



Inverse flow prediction using ensemble PINNs and uncertainty quantification

Jerol Soibam ^{a,*}, Ioanna Aslanidou ^a, Konstantinos Kyriyanidis ^a, Rebei Bel Fdhila ^{a,b}

^a Mälardalen University, Västerås, Sweden

^b Hitachi Energy Research, Västerås, Sweden



ARTICLE INFO

Keywords:

Heat transfer
Mixed convection
Physics informed neural network
Optimal sensor placement
Transient simulation
Inverse method

ABSTRACT

The thermal boundary conditions in a numerical simulation for heat transfer are often imprecise. This leads to poorly defined boundary conditions for the energy equation. The lack of accurate thermal boundary conditions in real-world applications makes it impossible to effectively solve the problem, regardless of the advancement of conventional numerical methods.

This study utilises a physics-informed neural network to tackle ill-posed problems for unknown thermal boundaries with limited sensor data. The network approximates velocity and temperature fields while adhering to the Navier-Stokes and energy equations, revealing unknown thermal boundaries and reconstructing the flow field around a square cylinder. Optimal sensor placement, determined by the QR pivoting technique, enhances the capture of dynamics, improving model accuracy. An ensemble PINN approach is implemented to increase robustness and generalisability, mitigating overfitting and underfitting risks, and providing a measure of model confidence. This enables the identification of reliable prediction regions and highlights potential inaccuracies, broadening applicability in complex heat transfer problems with unknown boundary conditions. Key findings include the ensemble physics-informed neural networks' superior predictive accuracy over single models and its ability to quantify uncertainty, offering insights into model validity. The study further highlights the importance of sensor placement, boundary condition enforcement, and activation function choice, with a notable shift from tanh to sin functions improving vortex shedding depiction.

1. Introduction

Electronic devices typically comprise an outer casing and internal components that generate heat during operation. With advancements in technology and current trends, these components are continually being miniaturised, allowing designers to accommodate more equipment within smaller casings. This increased component density generates more internal heat that must be dissipated into the environment. If this heat is not effectively removed, it can lead to overheating and eventually system failure. However, accurately predicting the temperature of these components is a challenging task.

One approach to estimating thermal distribution within the casing and components is through Computational Fluid Dynamics (CFD) [1,2]. CFD simulations involve solving nonlinear partial differential equations (PDEs), such as the Navier-Stokes equations, for a specific geometry with initial and boundary conditions [3]. These simulations are

commonly used in addressing industrial issues involving complex geometries and known thermal boundary conditions (BC), such as mixed Robin BC or arbitrary temperature distributions, among others [4]. In real-world heat transfer applications, like those found in power electronics, determining the precise thermal boundary conditions (BC) for a given problem is a challenge, unlike the case for velocity BC. Moreover, these thermal BCs significantly affect the accuracy of the simulation. Identifying the exact BC requires comprehensive instrumentation, which is often not practical for industrial applications [5].

Recent developments in deep learning have markedly increased interest in the domain of computational modelling and simulation of physical systems [6]. The strength of deep neural networks (DNNs) lies in their ability to learn and model intricate nonlinear functions by processing large volumes of data. When provided with ample data, they can learn to model complex nonlinear functions and predict the outcomes [7]. This process, driven purely by data, depends on the networks learn-

* Corresponding author.

E-mail address: jerol.soibam@mdu.se (J. Soibam).

ing the underlying dynamics of the system from the available data without prior knowledge of the physical laws at play [8,9]. However, when the data is either sparse or noisy, the DNNs may struggle to generalise from the training data to new situations [10]. Addressing this limitation, current research in the field is branching out into two main directions to capitalise on the capabilities of DNNs in physical modelling. The first direction involves creating surrogate models or reduced-order models by using a vast amount of experimental or computational data [11,12]. Although this method can create an accurate generative model, it often fails to meet physical constraints due to its reliance on the data alone. Additionally, the underlying mechanism of these models remains somewhat opaque, often referred to as a ‘black box’. While the DNN models demonstrate a remarkable ability to accurately map high-dimensional inputs to outputs, they require a significant volume of data points. The accumulation of such data can be computationally demanding and necessitates careful design of experiments [13].

On the other hand, the second approach emphasises enhancing DNNs with recognised physical principles, leading to the emergence of physics-constrained networks [14–16]. The distinguishing factor of these networks lies in their ability to formulate solutions for partial differential equations, which govern the operations of physical systems. They can provide accurate solutions, even when a scarcity of or no labelled datasets are available [17]. The integration of physical laws into DNNs presents a unique blend of data-driven learning and physics-based intuition. This helps to regularise the model, enhancing its reliability and strengthening its predictive performance, particularly when data are scarce [18]. By imposing physical constraints on the trainable parameters, this approach mitigates the risk of overfitting, reduces dependency on large training datasets, and consequently improves the robustness of the model for accurate predictions. It is worth noting that the idea of using neural networks to resolve PDEs is not new and can be traced back to the last century. Initial studies, such as those conducted by Lagaris et al. [19], Lee and Kang [20], and Psichogios and Ungar [21], leveraged the function approximation potentials of feed-forward, fully-connected neural networks to address initial/boundary value issues.

In more recent studies, Rassi and his team [14,22,23] introduced a deep learning framework called Physics-Informed Neural Networks (PINN) designed to solve both forward and inverse problems. They demonstrated that PINNs could be effectively applied to a wide variety of 1D PDEs, like the viscous Burger’s equation, as well as to 2D/3D PDE-constrained inverse problems. They achieved this using a moderate volume of labelled data, such as velocity field measurements. Sun et al. [24] illustrated the application of PINNs without data in fluid flow cases such as circular pipe, stenotic, and aneurysmal flows. This was achieved by rigidly enforcing the initial and boundary conditions. Rao and colleagues [25] assessed the precision of PINN in modelling incompressible laminar flow around a cylinder and introduced a mixed-variable PINN approach. They improved accuracy by predicting the Cauchy stress tensor independently, bypassing the need for automatic differentiation to compute second-order derivatives. Ryno [26] conducted a comparative study between single-PINN and segregated-PINN architectures, applying them to a dry air humidification problem within a 2D rectangular domain. The research found that the segregated-PINN architecture significantly outperformed the single-network model, achieving a 62% reduction in losses and demonstrating enhanced performance in multi-physics problems.

In the context of heat transfer, Cai et al. [27] used PINN to infer velocity and pressure fields from temperature data collected via a background-oriented Schlieren experimental setup. Their method could predict velocity and pressure fields concurrently without needing initial and boundary conditions, even with sparse and limited experiments. Nvidia [28] developed Modulus (formerly SimNet), a framework rooted in PINNs designed to solve PDEs, including multi-physics and heat transfer problems. Xi et al. [29] employed a physics-constrained network to reconstruct temperature and flow fields in natural convection

with complex geometries. This was achieved by training the network with snapshot data from CFD simulations and predicting the flow field at a separate time step. Soibam et al. [30] used an inverse PINN method to identify unknown thermal boundaries and reconstruct the full flow field with few thermal sensor data. They also demonstrated the use of transfer learning in PINN for different shapes, sizes and numbers of sources in an enclosure. Lucor et al. [31] relied on direct numerical simulation data to train a network using PINN for surrogate modelling of turbulent natural convection flows. Jalili et al. [32] demonstrated the effectiveness of PINNs in simulating two-phase fluid dynamics with heat transfer, achieving a maximum error of only 5.2% at phase interfaces and 2.8% in tracking the centre of mass for rising gas bubbles. Their method accurately modelled complex flows, achieving a maximum mean-squared error of just 0.28 for predicting the properties of unseen fluids. More recently, Cai et al. [33] used PINN for inverse heat transfer problems and two-phase Stefan problems with a moving interface, providing insight into the optimal placement of thermal sensors in a domain to enhance the predictive accuracy of the PINN model. In their study, they emphasised the crucial role that sensor placement plays, hence the sensor points were selected based on the residuals of the energy equation.

Building on previous research, there has been a growing interest in using PINN for uncertainty quantification in various applications. Uncertainty quantification becomes an invaluable asset in complex problems as it quantifies the confidence in predictions and helps navigate the inherent unpredictability, leading to more reliable and robust solutions. Jiang et al. [34] introduce an operator learning-based generative adversarial network (OL-GAN) within the Bayesian inference framework to manage the complexities of ill-posed inverse problems. The method’s effectiveness was validated through various numerical experiments, showcasing its ability to offer resolution-independent predictions and handle misaligned data. Approaches like physics-informed generative adversarial networks and polynomial chaos expansions combined with dropout, used for uncertainty quantification, have demonstrated significant success [35,36]. In their work, Yang et al. [37] introduced a Bayesian Physics-Informed Neural Network (Bayesian-PINN) by integrating a Bayesian neural network with PINN, thereby enhancing the capacity for uncertainty quantification in model predictions. However, they highlighted that determining the most suitable network structure and the optimal prior distribution still presents significant challenges in the field. Expanding upon the fundamental principles of Bayesian-PINN, Perez et al. [38] put forth a novel approach for the automatic adjustment of weights within the model. This technique autonomously refines these weights, enhancing the stability and convergence of Bayesian-PINN training. Jiang et al. [39] demonstrate the use of ensemble PINNs to quantify the epistemic uncertainty in inverse heat conduction problems, where the goal was to estimate unknown quantities of interest from noisy temperature observations. Yang et al. [40] presented a framework for quantifying and propagating uncertainty in systems governed by non-linear differential equations using PINNs. They employ latent variable models to construct probabilistic representations for the system states and propose an adversarial inference procedure for training them on data. Zeng et al. [41] introduce an approximate Bayesian computation (ABC) method combined with non-parametric population Monte Carlo (NPMC) to accurately determine unknown boundary conditions in static and nonlinear dynamic inverse heat conduction problems. This approach enhances the convergence of approximate Bayesian computation by optimising the tolerance value, significantly reducing computational costs without requiring extra parameters. Zhu et al. [42] presented an approach to uncertainty quantification of surrogate modelling for PDEs, where the governing equations were imposed as likelihood functions in the loss term. They further showed that their model was capable of producing the prediction and associated uncertainty without any labelled data.

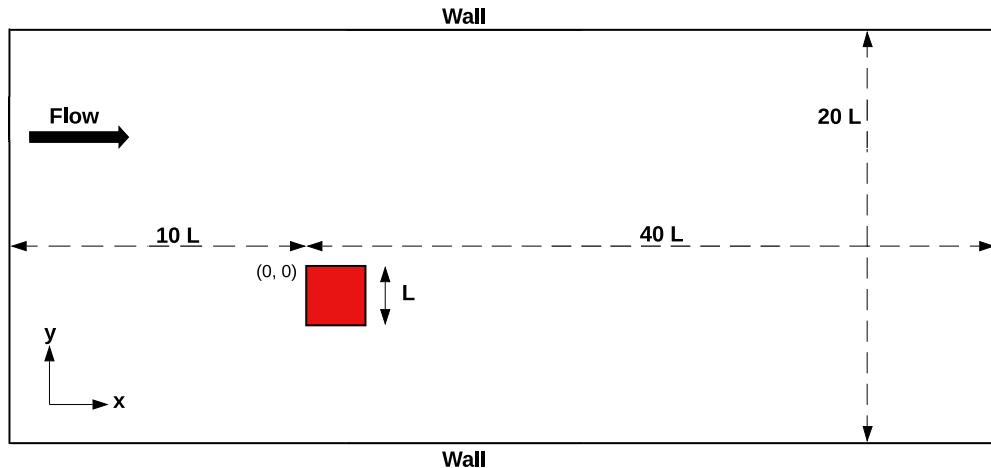


Fig. 1. 2D computational domain for numerical simulation.

Overall, advancements have been made in incorporating physical laws into deep neural networks for PDEs. However, identifying the best network structure and a prior distribution for uncertainty in model predictions is still challenging. Additionally, effectively using PINNs for heat transfer with limited or noisy data, or without comprehensive experimental setups, remains an active research area. To that extent, studying the impact of sensor placement on PINNs' predictive accuracy is critical, especially when employing PINNs as an inverse technique for tackling ill-posed problems. Such a method could unveil hidden boundaries and facilitate the reconstruction of the flow field. Hence, the current study looks into the following aspects:

- i) Implement a PINN method for ill-posed mixed convection problems where thermal boundary conditions are unknown.
- ii) Optimal sensor selection for the PINN architecture to reconstruct the flow field for transient simulation.
- iii) To construct an ensemble PINN to provide a probabilistic output which can then be used for estimating the posterior characterisation of the uncertainty associated with the predicted output.
- iv) Robustness and sensitivity analysis of the model for different sensors and flow conditions.

