International Communications in Heat and Mass Transfer 167 (2025) 109284

Contents lists available at ScienceDirect

International Communications in Heat and Mass Transfer

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

A PINN methodology for temperature field reconstruction in the PIV measurement plane: Case of Rayleigh–Bénard convection

Marie-Christine Volk [a,b,d,*], Anne Sergent [b,c], Didier Lucor [b], Michael Mommert [a], Christian Bauer [a], Claus Wagner [a,d]

[a] Department Ground Vehicles, Institute of Aerodynamics and Flow Technology, German Aerospace Center, Göttingen, D-37073, Germany
[b] Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique (LISN), Orsay, F-91405, France
[c] Sorbonne Université, Faculté des Sciences et Ingénierie, UFR Ingénierie, Paris, F-75005, France
[d] Institute of Thermodynamics and Fluid Mechanics, Technische Universität Ilmenau, Ilmenau, D-98684, Germany

ARTICLE INFO

Keywords:
Scientific machine learning
Physics-informed neural networks
Rayleigh–Bénard convection
Temperature reconstruction
PIV

ABSTRACT

We present a method to infer temperature fields from stereo particle image velocimetry (PIV) data in turbulent Rayleigh–Bénard convection (RBC) using physics-informed neural networks (PINNs). The physical setup is a cubic RBC cell with Rayleigh number $\mathrm{Ra} = 10^7$ and Prandtl number $\mathrm{Pr} = 0.7$. With data available only in a vertical plane, the residuals of the governing partial differential equations are minimized at a set of collocation points in an enclosing 3D domain of finite thickness along the direction perpendicular to the plane. Dynamic collocation point sampling strategies are used to overcome the lack of 3D labeled information and to optimize the overall PINN convergence. In particular, in the out-of-plane direction, the collocation points are distributed according to a normal distribution, in order to emphasize the region where data is provided. Along the vertical direction, we leverage direct numerical simulation (DNS) meshing information and sample points from an optimized kernel-density estimation. This sampling approach balances labeled information by pointing greater attention to critical regions, particularly in areas with high temperature gradients within the thermal boundary layers. Using DNS planar three-component velocity data, we successfully validate the accurate reconstruction of the temperature fields in the PIV plane. We evaluate the robustness of our method with respect to characteristics of the labeled data used for training: the data time span, the sampling frequency, some noisy data, and omission of boundary data, aiming to better accommodate the challenges associated with experimental data. Developing PINNs on controlled simulation data is a crucial step towards their effective training and deployment on experimental data. The key is to systematically introduce noise, gaps, and uncertainties in simulated data to mimic real-world conditions and ensure robust generalization.

1. Introduction

Inferring the temperature in thermally-driven flows is highly relevant in fundamental studies and in various technical applications that require precise analysis and control of heat transfer, such as the design of interior ventilation systems in aircraft or train cabins, to optimize air circulation and ensure thermal comfort [1]. Especially when dealing with such complex technical applications, experiments are crucial to complement, validate or extend theory and simulation. The flow dynamics can be captured within temporally and spatially highly-resolved three-dimensional (3D) velocity measurements using tomographic particle image velocimetry (tomo-PIV) or particle tracking velocimetry (PTV) [2,3]. These methods have also been applied to the canonical experimental setup for thermally-driven flows: Rayleigh–Bénard convection (RBC) (i.e. [4]). The simultaneous measurement of spatially resolved velocity and also temperature fields is also feasible using a more elaborated set-up with temperature-sensitive tracer particles (Thermochromic Liquid Crystals, TLCs). Experiments in RBC were performed for example by Schmeling et al. [5], Schiepel et al. and [6] or Käufer and Cierpka [7]. These experimental setups, however, can be challenging, since the TLCs usually have a small color-range, are sensitive towards background light and the color-to-temperature calibration is not trivial [8]. An alternative method for obtaining temperature trajectories involves the use of neutrally buoyant, wireless temperature sensors, as demonstrated by Gasteuil et al. [9]. However, the focus of

---

* Corresponding author at: Department Ground Vehicles, Institute of Aerodynamics and Flow Technology, German Aerospace Center, Göttingen, D-37073, Germany.

