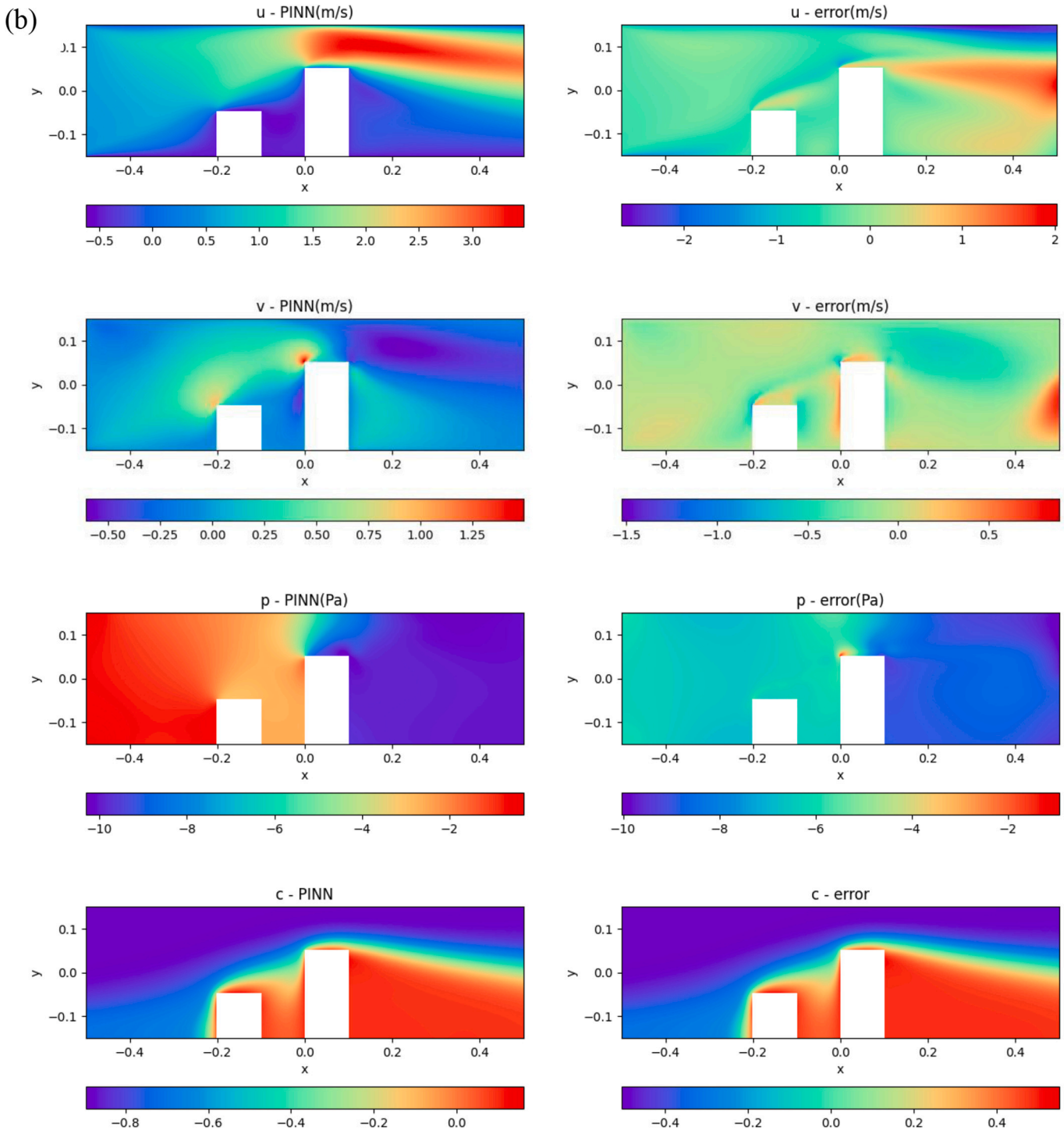


Fig. 8. (*continued*).

(c)

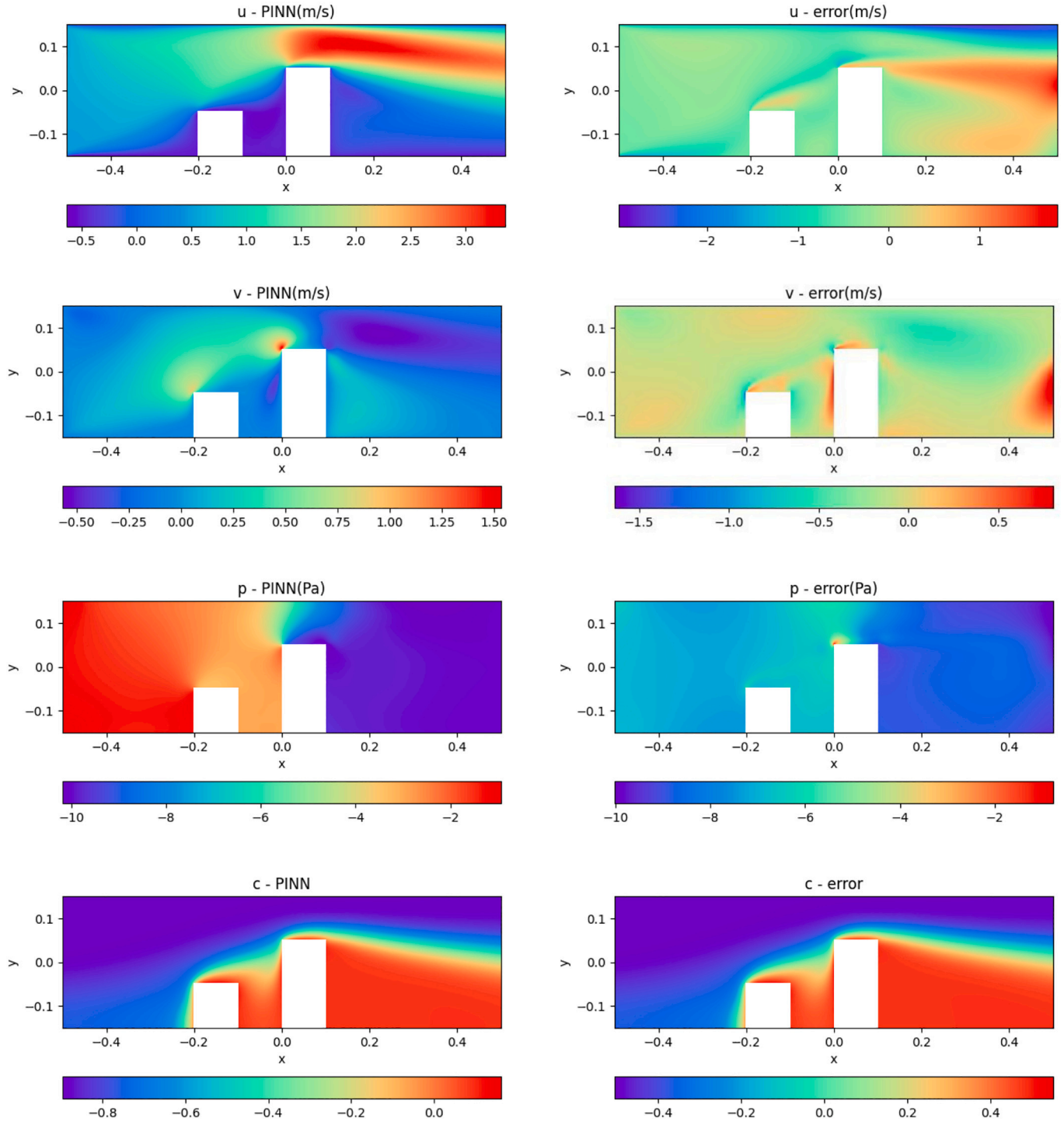


Fig. 8. (*continued*).