2. Methodology

2.1. Numerical simulation for mixed convection

The study considers a two-dimensional square cylinder, with a characteristic length L , situated in an unbounded domain, as depicted in Fig. 1. The cylinder's walls are maintained at a higher temperature (T_w) than the surrounding fluid temperature through either a constant heat flux or a constant temperature imposition. The fluid used is air, which enters the system via an inlet at a uniform velocity (U_∞) and a temperature of T_∞ . Gravity acts in the negative y -direction and perpendicular to the flow inlet, inducing thermal buoyancy. As a result of this setup, the heated square cylinder exchanges heat with the flowing fluid. The scenarios under consideration involve mixed convection, where the heat transfer is driven by both the inlet flow and the buoyancy force. The computational domain's boundaries are located at a significant distance from the heated square cylinder, with wall temperatures matching the ambient temperature. The outlet is situated on the right side boundary, which is placed sufficiently downstream from the square cylinder and assigned an outflow boundary condition. Concerning the velocity boundary condition on the walls, a no-slip boundary condition is imposed.

The study addresses an unsteady incompressible laminar flow scenario, and the thermo-physical properties are considered temperature-

independent, with the exception of the body force term in the y -momentum equation, which follows the Boussinesq approximation. The mixed convection problem is solved using the governing equations - the Navier-Stokes equations and the energy equation, which are presented in their non-dimensional form as follows:

$$\begin{aligned} \frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} &= 0 \\ \frac{\partial u^*}{\partial t^*} + u^* \frac{\partial u^*}{\partial x^*} + v^* \frac{\partial u^*}{\partial y^*} &= -\nabla p_x^* + \frac{1}{Re} \left(\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}} \right) \\ \frac{\partial v^*}{\partial t^*} + u^* \frac{\partial v^*}{\partial x^*} + v^* \frac{\partial v^*}{\partial y^*} &= -\nabla p_y^* + \frac{1}{Re} \left(\frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}} \right) + Ri\theta^* \\ \frac{\partial \theta^*}{\partial t^*} + u^* \frac{\partial \theta^*}{\partial x^*} + v^* \frac{\partial \theta^*}{\partial y^*} &= \frac{1}{Pec} \left(\frac{\partial^2 \theta^*}{\partial x^{*2}} + \frac{\partial^2 \theta^*}{\partial y^{*2}} \right) \end{aligned} \quad (1)$$

The parameters are made non-dimensional based on the free stream velocity U_∞ , free stream temperature T_∞ , and the characteristic length L , as shown below:

$$\begin{aligned} t^* &= \frac{t}{L/U_\infty}, x^* = \frac{x}{L}, y^* = \frac{y}{L}, u^* = \frac{u}{U_\infty}, v^* = \frac{v}{U_\infty}, p^* = \frac{p}{\rho U_\infty^2}, \\ \theta^* &= \frac{T_i - T_\infty}{T_{wall} - T_\infty}, \end{aligned} \quad (2)$$

where u^* , v^* , p^* , and θ^* are the non-dimensional velocity- x , velocity- y , pressure, and temperature fields respectively. ρ is the fluid density. The dimensionless quantities Re , Ri , and Pec correspond to the Reynolds number, Richardson number, and Péclet number respectively. The governing equations, along with the associated boundary conditions, are solved using Ansys Fluent and a structured mesh. To circumvent the pressure-velocity decoupling issue, the SIMPLEC algorithm is employed. Spatial discretization gradients are calculated using the Green-Gauss Cell-Based method. Temporal discretization for the transient solution is performed using the second-order backward Euler scheme. The convective terms in the momentum and energy equations are discretized using a second-order upwind scheme. The convergence absolute criteria are set to a limit of 10^{-6} for the continuity, x - and y -momentum, and energy equations. The flow is then initiated with $u^* = 1$, $v^* = 0$, and $\theta^* = 0$. The sensitivity study for time and grid was conducted, for this study Δt^* was set to 0.01. The distance of the first grid point from the square cylinder Δ^* was set to 0.0013 and it is estimated using $\Delta^* = \Delta y/L$. A cut of section of the mesh used for the simulation is presented in Fig. 2a. Several simulations were conducted for different Reynolds numbers and compared against the literature to check their convergence, stability and solution accuracy. The Strouhal number for different Reynolds numbers is plotted against the literature [43–45] as shown in Fig. 2b. The current simulation result shows a good agreement with those in the literature.

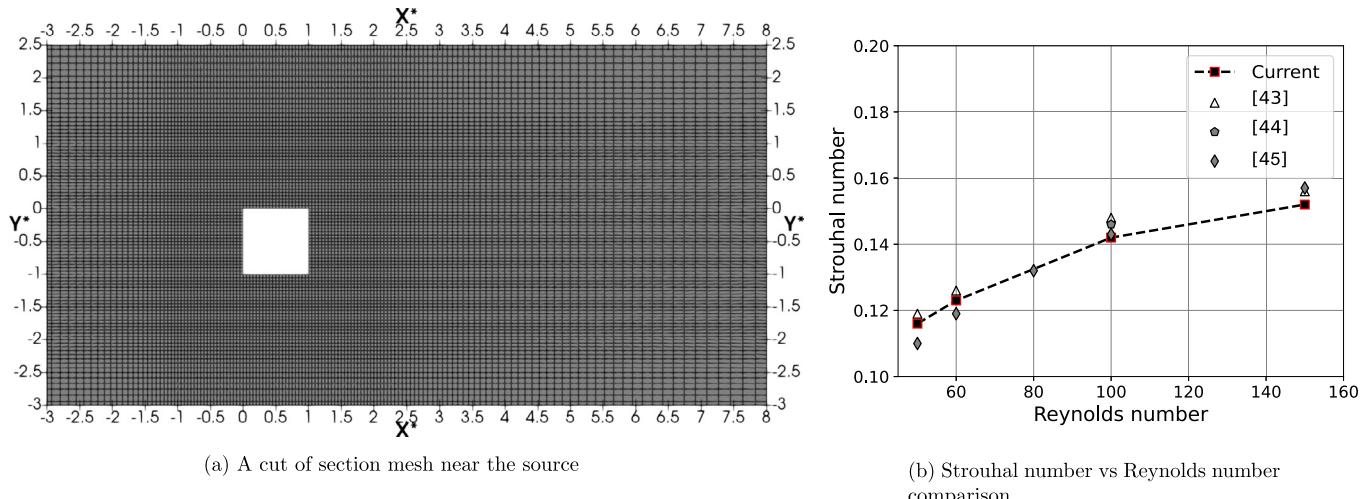


Fig. 2. Mesh grid structure of the computational domain near the source and the comparison of Strouhal numbers with the literature.

This simulation is designed to serve as a comprehensive approach towards creating a dataset for optimal sensor selection using the QR pivoting technique. Data collected from various computational scenarios are used in training the PINN methods. Moreover, a crucial aspect of this study involves the use of sensor data selected via QR pivoting to feed into the PINNs. These sensor points are used for training the PINN architecture and subsequently reconstructing the entire flow domain. Validating the accuracy of the reconstructed flow domain using PINNs is a crucial step to ensure the effectiveness of the sensor placement and the predictive power of the trained PINNs. The validation procedure is executed in two stages: Initially, by determining the L_2 error, it quantifies the difference between the sensor data from CFD and the predictions derived from PINN. Subsequently, it involves a comparative analysis of the flow fields—comparing the one generated by CFD with the reconstructed version from the PINNs—to evaluate the effectiveness of the PINNs' predictive capabilities.

2.2. Optimal QR sensor selection

In this section, the identification of the optimal sensor location for accurately estimating the flow field is discussed. The selected sensor location will then be utilised in the PINN simulation to reconstruct the flow field. It is crucial that these sparse sensor measurements can effectively capture the flow dynamics within the system. Nonetheless, locating these optimal sensor positions is a complex task [46]. To address the challenges in this study, QR pivoting is employed, which relies on QR factorisation with column pivoting [47,48]. The optimality of these sensor locations is rooted in the mathematical properties of QR factorisation. Applying QR pivoting directly to the matrix A representing the flow field will lead to increased computational demands [49]. The matrix A here is obtained from the first 20 timesteps of numerical simulation data, aimed at capturing the flow features. These features include critical flow structures such as vortices, shear layers, and potential zones of recirculation, which are essential markers for understanding how the flow evolves over time. Capturing these features is crucial not only for the accurate placement of sensors but also for ensuring that these sensors are positioned in areas where they can provide the most significant insights into the flow dynamics. To alleviate this, truncation based on energy content can be employed to focus on the most significant modes as shown in Fig. 3. This approach reduces the complexity while preserving essential information. Moreover, truncated singular value decomposition (SVD) offers robustness to noise by filtering out less significant energy modes, which are typically more susceptible to the influence of noise [50]. By utilising truncated SVD, a balance between computational efficiency and approximation qual-

ity can be achieved, making it a suitable approach for complex flow field analysis. The SVD decomposes the flow field matrix A into three matrices U , Σ , and V^* , such that:

$$A = U \Sigma V^* \quad (3)$$

Here, U and V^* are orthogonal matrices, and Σ is a diagonal matrix containing the singular values in descending order. Once the full SVD is computed it is truncated to retain only top k singular values. This results in the creation of the matrices U_k , Σ_k , and V_k^* . Here, U_k and V_k^* are composed of the first k columns from their respective full matrices, and Σ_k is a $k \times k$ diagonal matrix containing the top k singular values. Then the reduced matrix A_k is computed by multiplying the truncated matrices:

$$A_k = U_k \Sigma_k V_k^* \quad (4)$$

The reduced matrix A_k has the same dimensions as the original matrix A but represents the flow field data with fewer modes, thus reducing the computational cost. The singular values in Σ_k represent the significance of each spatial mode in the flow field. The spatial modes are ranked by their energy, with larger singular values corresponding to more dominant modes.

The goal is to find the sensor location that best captures the most dominant modes of the flow field. To achieve this, QR factorisation with column pivoting is applied to the reduced matrix A_k . This process rearranges the columns of A_k so that the diagonal elements of the upper triangular matrix R are sorted by their magnitude. As the diagonal elements of R correspond to the energy of the spatial modes, this reordering emphasises the modes that represent the most significant flow structures. Consequently, the matrix A_k is decomposed into an orthogonal matrix Q , an upper triangular matrix R , and a permutation matrix P as follows:

$$A_k P = Q R \quad (5)$$

The initial column of the permutation matrix P is associated with the optimal sensor location that captures the most dominant mode in the flow field as shown in Fig. 4. This is due to the first-row diagonal element in R having the largest magnitude, which implies it captures the highest energy mode. In the context of flow dynamics, an energy mode represents a specific pattern or structure within the flow that carries a significant portion of the flow's kinetic energy. These modes can be thought of as the dynamic signatures of the flow, each with its own characteristic energy level. The highest energy mode is typically associated with the most energetically dominant features of the flow, such as major vortices or shear layers, which play a pivotal role in the over-

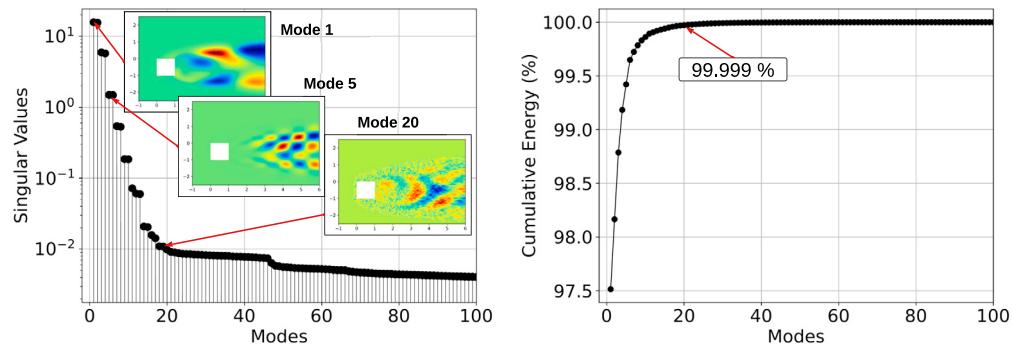


Fig. 3. Number of energy mode selection for truncated SVD and the response of singular values in the flow field.