E-mail address: marie-christine.volk@dlr.de (M.-C. Volk).

https://doi.org/10.1016/j.icheatmasstransfer.2025.109284

this approach is on obtaining single temperature traces rather than fully resolved spatial temperature fields. Therefore, the establishment of new methods to reconstruct temperature fields is of great interest.

A significant amount of work has been done in the past on temperature reconstruction from turbulent flows. One approach for data assimilation or reconstruction in turbulent flows uses methods from Direct Numerical Simulations (DNS) and is called nudging. It was first systematically applied in Clark Di Leoni et al. [10] to assimilate full-resolved velocity fields out of sparse velocity data. This technique is also promising for multi-field physical problems, as Bauer et al. [11] demonstrated that applying it to sparse velocity data from turbulent tomo-PIV measurements allows for temperature field reconstruction. One challenge in this approach is that errors in the measured velocity fields can significantly impact the evolution and outcome of the temperature field.

Also based on the Navier–Stokes equations, Weiss et al. [12] suggested recently a method to solve a Poisson equation for the temperature, using velocity fields of turbulent RBC. While this technique is computationally efficient, the challenge arises when applying it to experimental data, as it requires spatially and temporally fully resolved fields.

Another method employs machine learning (ML) through the integration of data and partial differential equations (PDEs) within so-called physics-informed neural networks (PINNs). PINNs are first introduced by Raissi et al. [13] and offer a powerful framework for either forward or inverse problems or assimilation and reconstruction tasks. An example for forward problems is solving the compressible Euler equations (Wassing et al. [14]). PINNs are meshfree, and therefore, compared to classic numerical solvers, highly flexible in applications with no access to fully resolved 3D fields. Several approaches assimilate velocity fields from PTV (e.g. Cai et al. [15], Wang et al. [16] and Steinfurth et al. [17]) or from different frameworks (i.e. Eulerian or Lagrangian) for efficient pressure recovery in complex moving systems (Sundar et al. [18]). Recent studies revealed the capabilities of physics-informed neural networks to predict instantaneous temperature fields in turbulent convective flows using 3D velocity fields. Toscano et al. [19] reconstruct temperature fields from PTV data using a physics-informed Kolmogorov–Arnold architecture. Mommert et al. [20] explore the influence of activation functions when reconstructing temperature fields out of fully resolved DNS velocity fields and vice versa and Lucor et al. [21] evaluate the robustness with respect to low resolution in the observation data consisting of sparse temperature data in the bulk and velocity data at the boundaries of the spatial training domain.

Although tomo-PIV and PTV are well-established and extensively validated techniques, they remain demanding in terms of equipment. Therefore, in this work, we investigate whether it is feasible to rely only on more standard two-dimensional (2D) velocity measurements for temperature reconstruction. We challenge the PINN to reconstruct temperature fields – in particular with high accuracy in the PIV plane – from stereoscopic velocity data in turbulent RBC by using spatially fully resolved velocity snapshots from DNS (cf. numerical methods in Section 2). The use of synthetic data ensures a reliable ground truth, which allows accurate validation of the results. Our PINN approach employs a padding technique, motivated in [21], by calculating the PDEs in an enclosing 3D domain around the PIV plane and by employing collocation point sampling strategies to further overcome the lack of 3D information (Section 3). In order to mimic PIV data and to quantify the challenges of not having access to spatially and temporally fully resolved velocity fields, we test the robustness of the temperature reconstruction with respect to the time span of the data and the time increment between snapshots (Section 4.2), as well as noisy data and missing labels (Section 4.4). In Section 4.3, we evaluate the accuracy of the reconstructed temperature fields along with relevant physical quantities. A key advantage of PINNs is the ability to use automatic differentiation, which provides access to all partial derivatives of the physical fields, including those in the out-of-plane direction. This allows the calculation and analysis of viscous and thermal dissipation rates (Section 4.3.2).