(d)

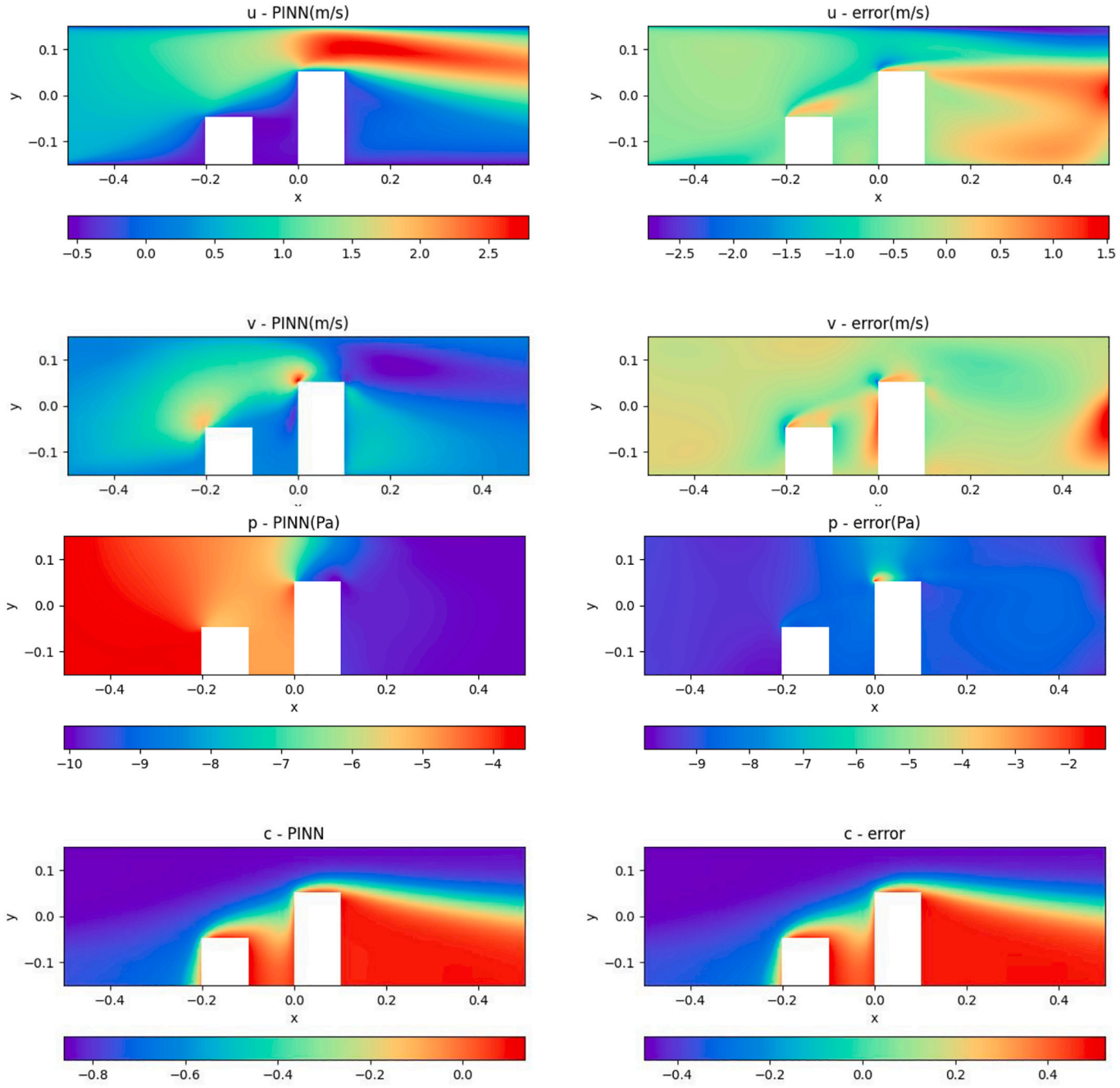


Fig. 8. (*continued*).

(e)

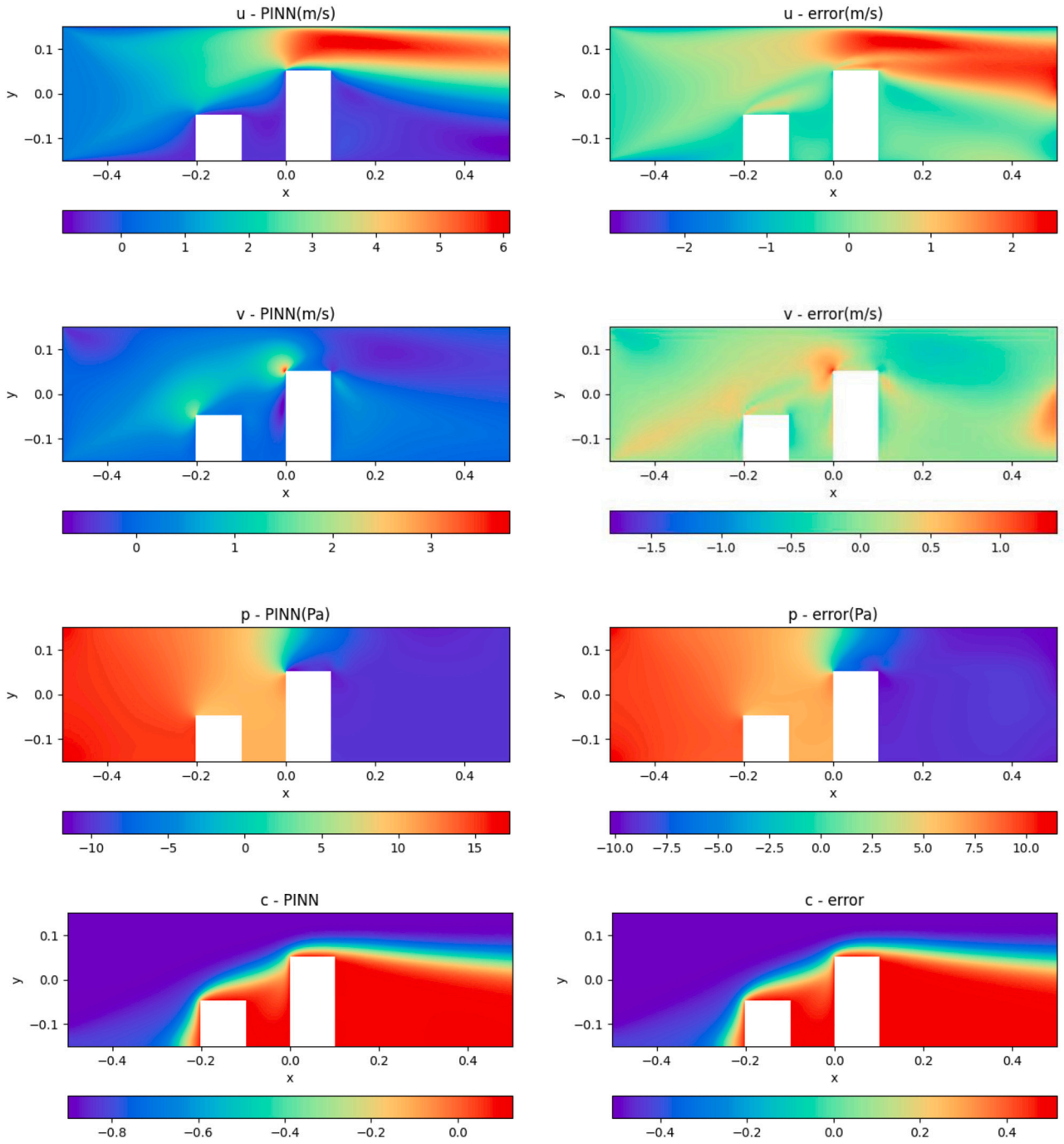


Fig. 8. (*continued*).

Table 2
The RRMSE and R^2 with point clouds density from $n = 1200$ to $n = 6000$.

n	RRMSE	R^2
1200	1.0601	0.3472
2400	1.0414	0.3700
3600	0.9996	0.4195
4800	0.9638	0.4602
6000	0.9211	0.5071

temperature is

$$u\frac{\partial c}{\partial x}+v\frac{\partial c}{\partial y}=\alpha_f\left(\frac{\partial^2 c}{\partial x^2}+\frac{\partial^2 c}{\partial y^2}\right). \tag{17}$$

Two objects within the channel are maintained at a fixed temperature (with Dirichlet boundary $c = 1$). The walls of the channel are treated as adiabatic with no heat flux through them *as* $\frac{\partial c}{\partial n} = 0$.