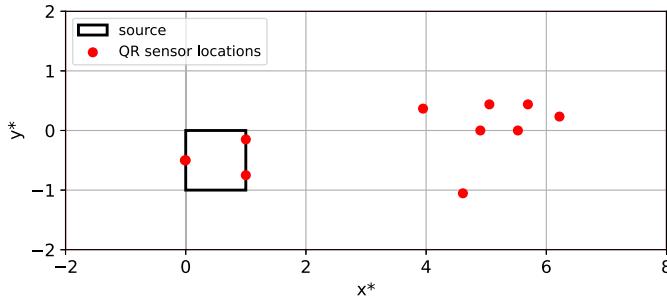


Fig. 4. Location of sensors in the domain after QR pivoting.

all behaviour and evolution of the flow field. By associating the sensor location with the row index corresponding to the highest energy mode, the most optimal sensing position is identified. This position is optimal because it captures the most significant flow structures—those that play a crucial role in the flow's dynamics. Therefore, the sensor at this location is essential for gathering the critical data needed to understand and reconstruct the flow field accurately. Similarly, the second most important sensor location can be acquired, and the process can be continued until the desired number of sensor locations is attained. The selection of the necessary number of sensors is based on user input, and it is refined based on the level of accuracy needed.

2.3. PINN for heat transfer

In this section, the proposed framework of the inverse PINN for solving the heat transfer problem is presented. The PINN framework is shown in Fig. 5 for transient mixed convection heat transfer. In particular, the PINN architecture is composed of a deep fully-connected neural network and a residual network. The first part of the architecture is trained to represent the fields of u^* , v^* , p^* , and θ^* , while the second part aims at enforcing the Navier-Stokes and energy equations at randomly selected points.

$$(x^*, y^*, t^*) \mapsto (u^*, v^*, p^*, \theta^*), \quad (6)$$

where (x^*, y^*, t^*) are the inputs of the network, which are the non-dimensional coordinates and time of the given domain. The expected outputs from the PINN are u^* , v^* , p^* , θ^* velocity in x component, velocity in y component, pressure and temperature respectively. The loss term needed for training the network is formulated below:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_{ub} + \mathcal{L}_{\theta b} + \mathcal{L}_{QR}, \quad (7)$$

where

$$\begin{aligned} \mathcal{L}_r &= \frac{1}{N_r} \sum_{j=1}^{N_r} \sum_{i=1}^{N_r} |e_j(x^i, y^i, t^i)|^2 \\ \mathcal{L}_{ub} &= \frac{1}{N_{ub} N_t} \sum_{i=1}^{N_{ub}} |u(x^i, y^i, t^i)) - u_b^i|^2 \\ \mathcal{L}_{\theta b} &= \frac{1}{N_{\theta b} N_t} \sum_{i=1}^{N_{\theta b}} |\theta(x^i, y^i, t^i)) - \theta_b^i|^2 \\ \mathcal{L}_{QR} &= \frac{1}{N_{QR} N_t} \sum_{i=1}^{N_{QR}} |\Pi(x^i, y^i, t^i)) - \Pi_{QR}^i|^2. \end{aligned} \quad (8)$$

\mathcal{L}_r is the loss term that is used to penalise the continuity, momentum, and heat equation, as shown in equation (1). The residual points $N_r = 140000$, used to minimise \mathcal{L}_r , are shown in Fig. 6a. These residual points are randomly selected based on Sobol's technique in the domain. Furthermore, the residual points were further refined near the region of the square, as shown in the figure. To compute the residuals for \mathcal{L}_r , derivatives of the output with respect to the inputs are needed. These derivatives can be obtained using automatic differentiation with TensorFlow [51], and they are the key to enabling the PINN framework. \mathcal{L}_{ub} is the loss function for the velocity boundary conditions on the walls of the source. u_b is generally known at these points, as walls are stationary and no-slip boundary conditions can be applied. In this study, velocity non-slip boundary conditions are rigorously evaluated by implementing them through both soft and hard approaches to assess their impact on the results. The third term, $\mathcal{L}_{\theta b}$, represents the loss term used for minimising the inlet temperature. $N_{\theta b}$ is the number of temperature boundary points at the inlet of the domain, which is assigned as the ambient temperature. The last term, \mathcal{L}_{QR} , is the loss term used to minimise the difference between the sensor values from the OR and the predicted values obtained from PINN. Π_{QR} are the sensor values for the velocities and temperature based on the locations obtained from QR pivoting. N_{QR} is the number of sensors, a user-defined value that can be adjusted based on the accuracy of the PINN. The N_t corresponds to the number of time steps in the PINN simulation.

The PINN model training takes into account the initial and inlet conditions from a numerical simulation. Inlet parameters, namely u^* , v^* , p^* , and θ^* , are adjusted according to the conditions specified as u_{cfd} , v_{cfd} , 0, and 0 respectively. Sensor locations, as determined by QR pivoting, are used for optimisation at each time step. It's noteworthy that these sensor locations serve both temperature and velocity (u and v) measurements. For the boundaries at the outlet, and the upper and lower walls, no specific conditions are enforced in the PINN simulation. This is done intentionally to create a smaller simulation domain compared to the original numerical simulation. Therefore, these new boundaries behave as if they were in an infinite free stream, instead of being confined by the previous wall or outlet conditions. This can be seen in Fig. 6a.

The training process for the PINN employs a fully connected neural network, configured with 15 hidden layers and 80 neurons per layer.

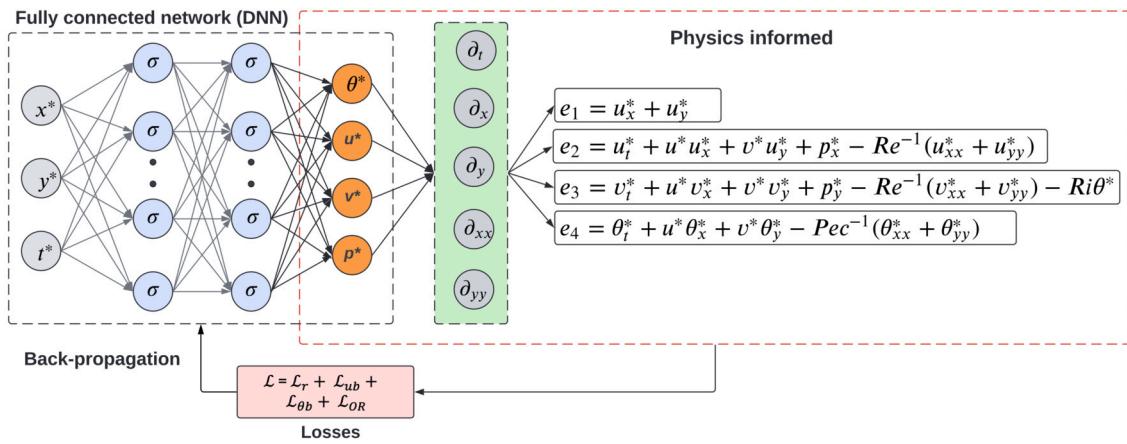


Fig. 5. PINN framework for transient mixed convection.

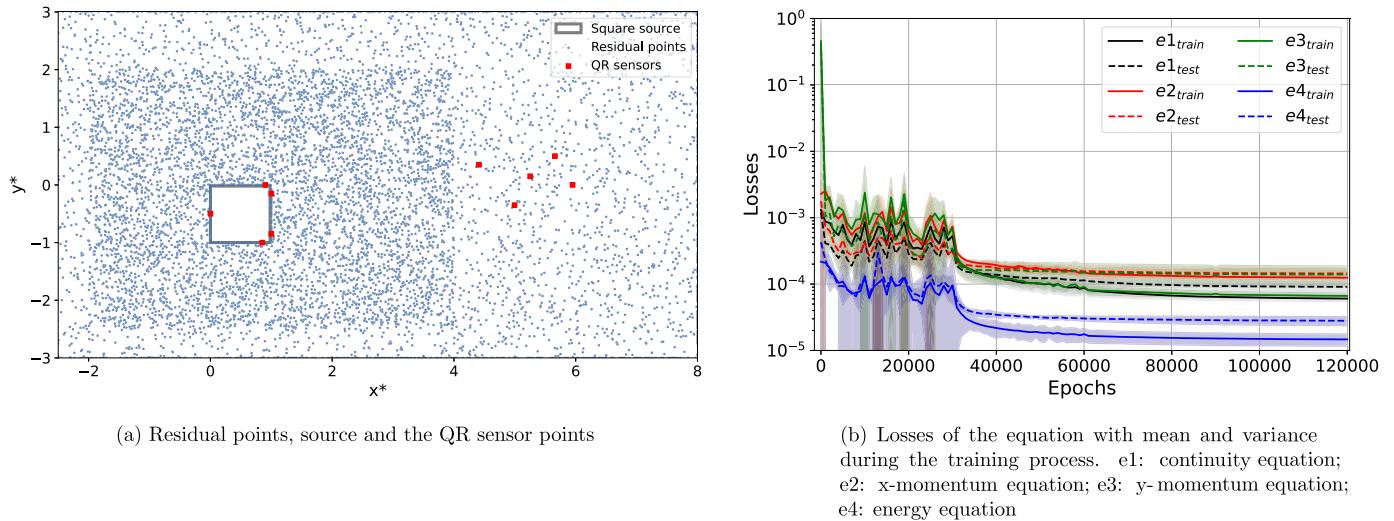


Fig. 6. PINN residuals points and the training and validation loss.

A multitude of case studies, testing various numbers of hidden layers and neurons, reveals that this setup offers optimal accuracy. Activation functions such as tanh and sin have been experimented with, the sin activation function being selected for this case study. The reasoning behind this selection is discussed in the results section. Network parameters are randomly initialised following the Glorot scheme [52], and the training is conducted using Adaptive Moment Estimation (Adam), with a learning rate that decreases from 1×10^{-3} to 1×10^{-6} . The PINN model undergoes 30k iterations of training at each learning rate, as depicted in Fig. 6b. As can be observed, the network begins to converge around the 80k epoch mark. Concurrently with the training process, the best weights and biases of the model are conserved. To prevent overfitting, early stopping is implemented every 1000 epochs.

2.4. Uncertainty quantification

The fusion of PINNs with the Deep Ensemble approach provides a potent tool for both prediction and uncertainty quantification (UQ). This combination enhances the predictive power of PINNs and delivers a reliable method for gauging epistemic uncertainties. The foundation of deep ensemble learning is based on the idea that an ensemble of multiple neural networks can significantly outperform a single network [53,54]. Deep Ensemble is an ensemble learning paradigm that utilises several neural networks, each trained independently with different random initialisations. The predictions from these networks are combined, forming a uniformly weighted mixture model. The resulting ensemble

prediction is typically approximated as a Gaussian distribution. The mean and variance of this distribution offer a measure of the prediction and its corresponding uncertainty [55]. This is mathematically depicted in the following equations:

$$\begin{aligned}\mu^*(X) &= \frac{1}{M} \sum_m \mu_m(X) \\ \sigma^{*2}(X) &= \frac{1}{M} \sum_m (\sigma_m^2(X) + \mu_m^2(X)) - \mu_*^2(X)\end{aligned}\quad (9)$$

where $\mu_m(X)$ and $\sigma_m^2(X)$ are the mean and variance of individual networks, respectively. $\mu^*(X)$ and $\sigma^{*2}(X)$ are the mean and variance of the ensemble model, respectively, and M is the total number of networks in the ensemble (5 in this case). M was carefully considered, to balance the trade-offs between computational efficiency and the performance of the ensemble model. The choice of the M is based on previous research [56] and also from the literature [55].