2. Numerical data generation

We use data generated from a DNS to mimic PIV measurements. This way, we have access to fully resolved velocity and temperature fields to evaluate the ability of PINNs to reconstruct the temperature field. A schematic of the geometry and boundary conditions of the cubic RBC cell is depicted in Fig. 1, where T_h is the temperature of the heating plate, T_c the temperature of the cooling plate, g the gravitational force in z-direction and H the height of the cell. All boundaries fulfill the no-slip condition and the sidewalls are adiabatic. The Navier–Stokes equations for an incompressible fluid are the three-dimensional momentum conservation equation together with the mass conservation equation and the energy conservation equation. In the Boussinesq approximation and in their non-dimensional form, the equations yield

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \sqrt{\mathrm{Pr}/\mathrm{Ra}}\, \nabla^2 \mathbf{u} + T \mathbf{e}_z, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \sqrt{1/(\mathrm{Pr}\mathrm{Ra})}\, \nabla^2 T, \tag{3}$$

with the three-component velocity vector $\mathbf{u} = (u, v, w)$, the temperature T, pressure p, the Prandtl and Rayleigh number $\mathrm{Pr} = \tilde{\nu}/\tilde{\kappa}$, $\mathrm{Ra} = \tilde{g}\tilde{\alpha}\Delta\tilde{T}\tilde{H}^3/\tilde{\nu}\tilde{\kappa}$ and $\mathbf{e}_z$ the unit vector pointing upward. Furthermore, $\tilde{\nu}$ is the kinematic viscosity, $\tilde{\kappa}$ the thermal diffusivity and $\tilde{\alpha}$ the thermal expansion coefficient and the tilde $(\tilde{\cdot})$ denotes the dimensional variables. The reference velocity used for non-dimensionalization is the free-fall velocity $\tilde{u}_{\mathrm{ref}} = \sqrt{\tilde{\alpha}\tilde{g}\Delta\tilde{T}\tilde{H}}$, the reference length scale is the cell height $\tilde{x}_{\mathrm{ref}} = \tilde{H}$, the reference time is accordingly $\tilde{t}_{\mathrm{ref}} = \tilde{x}_{\mathrm{ref}}/\tilde{u}_{\mathrm{ref}}$ and the reference pressure is $\tilde{p}_{\mathrm{ref}} = \tilde{\rho}\tilde{u}_{\mathrm{ref}}^2$ with the density $\tilde{\rho}$. The temperature is non-dimensionalized with $T = (\tilde{T}-\tilde{T}_0)/\Delta\tilde{T}$, where $\tilde{T}_0 = (\tilde{T}_h+\tilde{T}_c)/2$ and $\Delta\tilde{T} = \tilde{T}_h - \tilde{T}_c$. The Eqs. (1) to (3) are solved spatially with a finite volume scheme of fourth-order accuracy using a second-order Euler-leapfrog time integration scheme [22]. In the underlying study, we consider RBC of air with the non-dimensional parameters $\mathrm{Pr} = 0.7$ and $\mathrm{Ra} = 10^7$. To perform a DNS, the mesh widths have to be fine enough to fully resolve the smallest relevant scales of turbulence. Shishkina et al. [23] derived approximations for suitable grid resolutions for Rayleigh–Bénard convection based on local Kolmogorov and Batchelor length scales. Based on these estimates, we use the hyperbolic tangent function to construct a grid that progressively refines within the boundaries and transitions to an equidistant spacing in the bulk flow region (cf. Fig. 2), resulting in a total number of grid points of $N_{\{x,y,z\}} = 162$, of which 11/12 points are within the $\delta_{\mathrm{kin}}/\delta_{\mathrm{th}}$ boundary layers in z-direction. The physical fields of the DNS lie on a staggered grid, with the scalar quantities located at the cell centers and the velocity fields positioned at the cell boundaries; therefore, we interpolate the velocity fields to the cell centers using a fourth-order interpolation method. To mimic data from stereo PIV measurements, the data of the three velocity components on a vertical plane in the center of the cubic domain at $x_0 = 0.5$ is used as labeled data for training. Unless otherwise stated, we use a time span of $I_t = 5$, which corresponds to about half of a large scale circulation (LSC) turnover. We output every 500th snapshot of the DNS with time step $\Delta t_{\mathrm{DNS}} = 1.5 \times 10^{-4}$, resulting in a dataset consisting of $N_t = 68$ equidistant snapshots.