Physics-Informed Neural Networks (PINNs) are an innovative approach that leverages neural networks to solve partial differential equations (PDEs) by embedding physical laws directly into the loss

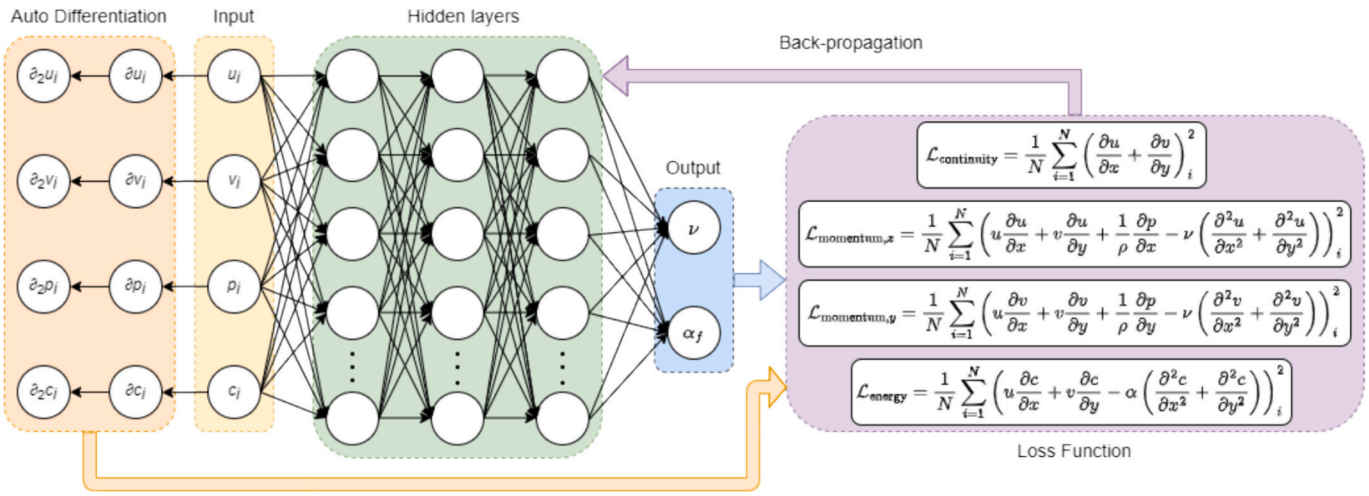


Fig. 9. Schematic of the PINN for the inverse problem to predict critical physical parameters, specifically fluid viscosity ν and thermal diffusivity α_f, based on training data generated from OpenFOAM simulations. The input to the network consists of data points (u_i, v_i, p_i, c_i) which represent the velocity components, pressure, and dimensionless temperature fields obtained from the simulation. The training process incorporates the governing equations, including continuity, Navier-Stokes, and advection-diffusion, as embedded constraints in the loss function.

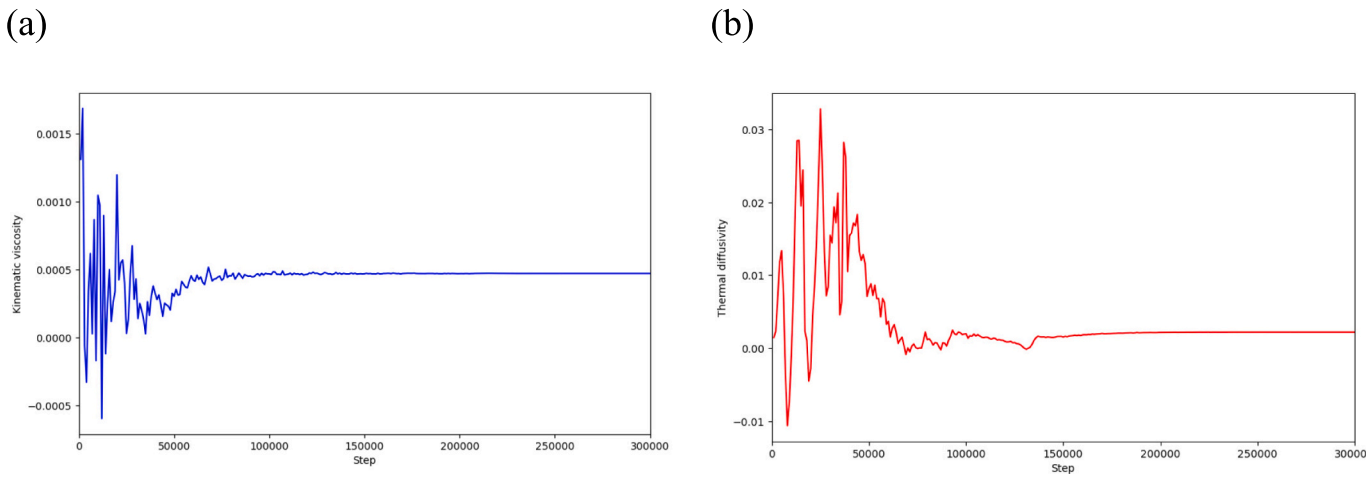


Fig. 10. Variation of predicted kinematic viscosity ν and thermal diffusivity α_f with steps when $u = 1.0m/s$.

function. This study seeks to validate the model's robustness and precision in predicting heat transfer phenomena within a controlled scenario, thereby establishing a benchmark for future assessments of the model's performance in more complex cases.

3.1. Point clouds distribution

The accuracy and robustness of PINNs are contingent upon several factors, notably the density and distribution of point clouds used during training. Point cloud density emerges as a critical characteristic, reflecting the spatial distribution and intensity of the laser point cloud, and directly mirroring the spatial features of the target terrain. A higher point cloud density typically correlates with enhanced resolution and richer information, facilitating more precise extraction and processing of terrain data. However, an excessively high point cloud density can result in significant data redundancy, thereby complicating data processing and analysis. In the context of evaluating the impact of point cloud density on the robustness and accuracy of the PINN model, this study systematically generates nodes within the flow channel's interior domain with varying point cloud densities—specifically, 1200, 2400, 3600, 4000, and 6000 points. This variation allows for a comprehensive analysis of how point cloud density influences the model's performance,

Table 3
Coefficients obtained in inverse problem for different inlet velocities.

$u/(m/s)$	$\nu/(10^{-4}m^2/s)$	$\alpha_f(10^{-3}m^2/s)$
0.5	2.85	0.791
1.0	4.22	2.304
1.5	3.81	3.473
2.0	4.71	2.157
2.5	4.36	1.662
3.0	6.83	1.316

balancing the benefits of increased data resolution against the potential challenges posed by data redundancy.

The velocity components (u and v), pressure (p), and normalized temperature (c) obtained from traditional Computational Fluid Dynamics (CFD) simulations are depicted in Fig. 7. Fig. 8 illustrates the distributions of results and errors obtained from Physics-Informed Neural Networks (PINNs) for varying point cloud densities, each subplot shows the model's prediction results alongside the corresponding error distributions. As point cloud density rises from 1200 to 6000 points, the model's predictions become more precise, with error distributions narrowing and aligning more closely with actual data. This

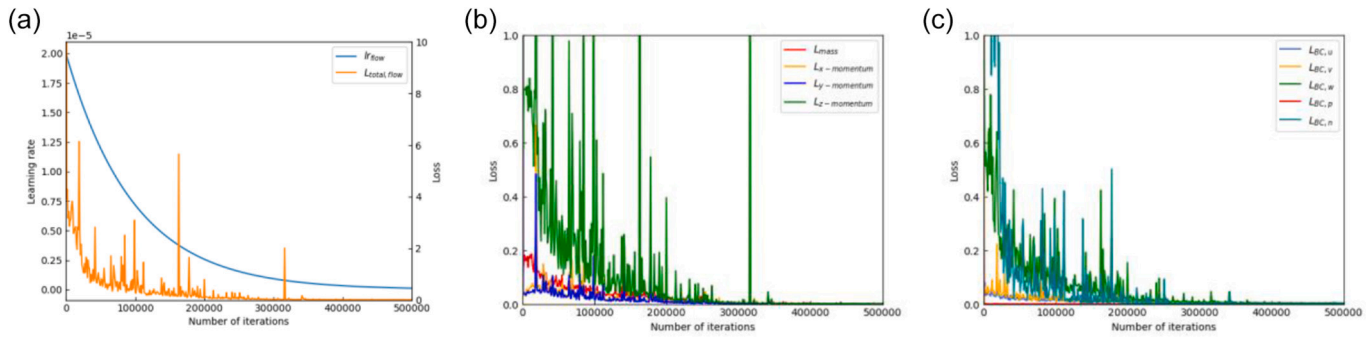


Fig. 11. Figures of the learning rate and loss values during the training of the flow-PINN.