The effectiveness of the Deep Ensemble method originates from its comprehensive coverage of the possible behaviours within the physical processes under study, leading to enhanced prediction accuracy. This is particularly beneficial for PINNs, which deal with the intricacies and uncertainties inherent in modelling physical phenomena. Unlike single-model approaches, the ensemble technique leverages the variability among its models as a means to assess prediction uncertainty, thereby offering a quantifiable measure of confidence in its outputs. In doing so, it transforms PINNs into probabilistic models capable of systematically quantifying uncertainties, an essential feature given the

complex nature of physical problems. Furthermore, the ensemble approach provides a form of redundancy, ensuring that the collective output of multiple models increases the resilience and reliability of the PINNs, making it a superior choice for tackling complex and uncertain physical modelling tasks. Training a single model within the PINN architecture takes approximately 80 minutes, and expanding this to an ensemble model comprising 5 models increases the total training time to about 6.5 hours. Although this necessitates a significant initial training time investment, the potential for reducing computational costs through strategies like transfer learning is notable [30].

2.5. Data for training

In this study, two types of data are considered. Firstly, a constant heat flux condition is applied for the thermal boundary to carry out the numerical simulation. The data derived from this numerical simulation is employed to ascertain the QR sensor locations and subsequently used to train the PINN. The flow conditions for the constant heat flux are $Re = 100$, $Pec = 71$, and $Ri = 0.4$, and this is resolved using unsteady RANS as demonstrated through equation (1). As the case under consideration is a transient simulation, the sensor values are provided to the PINN for each time step to be simulated. For the constant temperature thermal boundary condition, data derived from direct numerical simulation (DNS) are utilised [57]. The flow conditions for this case are $Re = 50$, $Pec = 50$, and $Ri = 1.0$, with equation (1) remaining valid. In a similar manner, the QR sensor values are once again employed to train the transient PINN. The L_2 relative error is used to assess the performance:

$$\epsilon = \frac{\|\mathcal{P} - \mathcal{R}\|_2}{\|\mathcal{R}\|_2}, \quad (10)$$

where \mathcal{P} represents the predicted quantities (u^* , v^* , p^* , θ^*) obtained from the PINN model and \mathcal{R} is the corresponding reference values from URANS or DNS simulation. The PINN model is then used to reconstruct/predict the flow field in the domain and it is compared against the reference flow field obtained from URANS or DNS.

3. Results and discussions

3.1. Robustness of the model

In this section, a systematic investigation into the robustness of the PINN model is conducted, assessing its performance under a spectrum of conditions to determine its adaptability and effectiveness. The study further evaluates the strategic advantage of QR-selected sensor placements in optimising model accuracy and looks into the quantification of uncertainties associated with PINN predictions, thereby providing a comprehensive understanding.

3.1.1. Sensor sensitivity

The current study underlines the crucial aspect of sensor placement within the flow domain. This is primarily due to the thermal boundary conditions on the square source remaining unknown to the PINN simulation, presenting an ill-posed problem. For an accurate reconstruction of the flow fields in the domain using PINNs, the selected sensors need to capture dynamic changes accurately.

The QR pivoting technique has been utilised in this study to determine the optimal sensor locations. The number of sensors needed becomes a balance between the level of accuracy and the actual availability of sensors for measurements in a practical application. The effectiveness of QR pivoting has been demonstrated through a comparative case study between randomly selected and QR-determined sensors, as shown in Table 1. The relative L_2 errors reveal that a PINN with randomly selected sensors underperforms compared to a PINN with QR-determined sensors. The most substantial difference is observed in the temperature error, being the least well-known quantity to the PINN,

Table 1

Comparison of L_2 relative error for random sensors and QR sensors.

Number of sensors	15	10
Parameters	L_2 mean random	L_2 mean QR
u^*	0.079	0.022
v^*	0.106	0.045
p^*	0.121	0.032
θ^*	0.136	0.049

aside from the sensor points. Notably, the pressure field is procured by optimising the momentum equations, with no data or boundary conditions provided to the PINN model. Hence, it's projected that the L_2 error for p^* could be reduced by integrating pressure sensors into the simulation. Further comparison of PINN's performance over time, using sensors from both random and QR pivoting, is depicted in Fig. 7. These plots represent the L_2 errors for u^* and θ^* , reinforcing that a PINN with QR-determined sensors consistently outperforms a PINN with randomly selected sensors. This pattern remains constant, even as the temporal dynamics of the flow undergo continuous alterations due to vortex shedding behind the square source.

An additional point of consideration is the sensitivity analysis of the number of sensors that the PINN requires for accurate reconstruction of the temperature distribution, as demonstrated in Fig. 8a. The figure suggests that an increase in the number of sensors leads to a decrease in the L_2 error associated with the temperature distribution. The rationale behind this is that a larger number of sensors when positioned within the flow domain, can capture more detailed information about the flow's thermal characteristics. As a result, the performance of the PINN improves due to an increase in sensor locations for optimising the energy and y-momentum equations, which significantly impact the reconstruction of the temperature field. However, it's observed that the reduction in the L_2 error tends to plateau beyond a certain number of sensors. As one of the key objectives of this study is to identify both the optimal and the minimum required sensor locations for the PINN to reconstruct the flow field, a total of 10 optimal sensors were selected based on QR pivoting. This approach provides a balance between achieving the desired accuracy and limiting the number of sensors used.

3.1.2. Velocities boundary condition and activation function

The selection of boundary conditions plays a significant role in this study, particularly with respect to the adoption of the no-slip condition for velocities on the source wall. Initially, these velocities boundary conditions were softly imposed, steering the PINN model to optimise the prediction of the velocities u^* and v^* towards zero at the boundaries. This method, however, encountered difficulties in diminishing the residuals of the system's governing equations, as depicted in Fig. 8b. The inability to minimise these residuals adversely affected the predictive capacity of the PINN model, thereby limiting its effectiveness. To overcome this issue, hard enforcement of the no-slip boundary conditions was imposed on the walls. This strategy involved directly adjusting the network's output layer to comply with zero boundary values on the walls. Implementing hard boundary conditions had a notable positive effect on the performance of the PINN model, as shown in Fig. 8b. This adjustment led to an increase in the model's predictive accuracy and a significant reduction in the residuals of the equations. This change underlines the crucial role that effective enforcement of boundary conditions plays in improving the precision of PINN simulations.

The choice of activation function in PINNs is crucial for accurately simulating complex physical phenomena, as demonstrated in a study on transient heat transfer flow over a square source. Initially, a tanh activation function was employed, for its smooth, differentiable transition between -1 and 1 and symmetry around the origin. However, despite its efficiency in data normalization, tanh's asymptotic nature and lack of inherent periodicity fell short of representing the periodic pattern

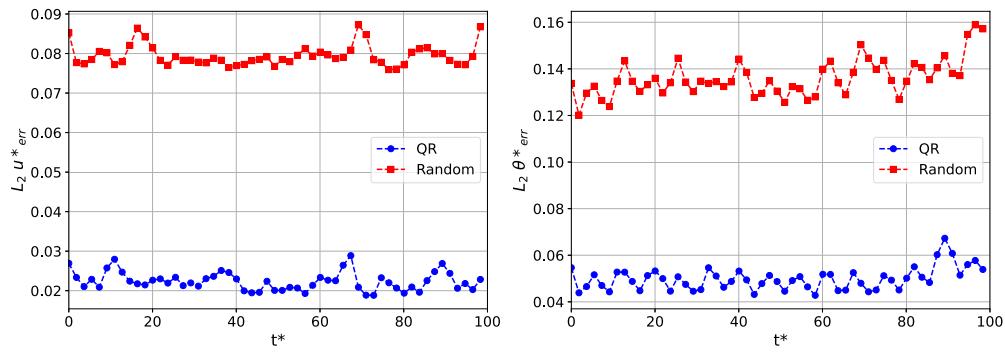


Fig. 7. Comparision of L_2 relative error between QR and randomly selected sensors over time for the component u^* and θ^* .

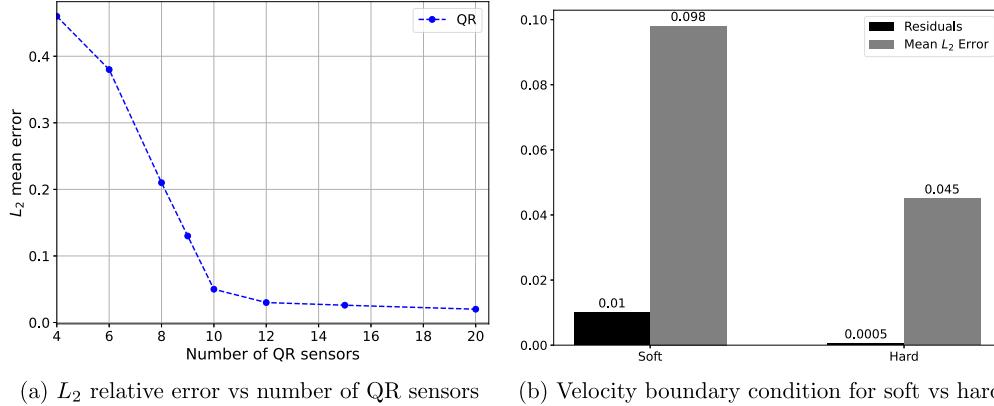


Fig. 8. Sensitivity analysis of the PINN predictive performance.

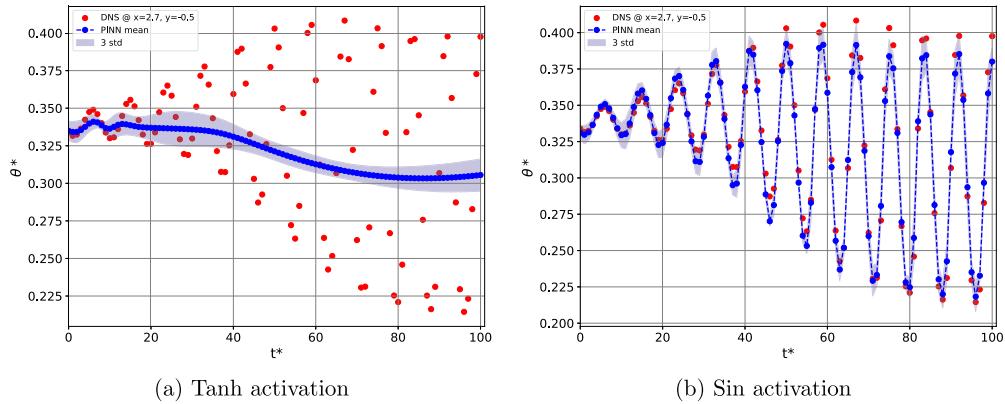


Fig. 9. Influence of activation function used in training and while predicting.

of vortex shedding. The limitations of the tanh function, depicted in Fig. 9a, were notably overcome by transitioning to a sin activation function, inherently suitable for modelling phenomena with periodic behaviour due to its natural periodicity, smoothness, and infinite differentiability. This shift, as illustrated in Fig. 9b, significantly enhanced the PINN model's ability to capture the dynamics of vortex shedding. Using the sin activation function enhances the model's precision in representing the amplitude and phase of vortex shedding which matches the physical process being modelled. This approach can be considered as providing additional knowledge to the network, enhancing its depiction of periodic phenomena which was also shown in [58]. This observation not only highlights the importance of choosing appropriate activation functions but also suggests a broader principle in PINN design: the alignment of network architecture and function choices with the underlying physics of the problem can lead to more accurate and ef-

ficient models. Thus, the sin activation function's effectiveness in this context demonstrates its utility in capturing the intrinsic properties of physical phenomena, particularly those exhibiting periodic behaviour.

3.1.3. Uncertainty quantification of PINN

This study explores two types of PINN models. The first is a single PINN model, trained on a limited set of sensor data and evaluated against the numerical dataset. The second type comprises an ensemble PINN model, consisting of five independently trained models, each with a random initialisation to the network. For both types, the number and location of sensors, determined by QR pivoting, remain identical. During the prediction phase, these independent models are merged to generate both the mean and variance of the predictions. The L_2 relative error for both models is presented in Table 2. A comparison of the L_2 relative error between the models clearly reveals the superior performance of the ensemble model over the single PINN model.

Table 2

Comparison of L_2 relative error for single PINN and ensemble PINN.

Number of sensors	10	10
Parameters	L_2 Single PINN	L_2 Ensemble PINN
u^*	0.035	0.022
v^*	0.052	0.045
p^*	0.08	0.032
θ^*	0.065	0.049

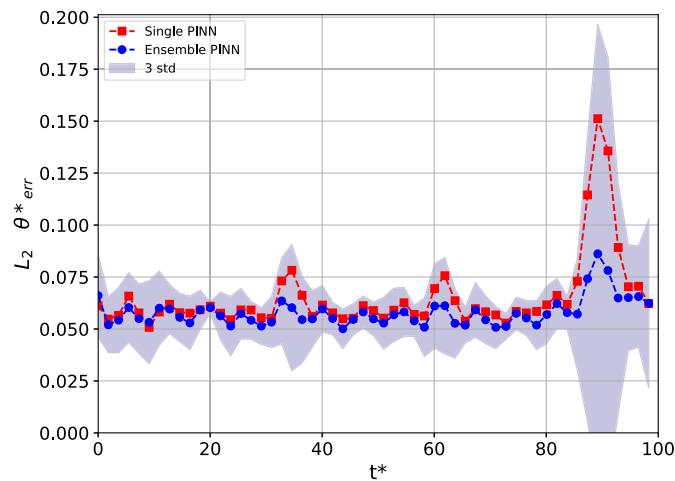


Fig. 10. Comparative analysis of the L_2 relative error over time for θ^* between single PINN and ensemble PINN Models.

It's important to acknowledge that the ensemble PINN model does indeed require greater computational resources compared to the single PINN model. Despite this, the ensemble approach presents its own unique advantages, with model generalisability being a prime benefit. By employing multiple models, the ensemble method effectively mitigates the risk of overfitting and underfitting, resulting in more robust and adaptable predictions, as illustrated in Fig. 10. Additionally, the ensemble model provides not only deterministic prediction but also estimates of the model's uncertainty. This feature proves particularly beneficial as it offers insight into the model's level of confidence. Uncertainty quantification can help identify areas where the model's predictions are trustworthy and areas where they might be less certain due to increased ambiguity. This ability to quantify uncertainty adds a level of transparency and can facilitate more efficient decision-making. Such interpretability is particularly valuable in complex physical systems where outcomes can have significant implications. In these contexts, understanding the range and likelihood of potential outcomes can be as critical as the prediction itself. For the remainder of this study, the predictions presented are based solely on the ensemble PINN method.

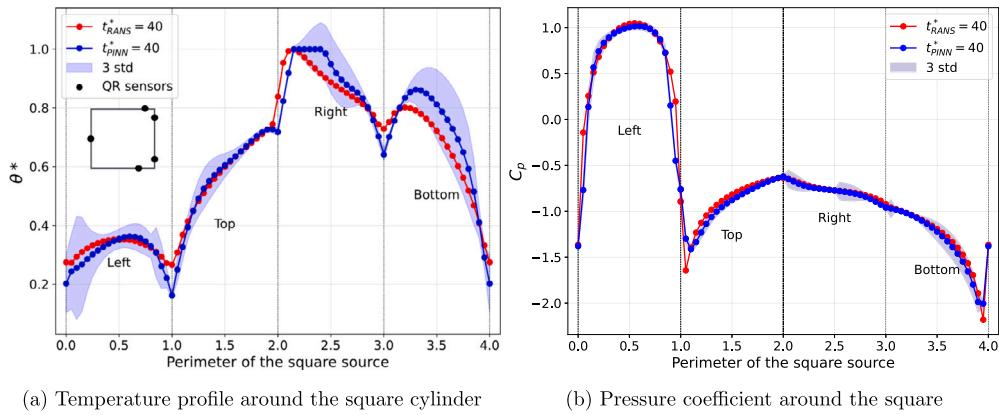
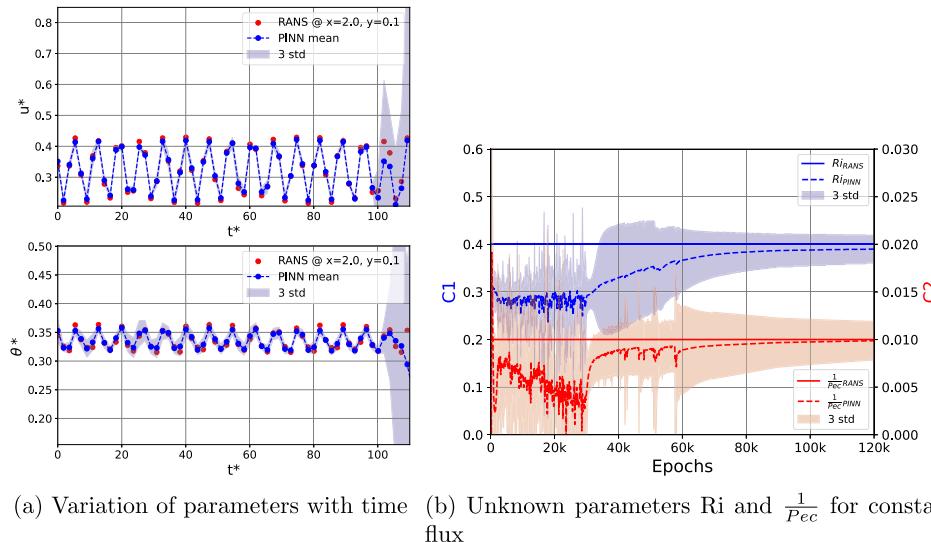
3.2. Constant heat flux

In this section, the PINN framework is applied to determine the unknown thermal boundary on the square source and the flow field, using sensor locations obtained through QR pivoting. The case under review is a constant heat flux mixed convection scenario with a Re of 100, a Pec of 100, and a Ri of 0.4. For this case, 10 sensors were employed based on the locations suggested by QR pivoting. Upon training the PINN, it was then utilised to recover the temperature profile around the square, as displayed in Fig. 11a. The plot reveals that the temperature profile delivered by the PINN follows the same trend as the values derived from URANS around the square. The plot showcased here is taken at $t^* = 40$, and possesses a maximum L_2 error of 8% on the right side where the flow has the least interaction with the square. Furthermore, it is ev-

ident that the model's uncertainty decreases in regions where sensor data is present, as anticipated. However, the uncertainty is relatively large on the left wall. A potential explanation for this might be that the PINN model lacks information about the temperature in that vicinity. Similarly, substantial uncertainty exists on the right and bottom walls, which could potentially be attributed to the effects of unsteady flow and buoyancy. Fig. 11b presents the variation of the pressure coefficient on the square wall, revealing a strong alignment between the PINN values and the URANS data. This alignment is noteworthy, particularly considering that no sensors measuring pressure values were included during the training phase of the PINN. Rather, the model minimised the residuals of the governing equations. Moreover, the relatively low level of uncertainty in this scenario suggests that the model has effectively generalised the solution by utilising the constraints provided in the form of residuals.

The predictive capability of the model is further explored by considering the downstream flow at varying time steps t^* , at a specific location $x^* = 2.0$ and $y^* = 0.1$, as displayed in Fig. 12a. The plots illustrate the evolution of u^* and θ^* , and it is evident that the values obtained from the PINN closely align with those from URANS, exhibiting very low uncertainty. A similar trend can be noted for the mean θ^* , but the uncertainty in this case is marginally higher. It is noteworthy that for $t^* > 100$, the uncertainty escalates significantly for both instances. This is attributed to the PINN model being trained only up to the 100 non-dimensional time step, indicating its inability to extrapolate the dynamics of flow over time. This issue can be mitigated by further training the model using transfer learning and extending the time step [30]. During the training of the PINN, an underlying assumption is made that the non-dimensional numbers, specifically the Ri and $\frac{1}{Pec}$, represented as C_1 and C_2 respectively in the governing equations (1), are unknown. Consequently, the PINN is tasked with estimating these values as part of its training process. This task is achieved while optimising the network based on the given QR sensor locations, which guide the network to accurately infer C_1 and C_2 using data from these sensors. The estimated values obtained via the PINN, along with the actual values from URANS, are presented in Fig. 12b. It is apparent that at the onset of training, both values are somewhat arbitrary and fluctuate significantly, exhibiting high uncertainty. However, as the model starts to converge around 100k epochs, the means of both C_1 and C_2 converge towards the actual values derived from URANS. On the other hand, despite a steady decline, the uncertainty remains relatively large. This can be attributed to the unknown thermal boundary condition in the square, as both C_1 and C_2 contribute to the heat transfer in the governing equation.

The flow fields (u^*, v^*, p^*, θ^*) reconstructed using the PINN are compared with the URANS fields in Fig. 13 at $t^* = 50$. The left side of the figure shows the predictions obtained from the PINN, the centre displays the results from the URANS simulation, and the right-hand side illustrates the L_2 relative error between the two. From the visual depiction in the figure, it is clear that the PINN model can accurately capture the dynamics of the flow fields, even encapsulating intricate patterns such as vortex shedding. This suggests that the PINN model is capable of providing not only a reliable representation of the overall flow field but also of effectively replicating complex fluid dynamics phenomena. However, it is important to note that the majority of prediction errors are primarily located around the square wall. This could be due to the complexities near the source area, which are influenced by boundary layer effects, wall friction, and flow separation, all of which are notoriously challenging to model. Enforcing the no-slip boundary condition for the velocities on the PINN model did not fully mitigate these complexities. Despite the aforementioned challenges, it is notable that the errors for u^* , v^* , and p^* remain below 3% and the error for θ^* stays under 8%. Considering that the flow fields are reconstructed using only a few sensor points, this demonstrates the robustness and reliability of the PINN model.

Fig. 11. Local parameters around the square wall for constant heat flux at $t^* = 40$.Fig. 12. Predictive performance of PINN for u^* and θ^* over time and estimation of unknown parameter using PINN.

3.3. Constant temperature

This section examines a scenario with a constant wall temperature around the source wall. The flow parameters under consideration for this particular case are $Re = 50$, $Pec = 50$, and $Ri = 1.0$. The data for this case study are derived from a DNS simulation [57]. QR pivoting was utilised to calculate optimal sensor points using the temperature distribution. The PINN subsequently utilised these sensors to reconstruct the flow field. The dynamics, as reconstructed by the PINN at the 80th time-step, are presented in Fig. 14. The left side of the figure shows the predictions obtained from the PINN, the centre displays the results from the DNS simulation, and the right-hand side illustrates the L_2 relative error between the two. The figure reveals that the error for u^* , v^* , and p^* remains around 3%, similar to the scenario of constant heat flux. However, for the temperature θ^* , the margin of error hovers around 10% near the square wall and most of the error is associated with the front side of the square.

Further investigation of the temperature contour near the square source is illustrated in Fig. 15. The figure clearly shows that the prediction provided by the PINN closely matches that of the DNS, barring some discontinuity near the walls. One possible explanation for such behaviour could be that the PINN architecture, with a limited number of sensor measurements, struggles to fully resolve the thermal boundary conditions of constant temperature. Moreover, from a practical perspective, maintaining a constant temperature on a wall is not physically feasible when both forced and natural convection are in play. The

PINN-generated prediction takes into account heat transfer due to both buoyancy and forced convection, thereby assuming that the wall should experience a cooling effect due to the presence of the flow, leading to lower predicted values on the walls.

The flow dynamics over time, based on two QR sensor locations, are depicted in Fig. 16. The first plot relates to the case where the sensor location is near the source square at $x^* = 1.1$ and $y^* = 0.3$, as shown in Fig. 16a. It is observable that the PINN predicted value tends to over-predict compared to the DNS for v^* . Nevertheless, the prediction successfully captures the trend of velocity fluctuations over time. Similarly, for the temperature profile, the PINNs prediction has a slightly higher value with a maximum L_2 error of 2% for the initial few time steps. However, as the flow advances over time, the discrepancy between the PINN and the DNS lessens. Despite the inclusion of this sensor location in the PINN training, the uncertainty indicated in both scenarios is quite high. This elevated uncertainty at this location could be attributed to the influence of the boundary layer, where the gradients are high, and the dynamics are continually changing. Regarding the sensor placement downstream of the flow at $x^* = 5.1$ and $y^* = 0.5$, a comparison of the predicted values with those of the DNS is depicted in Fig. 16b. In this case, the values align well with the DNS, with a maximum L_2 error of 0.5%. Furthermore, the uncertainty is quite low in this case, indicating the model's high confidence when predicting these values.