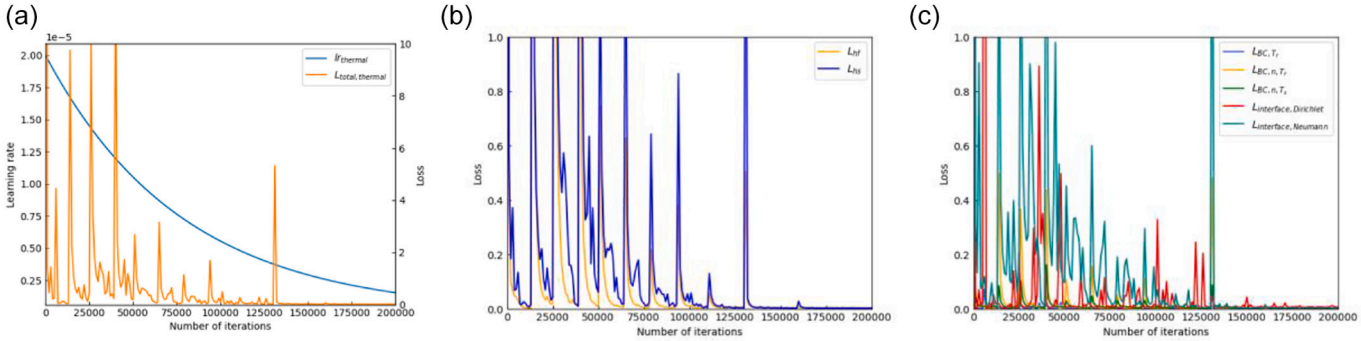


Fig. 12. Figures of the learning rate and loss values during the training of the thermal-PINN.

trend indicates that higher point cloud densities provide the model with more detailed and representative data, which improves its ability to capture complex patterns and reduces prediction errors.

Table 2 presents the Relative Root Mean Squared Error (RRMSE) and R^2 values across varying point cloud densities. RRMSE quantifies the relative discrepancy between predicted and observed values, with a reduction in RRMSE indicating that the model's predictions are increasingly aligning with the actual values. Concurrently, an increase in R^2 reflects the model's enhanced ability to explain the variance in the target variable. A higher R^2 value suggests that the model more effectively accounts for the variability within the data, resulting in less unexplained error. As the point cloud density increases, the model benefits from a richer and more detailed dataset, leading to a significant reduction in relative prediction error and a stronger correlation with the observed data. This trend highlights the critical importance of data quantity and quality in determining model performance, particularly in applications where precision and accuracy are essential.

The results observed in Fig. 8 and Table 2 can be attributed to the fact that a higher point cloud density offers a more comprehensive representation of the underlying data structure. A denser point cloud supplies the model with a larger and more detailed dataset, which enhances its ability to capture complex patterns and subtle nuances within the data. Consequently, the model's performance improves significantly as point cloud density increases, underscoring the importance of dense data in achieving accurate and robust predictions. This highlights the critical role that data quality and quantity in the effectiveness of PINNs.

3.2. Inverse problem

In Physics-Informed Neural Networks, the resolution of inverse problems is facilitated through the integration of neural networks with the governing partial differential equations of the system. With the utilization of PINNs, this study not only approximates the solution of these PDEs but also concurrently infers the unknown parameters within the system. The inverse network relies on the minimization of a composite loss function, which encapsulates the residuals of the PDEs. By fusing the robustness of physics-based modeling with the adaptability of machine learning, PINNs are able to deliver solutions that are both accurate and consistent with the underlying physical laws [25].

Inverse problems, at their core, seek to deduce unknown parameters or inputs of a system by leveraging observed outputs. In the context of heat transfer, recent advancements have highlighted the application of inverse problem-solving techniques to infer critical physical parameters, such as thermal diffusivity, boundary conditions, and source terms, from available data [26–28]. These studies underscore the efficacy of using known governing equations to extract valuable information about the system's hidden characteristics, thereby advancing the ability to model and predict complex physical phenomena with greater precision.

Fig. 9 illustrates the architecture of a PINN designed to solve an inverse problem, specifically for predicting the values of kinematic viscosity ν and thermal diffusivity α_f. The training data for this problem is (u_i, v_i, p_i, c_i) from OpenFOAM simulation and the model is trained to predict (ν, α_f) with the constraints of satisfying the governing equations of continuity, Navier-Stokes and advection-diffusion. This data is then fed into Physics-Informed Neural Networks (PINNs) to predict critical parameters like fluid viscosity and thermal diffusivity. By leveraging the strengths of both data-driven and physics-based approaches, PINNs can accurately model complex physical phenomena and infer essential physical parameters even from limited observational data. By incorporating simulation data alongside physical constraints, this methodology significantly improves the accuracy and reliability of parameter estimation, thereby enhancing its applicability in engineering and scientific research. In this particular example, the parameters used in the simulations are $\nu = 4 \times 10^{-4} m^2/s$ for kinematic viscosity and $\alpha_f = 2 \times 10^{-3} m^2/s$ for thermal diffusivity. The variation of predicted kinematic viscosity and thermal diffusivity when $u = 1.0 m/s$ is shown in Fig. 10. The estimated values for viscosity and thermal diffusivity are shown in Table 3.

The results in Table 3 reveal that the estimated kinematic viscosity ν

and thermal diffusivity α_f vary significantly across different inlet velocities, ranging from 0.5 m/s to 3.0 m/s. The kinematic viscosity ν fluctuates between $2.85 \times 10^{-4} m^2/s$ and $6.83 \times 10^{-4} m^2/s$, while the thermal diffusivity α_f varies from $0.791 \times 10^{-3} m^2/s$ to $3.473 \times 10^{-3} m^2/s$. These results demonstrate the ability of the PINN to adapt and infer critical physical parameters under varying flow conditions. The variation in ν and α_f indicates that the model is sensitive to changes in inlet velocity, which is advantageous for capturing complex flow dynamics and ensuring accurate parameter estimation. This sensitivity suggests that the PINN can effectively model different scenarios and adjust predictions based on the specific conditions of the system. However, the wide range of estimated values also raises concerns about the consistency and reliability of the model, especially at higher inlet velocities. The significant fluctuations in the inferred parameters may indicate that the model is highly dependent on the input data quality or that certain aspects of the flow dynamics are not being fully captured. This variability could potentially lead to less accurate predictions in scenarios with less stable or more complex flow conditions. Further refinement of the model, such as incorporating additional constraints or improving the resolution of the input data, may be necessary to enhance its robustness and consistency across different flow regimes.

4. Results and discussion

The implementation of the PINN model for conjugate heat transfer in a manifold microchannel heat sink is conducted using PyTorch 2.2.1. The training process is performed on an NVIDIA GTX 4090 GPU with 24GB of memory. The training schedule employs an exponential decay learning rate, initialized at 0.00002, with a decay rate of 0.95 and decay steps set to 5000. This strategy aims to optimize model performance by systematically adjusting parameters, monitoring progress, and dynamically adapting as necessary. The learning rates and total loss of the flow-PINN and thermal-PINN are illustrated in Figs. 11a and 12a, respectively. The maximum number of training steps for the flow-PINN is set to 500,000, while for the thermal-PINN, it is set to 200,000.

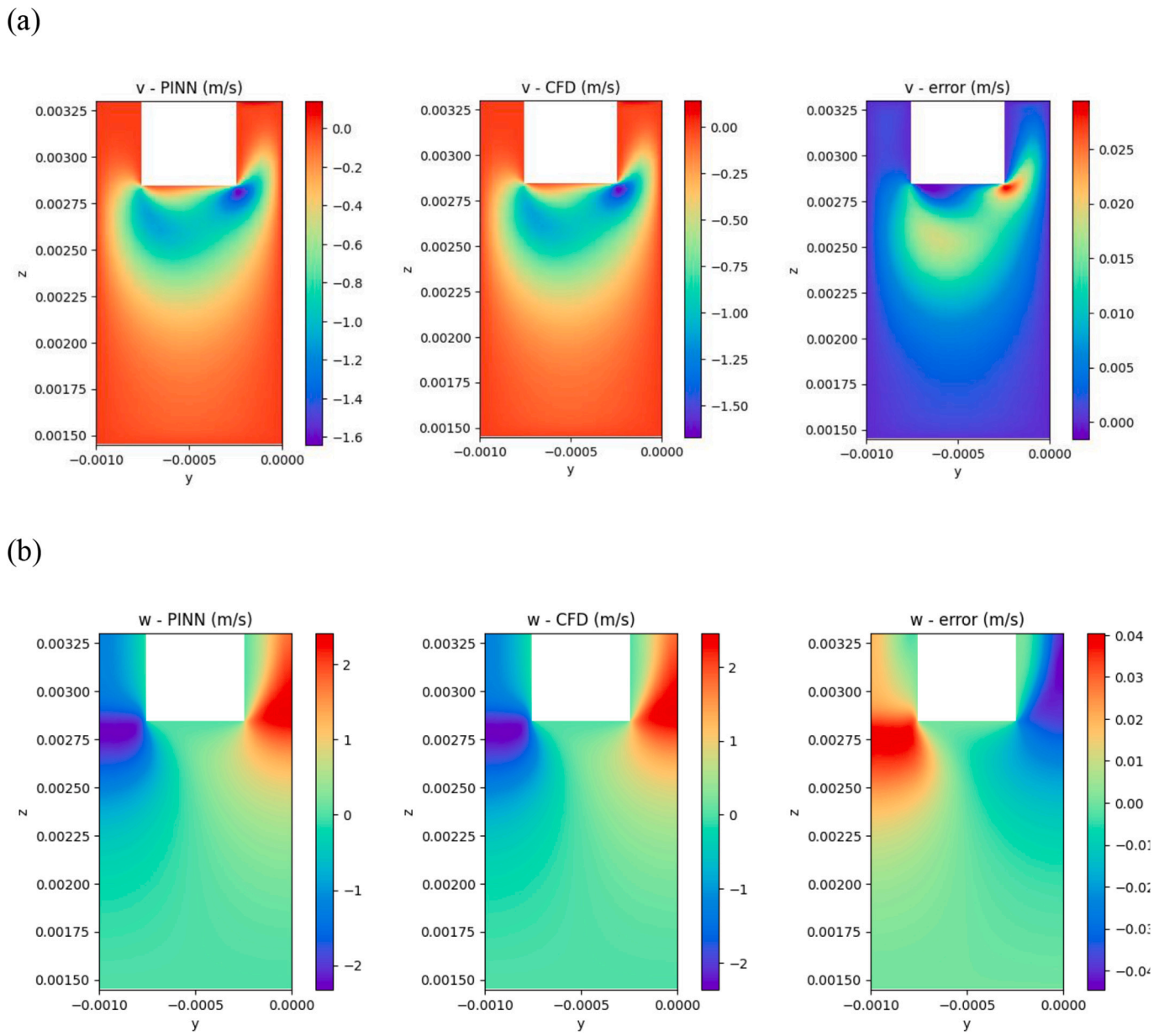


Fig. 13. Comparison of velocity components v, w, pressure p, and fluid temperature T_f obtained from PINN and CFD simulations when $\overline{w} = 0.8 m/s$, viewed from the main perspective.

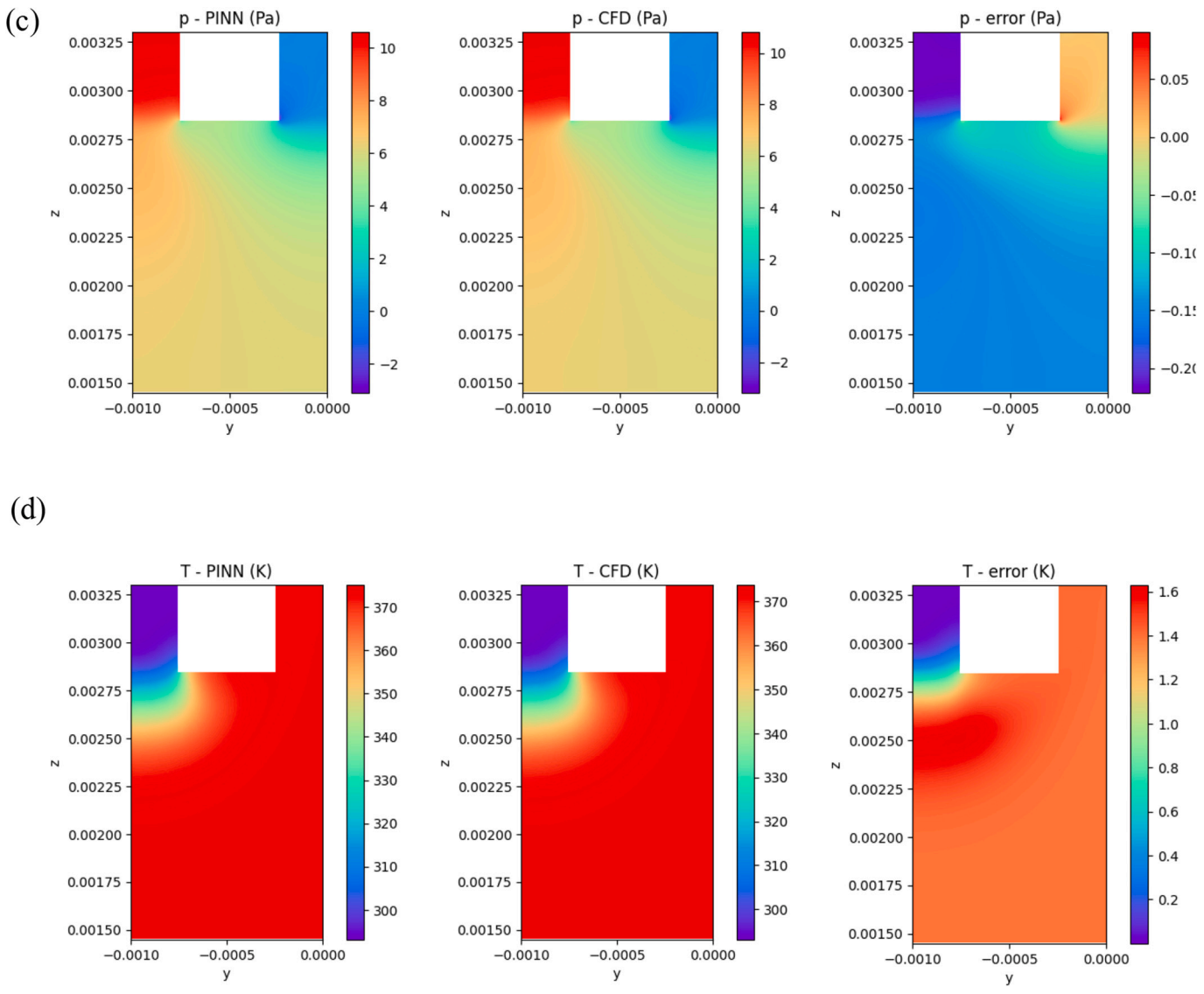


Fig. 13. (*continued*).