In a subsequent analysis, the predictive performance of the PINN model was tested at a random point selected in the wake region for

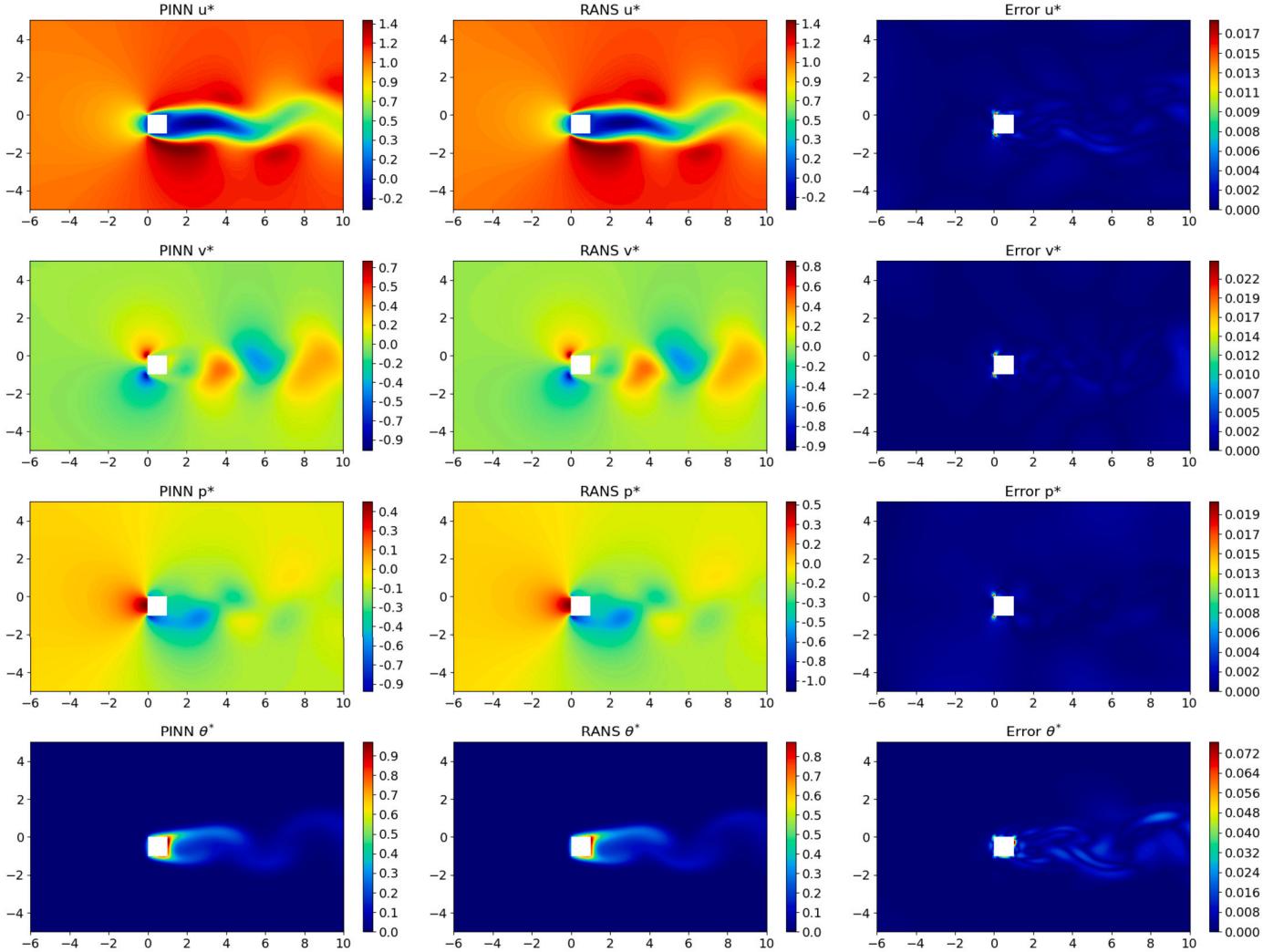


Fig. 13. Comparison of flow field profiles for constant heat flux at $t^* = 50$, as predicted by PINN and URANS, with corresponding L_2 error for u^* , v^* , p^* , θ^* .

$x^* = 2.5$ and $y^* = 0.1$. This chosen point was neither a QR sensor point nor was it part of the PINN model's training dataset. The flow dynamics at this specific point, as predicted by the PINN model, are displayed in Fig. 17. The figure reveals that all the components (u^* , v^* , p^* and θ^*) strongly align with the DNS data. This observation is consistent with previous results, further reinforcing the consistency and accuracy of the PINN model. The uncertainty levels associated with these predictions are notably low except for the pressure. As mentioned earlier no sensor values were assigned to the PINN simulation for the pressure and it is a derivative of optimising the system equations in the residual points. Hence, it is expected that the model shows higher uncertainty when predicting the pressure component. Despite the increased deviation in pressure prediction, the model's uncertainty still manages to encapsulate the range of the actual DNS values. This broad coverage provides evidence of the model's quality and accuracy, despite the fluctuating flow behaviour over time. On the other hand, for the rest of the components, the uncertainty estimated is quite low, which showcases the robustness of the model and its accuracy. Additionally, the PINN model demonstrates resistance to overfitting, indicating that it does not strictly depend on the specific data it was trained on or a specific time step. The versatility of this model was previously evidenced when it successfully reconstructed the full flow field using the QR sensor location data.

Similar to the constant heat flux case, the additional task for the PINN simulation was to estimate Ri and $\frac{1}{Pec}$. The values estimated by the PINN simulation and the actual values obtained from the DNS are displayed in Fig. 18. Here, C_1 refers to Ri and C_2 refers to $\frac{1}{Pec}$. It can be

observed that at the start of the simulation, the predicted C_1 and C_2 values exhibit significant fluctuation, despite their means being fairly close to the DNS values. This instability stems from random weight initialisation, a high learning rate, and the use of mini-batch gradient descent in the training process. However, as the training progresses, this fluctuation diminishes as the model starts to stabilise, converging towards an optimal solution around the 80k epoch mark. It can be observed from the figure that the uncertainty associated with C_1 is lower compared to that of C_2 , suggesting that C_2 might be a more complex feature to predict. Despite this, the mean values obtained from the PINN for both C_1 and C_2 closely match the DNS values. This clearly illustrates that the PINN architecture can successfully capture the dynamics of the flow, even when a limited number of QR sensors are used for the unknown thermal boundary.

3.4. Discussion and limitations

The robustness and adaptability of PINNs, as observed, largely depend on optimal sensor placement, effective enforcement of boundary conditions, and the choice of the activation function. Furthermore, ensemble models have emerged as superior to a single PINN model, offering not only more accurate predictions but also providing a measure of the uncertainty associated with those predictions. While the PINN model has achieved a remarkable level of accuracy in predicting the flow field, there are some notable limitations to consider. A primary challenge lies in the careful selection of sensor locations and the num-

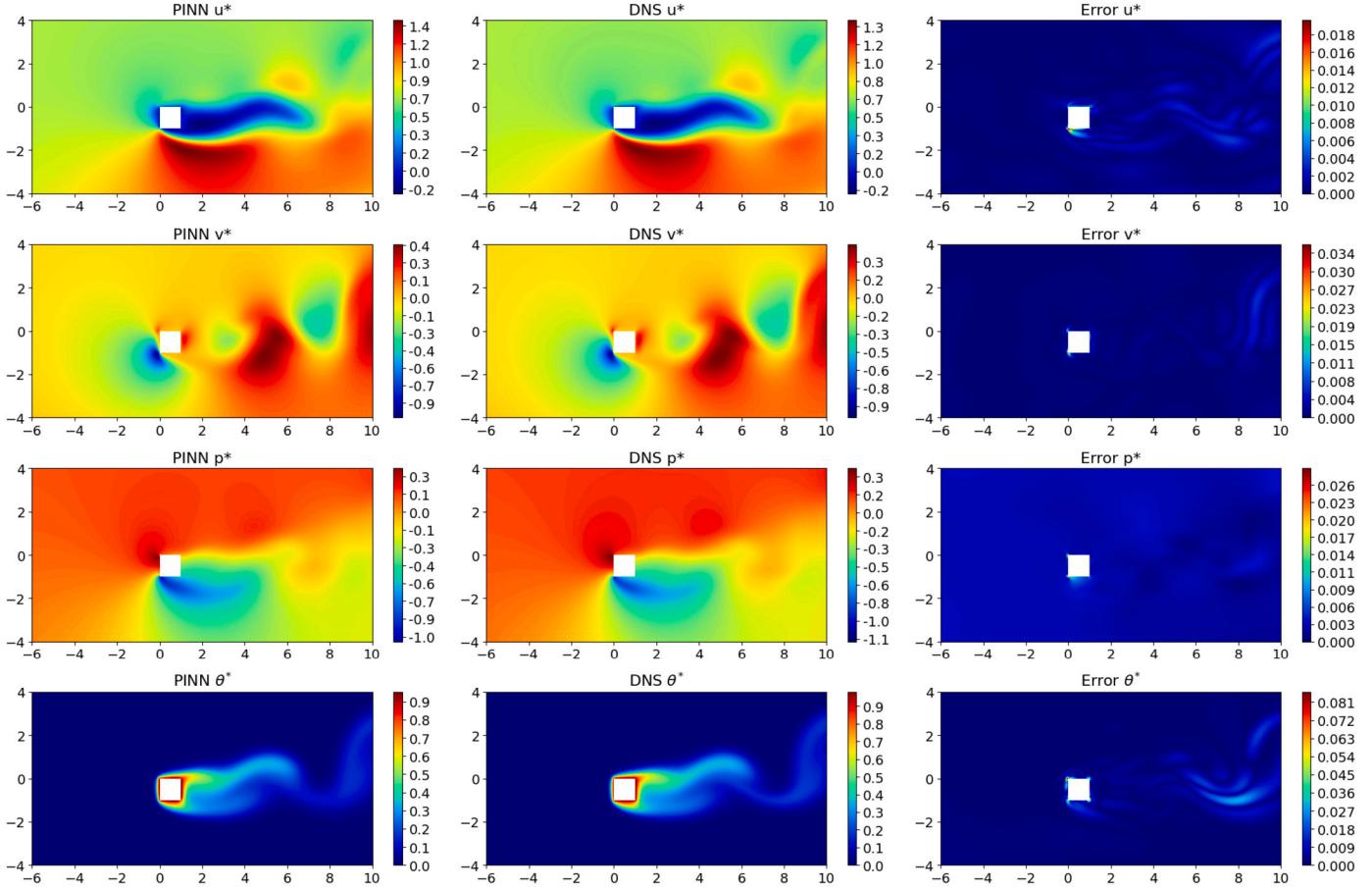


Fig. 14. Comparison of flow field profiles for constant heat flux at $t^* = 80$, as predicted by PINN and DNS, with corresponding L_2 error for u^* , v^* , p^* , θ^* .