4.1. Training loss

Fig. 11b and c display the loss values in the flow-PINN concerning the governing equations and boundary conditions, where L_{mass} represents the loss associated with mass conservation; $L_{x-momentum}$, $L_{y-momentum}$, and $L_{z-momentum}$ denote the loss associated with momentum conservations; $L_{BC,u}$, $L_{BC,v}$, $L_{BC,w}$, and $L_{BC,p}$ denote the boundary loss related to u, v, w, and p; while $L_{BC,n}$ indicates the loss related to the normal gradient terms on the boundary. Throughout the training process, these loss terms exhibit a decreasing trend, and the contributions of $L_{z-momentum}$ and $L_{BC,w}$ to the total loss function are particularly noteworthy. This could be attributed to the prevailing flow direction along the z-axis, which also significantly influences the residuals about momentum conservation.

Fig. 12b and c depict the loss values in the thermal-PINN concerning the governing equations and boundary conditions. The loss functions, including L_{hf} and L_{hs} for the convective heat transfer equation of the fluid domain and the heat conduction equation within the solid domain, as well as various boundary loss terms denoted by L_{BC,T_f}, L_{BC,n,T_f}, L_{BC,n,T_s}, $L_{interface,Dirichlet}$, and $L_{interface,Neumann}$ are presented. Interestingly, despite sharing identical architectures and learning rates with the flow-PINN, the thermal-PINN exhibits a tendency to converge faster. This phenomenon could be attributed to the thermal-PINN having fewer parameters and a simpler structure.

4.2. Comparison with traditional CFD simulations

Fig. 12 illustrates a comparison between the velocity components v, w, pressure p, and fluid temperature T_f obtained from both PINN and CFD simulations, viewed from the main perspective, at an average inlet velocity $\overline{w} = 0.8m/s$, along with the associated errors. For the y-direction velocity component v, the values obtained via PINN are slightly higher than those from CFD. The maximum discrepancy occurs near the narrowing section, the outlet channel, where the error reaches approximately 0.025 m/s. Overall, the average values of w within the fluid domain obtained from PINN and CFD simulations are nearly identical. Notably, the error reaches a maximum of approximately 0.04 m/s near the widening section of the inlet channel and −0.04 m/s near the narrowing section near the outlet channel. According to Fig. 13c, the pressure obtained via PINN is lower near the inlet and higher near the narrowing section compared to CFD results. As shown in Fig. 13d and 14, the most significant variations in fluid temperature occur near the sudden widening of the inlet channel, and the temperature obtained via PINN is slightly higher than that from CFD simulations, with errors ranging from 0 to 1.6 K. Fig. 13 compares the temperature distributions

along the left surface under varying average inlet velocities. The results from PINN and CFD exhibit a close resemblance, indicating good agreement between the two methods. It's noticeable that both fluid and solid temperatures decrease with increasing inlet velocity, reflecting the cooling effect induced by higher fluid flow rates. The fluctuation in temperature gradients is primarily concentrated within the fluid domain, particularly near the upper section of the phase interface, which arises due to the lower conductivity of the fluid compared to the solid.

Fig. 15 compares the pressure drop and IGBT temperature obtained from both PINN and CFD simulations across different average inlet velocities. Both methods demonstrate that the pressure drop increases with higher inlet velocities, while the average IGBT temperature decreases. Notably, the values of pressure drop predicted by PINN are consistently lower than those from CFD simulations, whereas the average IGBT temperatures obtained from PINN are generally higher than those from CFD. The analysis reveals that the minimum error in pressure drop is 0.04 Pa at an average inlet velocity of 0.8 m/s, with subsequent errors increasing, reaching a maximum of 1.14 Pa at an average inlet velocity of 1.4 m/s. The minimum error in the average temperature of IGBT is 6.65 K at an average inlet velocity of 0.4 m/s, with subsequent errors decreasing, reaching a minimum of 0.95 Pa with an inlet velocity of 1 m/s.

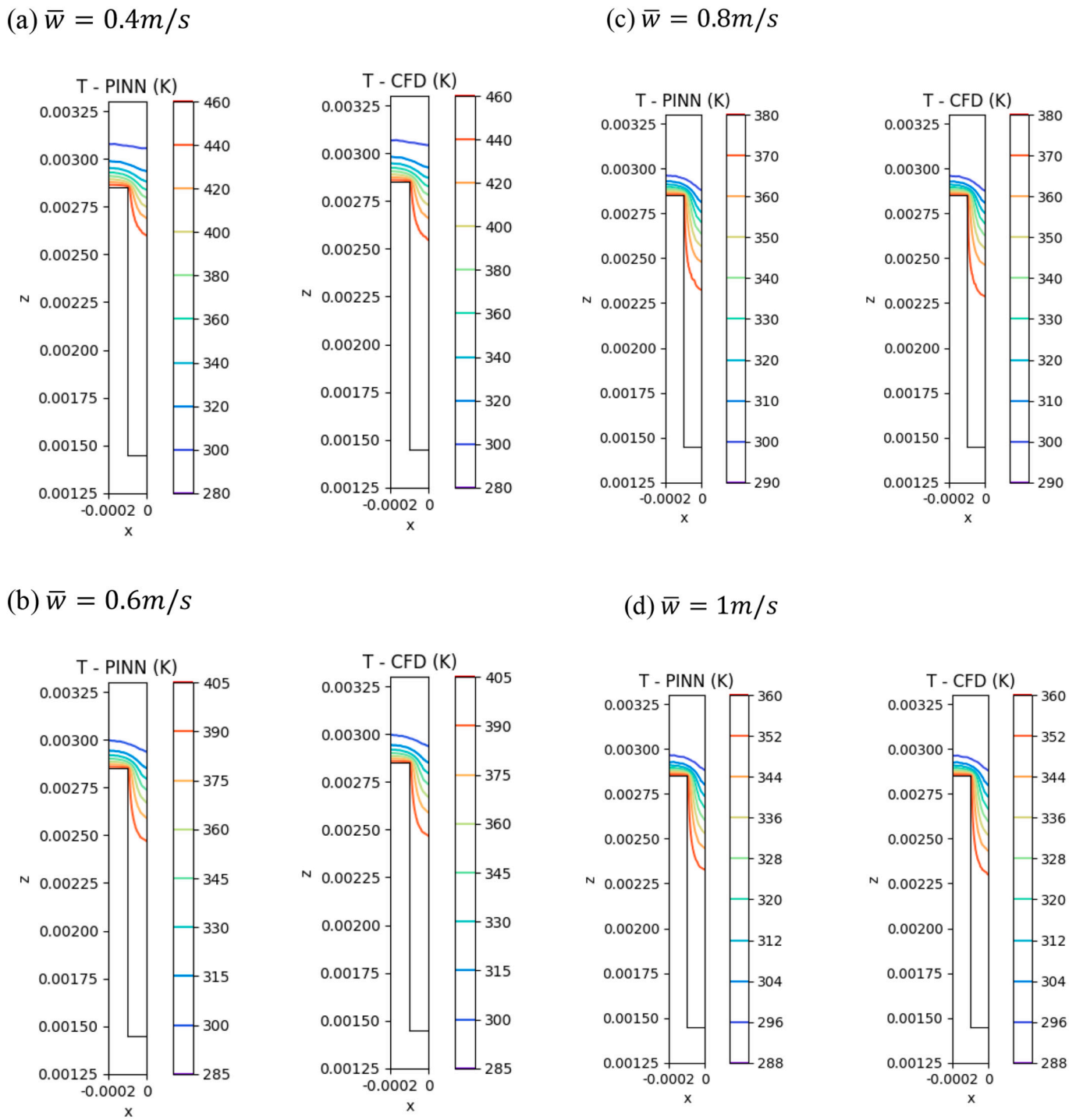
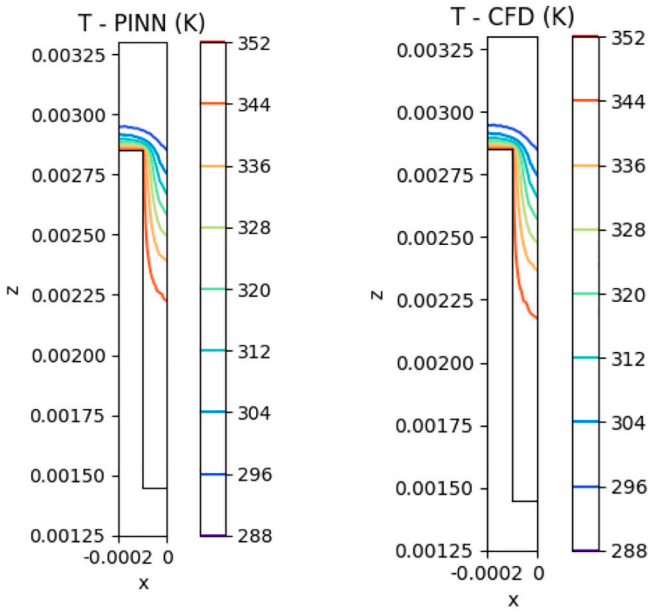


Fig. 14. Comparison of temperature distributions obtained from PINN and CFD simulations, viewed from the left perspective.

(e) $\bar{w} = 1.2m/s$



(f) $\bar{w} = 1.4m/s$

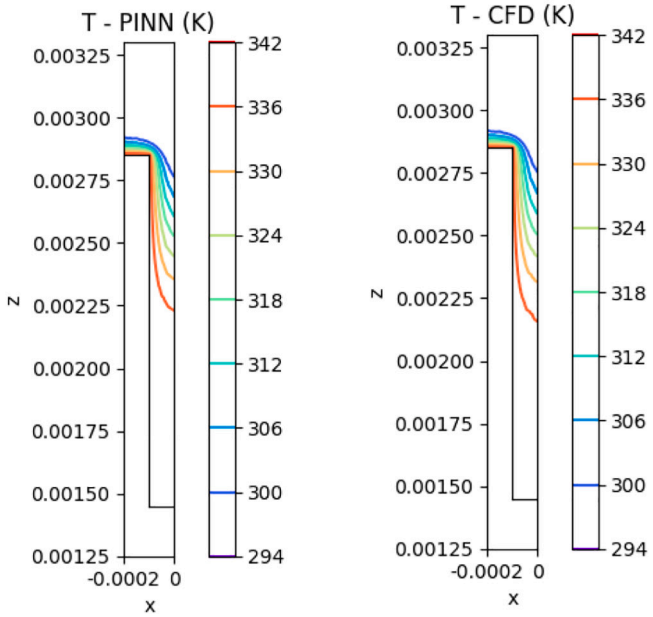


Fig. 14. (*continued*).

According to the results above, sudden changes in geometry, such as the sudden widening or narrowing of the channel, can introduce numerical instabilities in PINN simulations. These instabilities may arise due to abrupt changes in flow patterns or gradients, leading to inaccuracies in the computed flow and temperature fields. In addition, the distribution of point clouds can be a reason since insufficient resolution in channel geometry may lead to inaccuracies in predicting flow and temperature fields, resulting in discrepancies between the two approaches., especially near geometric discontinuities. Overall, the discrepancies near sudden changes in channel geometry highlight the sensitivity of PINNs to geometric complexity and numerical stability considerations. Ensuring adequate grid resolution, capturing geometric details can help mitigate these discrepancies and improve the accuracy of PINN simulations.

Based on the results discussed above, differences between PINN and CFD simulations, particularly near abrupt changes in channel geometry, highlight the sensitivity of PINNs to geometric complexity and numerical stability. These differences may arise due to rapid changes in flow patterns or gradients, leading to inaccuracies in predicted flow and temperature fields. Additionally, the distribution of point clouds is critical, as inadequate resolution near geometric discontinuities can contribute to inaccuracies in predicting fluid flow and temperature

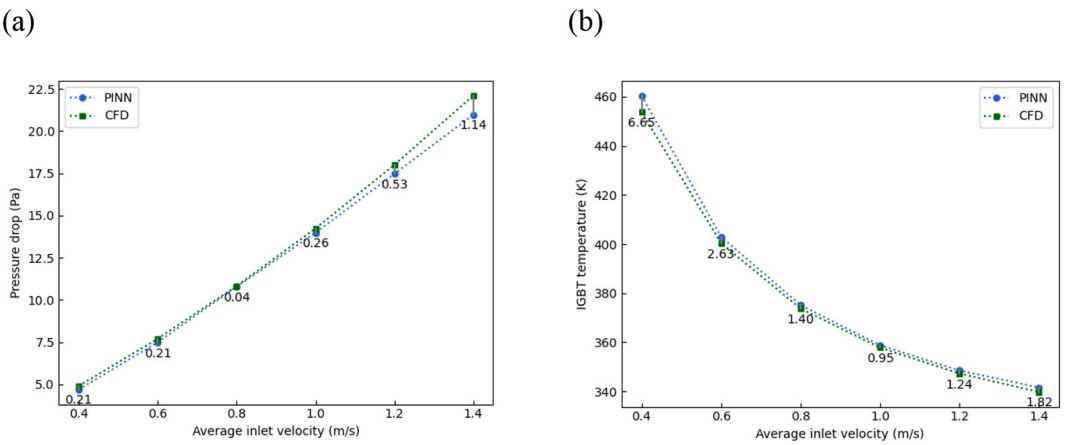


Fig. 15. Comparisons of (a) pressure drop and (b) IGBT temperature obtained from PINN and CFD simulations at varying average inlet velocities.

distribution parameters. To enhance the precision of PINN simulations, it's crucial to ensure sufficient resolution in point cloud distribution and accurately capture geometric details, which ensures more dependable predictions of flow and heat transfer phenomena with complex constraints within complicated geometric structures.

5. Conclusion

This study has developed a multi-physics coupled Physics-Informed Neural Network model designed to predict fluid flow and heat transfer within complex geometries, specifically targeting the thermal management of high-power IGBTs using manifold microchannel heat sinks. By integrating fundamental physical laws directly into the PINN framework, the model represents a significant advancement in simulating complex phenomena under diverse boundary conditions. A key feature of the proposed PINN approach is its dual sub-model structure, each utilizing a 6-layer Multi-Layer Perceptron (MLP) with a sine activation function in the initial layer. This configuration effectively captures high-order derivatives and mitigates the issue of vanishing gradients, which is crucial for achieving accurate simulations. The PINN model was rigorously compared against traditional CFD simulations, with results from both methods showing strong agreement, thereby underscoring the reliability of the PINN approach. Notably, our findings reveal that increasing the inlet velocity results in a decrease in both fluid and solid temperatures, attributable to enhanced cooling effects from higher fluid flow rates. While the PINN model exhibited remarkable consistency with CFD simulations, discrepancies were observed in scenarios involving rapid changes in flow patterns or gradients. These differences underscore the PINN model's sensitivity to geometric complexities and numerical stability—areas that warrant further refinement. Furthermore, this study explored the application of PINNs to inverse problems, an area where traditional CFD methods have not yet been extensively applied. This investigation highlights the potential of PINNs not only in forward simulations but also in expanding the capabilities of CFD to address inverse problems, offering new avenues for research and development in thermal management and beyond.

In the domain of electronic device thermal management, the selection between traditional machine learning prediction algorithms and PINNs is contingent upon the specific problem context and requirements. Machine learning algorithms are particularly effective when large volumes of high-quality data are available, enabling the model to learn intricate patterns and deliver accurate predictions. However, these algorithms may encounter limitations in scenarios characterized by limited data availability, where their purely data-driven nature can lead to predictions that overlook critical physical laws, thus compromising reliability in physics-constrained environments. Conversely, PINNs present a robust alternative by embedding governing physical laws, such as partial differential equations, directly into the neural network framework. This integration enables PINNs to perform effectively even with limited observational data, as the embedded physical constraints guide the model toward solutions that are consistent with established physical principles. Although PINNs entail higher computational costs due to the need to solve these embedded equations during training, they offer enhanced interpretability and reliability in the prediction of physical parameters vital for electronic thermal management. The decision between these methodologies should be informed by considerations of data availability, the necessity of adherence to physical laws, and the computational resources at disposal. For tasks that are predominantly data-driven and supported by abundant data, traditional machine learning may be more suitable. However, in contexts where physical consistency and the accurate modeling of complex thermal phenomena are paramount—particularly in data-constrained environments—PINNs are likely to represent the more advantageous approach. By integrating empirical data with physical principles, PINN is a promising tool for improving thermal management strategies in various engineering applications.