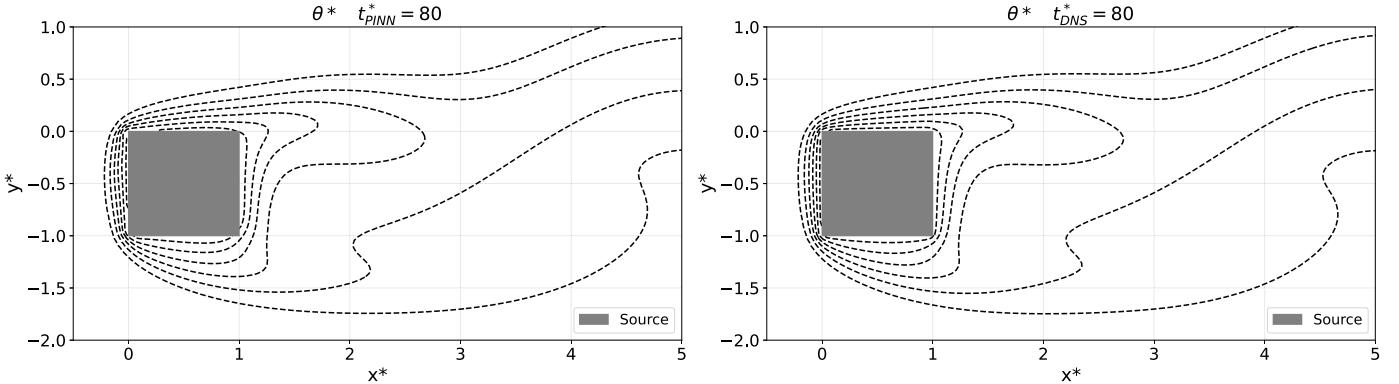


Fig. 15. Local contour profile for constant temperature at $t^* = 80$.

ber of sensors. Although the QR pivoting technique helped mitigate this issue, there is still potential for improvement in future work. For instance, integrating QR pivoting with adaptive sensor placement, informed by the residual of the energy equation [33], could dynamically adjust sensor locations to areas most critical for capturing the flow dynamics. This method ensures that data collection is both efficient and focused on regions of the highest impact on the model's accuracy. Furthermore, addressing changes in flow conditions, or variations in initial and boundary conditions, can be effectively managed through transfer learning. As demonstrated in our previous study [30], transfer learning allows for the quick adaptation of PINNs to new scenarios by applying knowledge from similar problems, thereby streamlining the adjustment process and significantly reducing the computational effort required for retraining.

The enforcement of boundary conditions, particularly the no-slip condition, also emerged as a crucial aspect affecting model performance. While hard enforcement showed superior results, it may not be the optimal solution for all scenarios. It is noteworthy to mention that the PINN model performs exceptionally well in capturing vortex shedding. Vortex shedding is a complex phenomenon, and the model's ability to accurately depict it indicates the sophistication and precision of its design. In addition, the choice of the activation function was another determinant of the model's accuracy. A shift from the tanh to the sin function improved the representation of the periodic nature of vortex shedding. However, the optimal choice of activation function may be problem-dependent and needs careful consideration.

Furthermore, it was demonstrated that the ensemble PINN not only yields better predictive accuracy but also provides the model's confi-

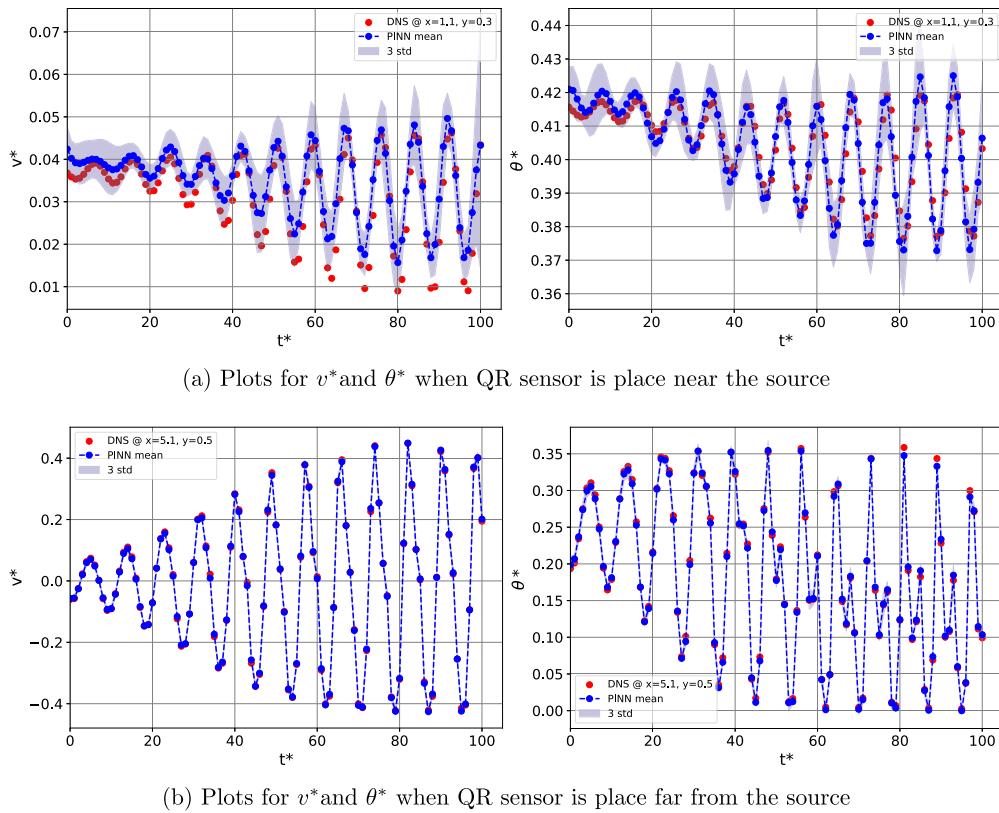


Fig. 16. Comparing the predictive performance between a QR sensor placed close to the source and a QR sensor positioned far from the source.

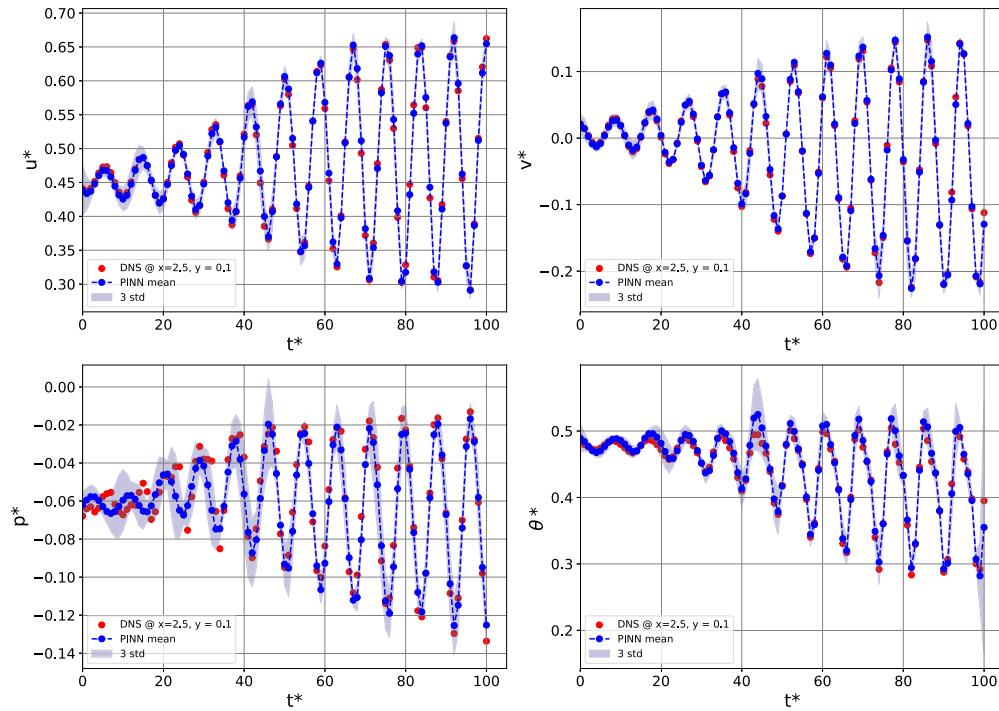


Fig. 17. Predictive performance of the PINN at random point which was not a part of QR sensor nor training.

dence level when compared to a single PINN. Quantifying the validity range of the model is a crucial aspect when dealing with complex dynamics and is as critical as the prediction itself. Even though the PINN is enforced with governing equations, a large part of the network remains as a ‘black box’. Therefore, quantifying uncertainty adds a layer

of transparency and can lead to more efficient decision-making. Despite its advantages, the current model only accounts for epistemic uncertainty, which is the model’s uncertainty. Future work should extend the model’s capabilities to also account for aleatoric uncertainty, which addresses the inherent randomness and unpredictability in the sensor

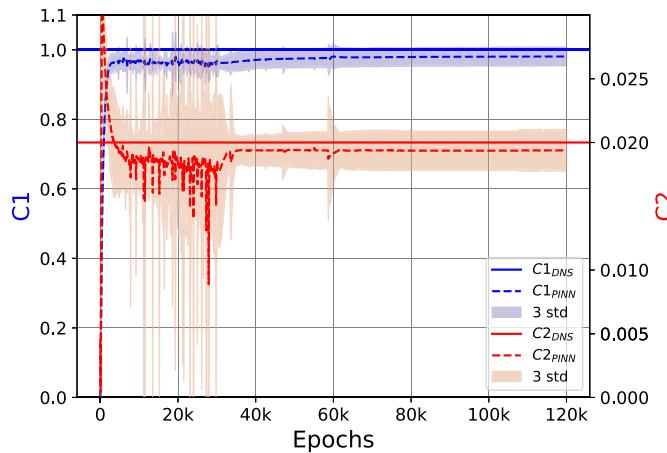


Fig. 18. Estimation of unknown parameters using PINN, where $C_1 = Ri$, and $C_2 = \frac{1}{PrRe}$.

data. This type of uncertainty captures the essential variability in the observed data, which is critical for applications subjected to fluctuating or noisy environments. By simultaneously accounting for both aleatoric and epistemic uncertainties, the model's robustness against data noise can be significantly improved, enhancing the reliability and applicability of its predictions.

4. Conclusion

The presented research highlights the effectiveness of PINNs in handling the intricate nature of mixed convection flows using limited sensor data. Given the ill-posed nature of the problem, where thermal boundary conditions are unknown, sensor data are indispensable. The QR pivoting strategy, used specifically for selecting optimal sensor placements, emerges as a crucial aspect of this process. Notably, the accuracy of the PINN model is influenced predominantly by the strategic placement of sensors rather than by their quantity. Models employing 10 QR-selected sensors showed errors below 7%, as opposed to the 16% error seen when 15 sensors are randomly positioned, thus demonstrating the significance of QR sensors in PINN simulations. This can be attributed to the fact that QR pivoting-based sensor locations coincide with the dominant energy modes in the flow field, which subsequently allows the PINN model to generalise more effectively when reconstructing the flow field.

From the comparative study of single and ensemble PINN models, the ensemble methodology clearly exhibits a significant advantage. Despite the increased computational requirement, the ensemble PINN confines the L_2 relative error below 8%, demonstrating a marked improvement over the single model's max error of 15%. An added feature of the ensemble model is its ability to quantify predictive uncertainty, introducing a crucial level of transparency, particularly for complex physical systems. From these findings, it's evident that the ensemble PINN model serves as a robust and reliable resource for exploring mixed convection flows, particularly when working with limited sensor data.

For varying thermal boundary conditions, the PINN model exhibits significant resilience and adaptability. Specifically, when dealing with constant heat flux, the model's margin of error for temperature prediction stands at approximately 7.5%. This error sees a slight increase to around 10% when subjected to a constant temperature condition. However, for the velocity components u^* , v^* , and pressure p^* , the error consistently remains below 3%, irrespective of the thermal boundary conditions. The study clearly shows that the PINN model is better equipped to adapt to constant heat flux scenarios. A plausible explanation for this could be that the model determines the temperature profile around the wall from a limited number of sensor points. Consequently, it infers that areas exposed to incoming flow (for instance, the front

wall of the square) should experience cooling due to both forced and natural convection, closely mirroring real-world applications. This assertion is further substantiated by the L_2 error for constant temperature conditions, where the maximum error occurred on the front side of the square cylinder.

CRediT authorship contribution statement

Jerol Soibam: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ioanna Aslanidou:** Writing – review & editing, Supervision, Formal analysis. **Konstantinos Kyprianidis:** Writing – review & editing, Supervision, Project administration. **Rebei Bel Fdhila:** Writing – review & editing, Supervision, Project administration, Methodology.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

The authors extend their gratitude to Yann for supplying the DNS data and offering clarifications as needed. Jerol wishes to thank Valentin, Yuanye, and Dimitra for their insightful feedback and ongoing discussions on the topic. Additionally, the authors acknowledge Eva Thorin for her significant contribution in establishing the computing system, a critical component for the successful completion of this work.