CRediT authorship contribution statement

Xiangzhi Zhang: Writing – original draft, Methodology, Investigation, Formal analysis. **Chaofan Tu:** Validation, Data curation. **Yuying Yan:** Writing – review & editing, Supervision, Project administration, Conceptualization.

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

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgements

This work was supported by H2020-MSCA-RISE-778104–ThermaSMART and Royal Society grant under reference No. IEC\NSFC \211210.

References

[1] A. Arshad, M.I. Alabdullatif, M. Jabbal, Y. Yan, Towards the thermal management of electronic devices: a parametric investigation of finned heat sink filled with PCM, Int. Commun. Heat Mass Transf. 129 (2021) 105643.

[2] G. Ghaisas, S. Krishnan, A critical review and perspective on thermal Management of Power Electronics Modules for inverters and converters, Trans. Indian Natl. Acad. Eng. 7 (2022) 47–60, https://doi.org/10.1007/s41403-021-00268-1.

[3] A. Arshad, M. Jabbal, H. Faraji, P. Talebizadehsardari, M.A. Bashir, Y. Yan, Thermal performance of a phase change material-based heat sink in presence of nanoparticles and metal-foam to enhance cooling performance of electronics, J. Energy Storage 48 (2022) 103882.

[4] F. Pourfattah, M. Sabzpooshani, Thermal management of a power electronic module employing a novel multi-micro nozzle liquid-based cooling system: a numerical study, Int. J. Heat Mass Transf. 147 (2020) 118928, https://doi.org/10.1016/j.ijheatmasstransfer.2019.118928.

[5] Y. Pan, H. Chen, Y. Shen, W. Cheng, Experimental study on the flow and heat transfer characteristics of pin-fin manifold microchannel heat sink, J. Phys. Conf. Ser. 2683 (2024), https://doi.org/10.1088/1742-6596/2683/1/012031.

[6] Y. Li, Q. Wang, M. Li, X. Ma, X. Xiao, Y. Ji, Investigation of flow and heat transfer performance of double-layer pin-fin manifold microchannel heat sinks, Water (Switzerland) 14 (2022), https://doi.org/10.3390/w14193140.

[7] D. Feng, J. Nan, Y. Feng, X. Zhang, Y. Yan, Numerical investigation on improving the heat storage and transfer performance of ceramic/D-mannitol composite phase change materials by bionic graded pores and nanoparticle additives, Int. J. Heat Mass Transf. 179 (2021) 121748.

[8] A. Arshad, M. Jabbal, P.T. Sardari, M.A. Bashir, H. Faraji, Y. Yan, Transient simulation of finned heat sinks embedded with PCM for electronics cooling, Therm. Sci. Eng. Prog. 18 (2020) 100520, https://doi.org/10.1016/j.tsep.2020.100520.

[9] L. Oxarango, P. Schmitz, M. Quintard, Laminar flow in channels with wall suction or injection: a new model to study multi-channel filtration systems, Chem. Eng. Sci. 59 (2004) 1039–1051, https://doi.org/10.1016/j.ces.2003.10.027.

[10] S. Wegmann, B. Eftekharzadeh, K. Tepper, K.M. Zoltowska, R.E. Bennett, S. Dujardin, P.R. Laskowski, D. MacKenzie, T. Kamath, C. Commins, C. Vanderburg, A.D. Roe, Z. Fan, A.M. Molliex, A. Hernandez-Vega, D. Muller, A. A. Hyman, E. Mandelkow, J.P. Taylor, B.T. Hyman, Tau protein liquid–liquid phase separation can initiate tau aggregation, EMBO J. 37 (2018) 1–21, https://doi.org/10.15252/embj.201798049.

[11] G. Coulaud, Physics-Informed Neural Networks for Multiphysics Coupling: Application to Conjugate Heat Transfer, 2023.

[12] C. Wang, K. Vafai, Heat transfer enhancement for 3D chip thermal simulation and prediction, Appl. Therm. Eng. 236 (2024) 121499, https://doi.org/10.1016/j.applthermaleng.2023.121499.

[13] C.G. Fraces, A. Papaioannou, H. Tchelepi, Physics Informed Deep Learning for Transport in Porous Media, Buckley Leverett Problem, 2020. http://arxiv.org/abs/2001.05172.

[14] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks for heat transfer problems, J. Heat Transf. 143 (2021) 1–15, https://doi.org/10.1115/1.4050542.

[15] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (2021) 422–440, https://doi.org/10.1038/s42254-021-00314-5.

[16] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: a review, Acta Mech. Sin. Xuebao 37 (2021) 1727–1738, https://doi.org/10.1007/s10409-021-01148-1.

[17] S. Wang, P. Perdikaris, Deep learning of free boundary and Stefan problems, J. Comput. Phys. 428 (2021) 1–27, https://doi.org/10.1016/j.jcp.2020.109914.

[18] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics–informed neural networks: where we are and what's next, J. Sci. Comput. 92 (2022) 1–62, https://doi.org/10.1007/s10915-022-01939-z.

[19] V. Sitzmann, J.N.P. Martel, A.W. Bergman, D.B. Lindell, G. Wetzstein, Implicit neural representations with periodic activation functions, Adv. Neural Inf. Proces. Syst. 33 (2020). ISBN: 9781713829546.

[20] X. Huang, H. Liu, B. Shi, Z. Wang, K. Yang, Y. Li, M. Wang, H. Chu, J. Zhou, F. Yu, B. Hua, B. Dong, L. Chen, A universal PINNs method for solving partial differential equations with a point source, IJCAI Int. Jt. Conf, Artif. Intell. (2022) 3839–3846, https://doi.org/10.24963/ijcai.2022/533.

[21] M. Pezzoli, F. Antonacci, A. Sarti, Implicit Neural Representation With Physics-Informed Neural Networks For The Reconstruction Of The Early Part Of Room Impulse Responses, 2024, pp. 2177–2184, https://doi.org/10.61782/fa.2023.1182.

[22] S. Saitta, M. Carioni, S. Mukherjee, C.B. Schönlieb, A. Redaelli, Implicit neural representations for unsupervised super-resolution and denoising of 4D flow MRI, Comput. Methods Prog. Biomed. 246 (2024), https://doi.org/10.1016/j.cmpb.2024.108057.

[23] F.A. Torres, M.M. Negri, M. Nagy-Huber, M. Samarin, V. Roth, Mesh-free Eulerian Physics-Informed Neural Networks, 2022, pp. 1–18. http://arxiv.org/abs/2206.01545.

[24] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc, 2015, pp. 1–15.

[25] J. Soibam, I. Aslanidou, K. Kyprianidis, R.B. Fdhila, Inverse flow prediction using ensemble PINNs and uncertainty quantification, Int. J. Heat Mass Transf. 226 (2024) 125480, https://doi.org/10.1016/j.ijheatmasstransfer.2024.125480.

[26] A.B. Buhendwa, S. Adami, N.A. Adams, Inferring incompressible two-phase flow fields from the interface motion using physics-informed neural networks, Mach. Learn. Appl. 4 (2021) 100029, https://doi.org/10.1016/j.mlwa.2021.100029.

[27] X. Jiang, X. Wang, Z. Wen, E. Li, H. Wang, Practical uncertainty quantification for space-dependent inverse heat conduction problem via ensemble physics-informed neural networks, Int. Commun. Heat Mass Transf. 147 (2023) 106940, https://doi.org/10.1016/j.icheatmasstransfer.2023.106940.

[28] S. Lu, K. Vafai, Thermal performance optimization of the three-dimensional integrated circuits employing the integrated Chip-size double-layer or multi-layer microchannels, ASME J. Heat Mass Transf. 145 (2023), https://doi.org/10.1115/1.4055245.