References

- [1] T.-Y. Lee, B. Chambers, M. Mahalingam, Application of CFD technology to electronic thermal management, *IEEE Trans. Compon. Packaging Manuf. Technol., Part B* 18 (1995) 511–520, <https://doi.org/10.1109/ECTC.1994.367557>.
- [2] R. Boukhanouf, A. Haddad, A CFD analysis of an electronics cooling enclosure for application in telecommunication systems, *Appl. Therm. Eng.* 30 (2010) 2426–2434, <https://doi.org/10.1016/j.applthermaleng.2010.06.012>.
- [3] T.T. Lee, M. Mahalingam, Application of a CFD tool for system-level thermal simulation, *IEEE Trans. Compon. Packaging Manuf. Technol., Part A* 17 (1994) 564–572, <https://doi.org/10.1109/95.335043>.
- [4] F. Kreith, R.M. Manglik, *Principles of heat transfer*, Cengage Learning, 2016.
- [5] O.M. Alifanov, *Inverse Heat Transfer Problems*, Springer Science & Business Media, 2012.
- [6] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* 52 (2020) 477–508, <https://doi.org/10.1146/annurev-fluid-010719-060214>.
- [7] O.A. Montesinos López, A. Montesinos López, J. Crossa, Fundamentals of artificial neural networks and deep learning, in: *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, Springer, 2022, pp. 379–425.
- [8] J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems, *Proc. Natl. Acad. Sci.* 104 (2007) 9943–9948, <https://doi.org/10.1073/pnas.0609476104>.
- [9] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, *Science* 324 (2009) 81–85, <https://doi.org/10.1126/science.1165893>.
- [10] L. Sun, J.-X. Wang, Physics-constrained Bayesian neural network for fluid flow reconstruction with sparse and noisy data, *Theor. Appl. Mech.* 10 (2020) 161–169, <https://doi.org/10.1016/j.taml.2020.01.031>.
- [11] H.F. Lui, W.R. Wolf, Construction of reduced-order models for fluid flows using deep feedforward neural networks, *J. Fluid Mech.* 872 (2019) 963–994, <https://doi.org/10.1017/jfm.2019.358>.
- [12] O. San, R. Maulik, M. Ahmed, An artificial neural network framework for reduced order modeling of transient flows, *Commun. Nonlinear Sci. Numer. Simul.* 77 (2019) 271–287, <https://doi.org/10.1016/j.cnsns.2019.04.025>.
- [13] 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>.

- [14] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707, <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [15] D. Liu, Y. Wang, Multi-fidelity physics-constrained neural network and its application in materials modeling, *J. Mech. Des.* 141 (2019), <https://doi.org/10.1115/1.4044400>.
- [16] D. Liu, Y. Wang, A dual-dimer method for training physics-constrained neural networks with minimax architecture, *Neural Netw.* 136 (2021) 112–125, <https://doi.org/10.1016/j.neunet.2020.12.028>.
- [17] C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for computational elastodynamics without labeled data, *J. Eng. Mech.* 147 (2021) 04021043, [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0001947](https://doi.org/10.1061/(ASCE)EM.1943-7889.0001947).
- [18] L. Sliwinski, G. Rigas, Mean flow reconstruction of unsteady flows using physics-informed neural networks, *Data-Cent. Eng.* 4 (2023), <https://doi.org/10.1017/dce.2022.37>.
- [19] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Netw.* 9 (1998) 987–1000, <https://doi.org/10.1109/72.721278>.
- [20] H. Lee, I.S. Kang, Neural algorithm for solving differential equations, *J. Comput. Phys.* 91 (1990) 110–131, [https://doi.org/10.1016/0021-9991\(90\)90007-N](https://doi.org/10.1016/0021-9991(90)90007-N).
- [21] D.C. Psychogios, L.H. Ungar, A hybrid neural network-first principles approach to process modeling, *AICHE J.* 38 (1992) 1499–1511, <https://doi.org/10.1002/aic.690381003>.
- [22] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations, *Science* 367 (2020) 1026–1030, <https://doi.org/10.1126/science.aaw4741>.
- [23] M. Raissi, Z. Wang, M.S. Triantafyllou, G.E. Karniadakis, Deep learning of vortex-induced vibrations, *J. Fluid Mech.* 861 (2019) 119–137, <https://doi.org/10.1017/jfm.2018.872>.
- [24] 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>.
- [25] C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for incompressible laminar flows, *Theor. Appl. Mech. Lett.* 10 (2020) 207–212, <https://doi.org/10.1016/j.taml.2020.01.039>.
- [26] R. Laubscher, Simulation of multi-species flow and heat transfer using physics-informed neural networks, *Phys. Fluids* 33 (2021) 087101, <https://doi.org/10.1063/5.0058529>.
- [27] S. Cai, Z. Wang, F. Fuest, Y.J. Jeon, C. Gray, G.E. Karniadakis, Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks, *J. Fluid Mech.* 915 (2021), <https://doi.org/10.1017/jfm.2021.135>.
- [28] O. Hennigh, S. Narasimhan, M.A. Nabian, A. Subramaniam, K. Tangsali, Z. Fang, M. Rietmann, W. Byeon, S. Choudhry, NVIDIA SimNet™: an AI-accelerated multi-physics simulation framework, in: International Conference on Computational Science, Springer, 2021, pp. 447–461.
- [29] T. Wang, Z. Huang, Z. Sun, G. Xi, Reconstruction of natural convection within an enclosure using deep neural network, *Int. J. Heat Mass Transf.* 164 (2021) 120626, <https://doi.org/10.1016/j.ijheatmasstransfer.2020.120626>.
- [30] J. Soibam, I. Aslanidou, K. Kyriyanidis, R.B. Fdhila, Inverse flow prediction using PINNs in an enclosure containing heat sources, in: March, 26–29, 2023, College Park, MD, USA, ASTFE Digital Library, Begell House Inc., 2023.
- [31] D. Lucor, A. Agrawal, A. Sergent, Simple computational strategies for more effective physics-informed neural networks modeling of turbulent natural convection, *J. Comput. Phys.* 456 (2022) 111022, <https://doi.org/10.1016/j.jcp.2022.111022>.
- [32] D. Jalili, S. Jang, M. Jadidi, G. Giustini, A. Keshmiri, Y. Mahmoudi, Physics-informed neural networks for heat transfer prediction in two-phase flows, *Int. J. Heat Mass Transf.* 221 (2024) 125089, <https://doi.org/10.1016/j.ijheatmasstransfer.2023.125089>, <https://www.sciencedirect.com/science/article/pii/S0017931023012346>.
- [33] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks for heat transfer problems, *J. Heat Transf.* 143 (2021), <https://doi.org/10.1115/1.4050542>.
- [34] X. Jiang, X. Wang, Z. Wen, H. Wang, Resolution-independent generative models based on operator learning for physics-constrained Bayesian inverse problems, *Comput. Methods Appl. Mech. Eng.* 420 (2024) 116690, <https://doi.org/10.1016/j.cma.2023.116690>, <https://www.sciencedirect.com/science/article/pii/S0045782523008137>.
- [35] D. Zhang, L. Lu, L. Guo, G.E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, *J. Comput. Phys.* 397 (2019) 108850, <https://doi.org/10.1016/j.jcp.2019.07.048>.
- [36] L. Yang, D. Zhang, G.E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations, *SIAM J. Sci. Comput.* 42 (2020) A292–A317, <https://doi.org/10.1137/18M1225409>.
- [37] L. Yang, X. Meng, G.E. Karniadakis, B-pins: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, *J. Comput. Phys.* 425 (2021) 109913, <https://doi.org/10.1016/j.jcp.2020.109913>.
- [38] S. Perez, S. Maddu, I.F. Shbalzarini, P. Poncet, Adaptive weighting of Bayesian physics informed neural networks for multitask and multiscale forward and inverse problems, *arXiv preprint, arXiv:2302.12697*, 2023.
- [39] 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>, <https://www.sciencedirect.com/science/article/pii/S0735193323003299>.
- [40] Y. Yang, P. Perdikaris, Adversarial uncertainty quantification in physics-informed neural networks, *J. Comput. Phys.* 394 (2019) 136–152, <https://doi.org/10.1016/j.jcp.2019.05.027>.
- [41] Y. Zeng, H. Wang, S. Zhang, Y. Cai, E. Li, A novel adaptive approximate Bayesian computation method for inverse heat conduction problem, *Int. J. Heat Mass Transf.* 134 (2019) 185–197, <https://doi.org/10.1016/j.ijheatmasstransfer.2019.01.002>, <https://www.sciencedirect.com/science/article/pii/S0017931018338766>.
- [42] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *J. Comput. Phys.* 394 (2019) 56–81, <https://doi.org/10.1016/j.jcp.2019.05.024>.
- [43] A. Sharma, V. Eswaran, Heat and fluid flow across a square cylinder in the two-dimensional laminar flow regime, *Numer. Heat Transf., Part A, Appl.* 45 (2004) 247–269, <https://doi.org/10.1080/10407780490278562>.
- [44] A. Sohankar, C. Norbergh, L. Davidson, Numerical simulation of unsteady low-Reynolds number flow around rectangular cylinders at incidence, *J. Wind Eng. Ind. Aerodyn.* 69 (1997) 189–201, [https://doi.org/10.1016/S0167-6105\(97\)00154-2](https://doi.org/10.1016/S0167-6105(97)00154-2).
- [45] S. Cao, Y. Ge, Y. Tamura, Shear effects on flow past a square cylinder at moderate Reynolds numbers, *J. Eng. Mech.* 138 (2012) 116–123, [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0000309](https://doi.org/10.1061/(ASCE)EM.1943-7889.0000309).
- [46] B.W. Brunton, S.L. Brunton, J.L. Proctor, J.N. Kutz, Sparse sensor placement optimization for classification, *SIAM J. Appl. Math.* 76 (2016) 2099–2122, <https://doi.org/10.1137/15M103671>.
- [47] K. Manohar, B.W. Brunton, J.N. Kutz, S.L. Brunton, Data-driven sparse sensor placement for reconstruction: demonstrating the benefits of exploiting known patterns, *IEEE Control Syst. Mag.* 38 (2018) 63–86, <https://doi.org/10.1109/MCS.2018.2810460>.
- [48] E. Clark, T. Askham, S.L. Brunton, J.N. Kutz, Greedy sensor placement with cost constraints, *IEEE Sens. J.* 19 (2018) 2642–2656, <https://doi.org/10.1109/JSEN.2018.2887044>.
- [49] S.L. Brunton, J.N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2019.
- [50] M. Gavish, D.L. Donoho, The optimal hard threshold for singular values is $4/\sqrt{3}$, *IEEE Trans. Inf. Theory* 60 (2014) 5040–5053, <https://doi.org/10.1109/TIT.2014.2323359>.
- [51] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., {TensorFlow}: a system for {Large-Scale} machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283.
- [52] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
- [53] M.A. Ganaie, M. Hu, A. Malik, M. Tanveer, P. Suganthan, Ensemble deep learning: a review, *Eng. Appl. Artif. Intell.* 115 (2022) 105151, <https://doi.org/10.1016/j.engappai.2022.105151>.
- [54] T.J. DiCiccio, B. Efron, Bootstrap confidence intervals, *Stat. Sci.* 11 (1996) 189–228, <https://doi.org/10.1214/ss/1032280214>.
- [55] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [56] J. Soibam, A. Rabhi, I. Aslanidou, K. Kyriyanidis, R. Bel Fdhila, Derivation and uncertainty quantification of a data-driven subcooled boiling model, *Energies* 13 (2020) 5987, <https://doi.org/10.3390/en13225987>.
- [57] Y. Fraigneau, 2d flow around a heated square-cylinder, $re=50$, $ra=5\times 10^6$, LISN Université Paris-Saclay. Available at <https://datasetmeca.lisn.upsaclay.fr>, 2019. (Accessed 31 March 2023).
- [58] D.E. Ozan, L. Magri, Hard-constrained neural networks for modeling nonlinear acoustics, *Phys. Rev. Fluids* 8 (2023) 103201, <https://doi.org/10.1103/PhysRevFluids.8.103201>, <https://link.aps.org/doi/10.1103/PhysRevFluids.8.103201>.