3. PINN configuration

The following section presents the general PINN configuration, including architecture, hyperparameters and optimization, and the methodology for reconstructing temperature fields from planar velocity data.

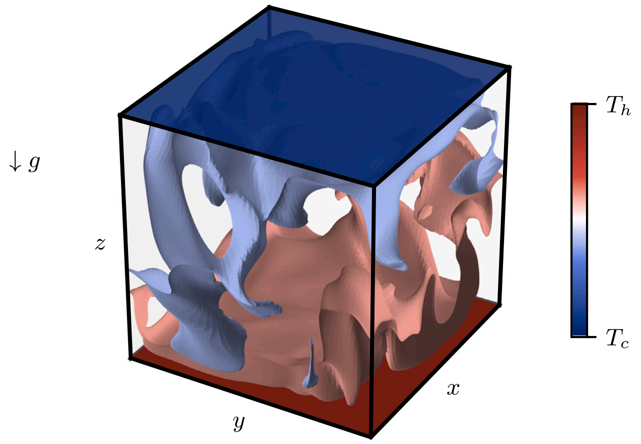

Fig. 1. Schematic of a cubic Rayleigh–Bénard convection cell with height H, temperature T_h at the heating plate and T_c at the cooling plate. All boundaries are no-slip and the sidewalls are adiabatic. Temperature isosurfaces at a representative time instant are shown for a DNS with $\mathrm{Ra} = 10^7$ and $\mathrm{Pr} = 0.7$.

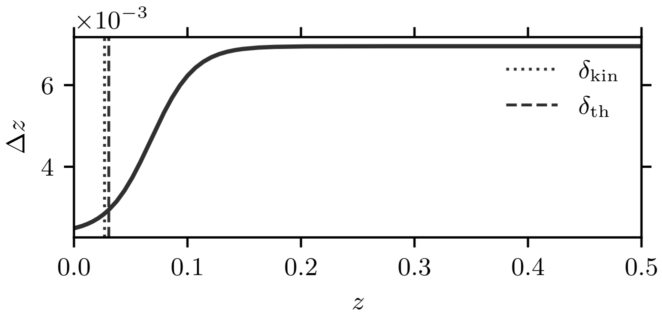

Fig. 2. The grid spacing for $\mathrm{Pr} = 0.7$ and $\mathrm{Ra} = 10^7$ is progressively refined within the thermal and kinetic boundary layers $\delta_{\mathrm{kin/th}}$ and uniform in the bulk. The cell height is $H = 1$. The required resolution is estimated from [23] to resolve the local Kolmogorov and Batchelor length scales.

3.1. General architecture and hyperparameters

The PINN (cf. Fig. 3) consists of a multilayer perceptron (MLP) with N_{L} densely connected layers, each containing N_{N} sine activated neurons, except for the output layer, which uses a linear activation.

We use periodically activated neural networks, because they have been shown to outperform monotonically activated neural networks in turbulent flow assimilation tasks, as demonstrated by an comparative study in [20]. This advantage arises from their resistance to vanishing gradients and their alignment with the inherently periodic nature of turbulent flow dynamics. The network takes as input a set of 4D spatial and temporal coordinates and outputs the five physical fields. It can therefore be seen as a function $f([x, y, z, t], \boldsymbol{\theta}) = [u, v, w, T, p]$, where $\boldsymbol{\theta}$ is the set of all trainable variables (weights $\boldsymbol{w}$ and bias $\boldsymbol{b}$ parameters). This architecture leads to the ability to obtain spatial and temporal derivatives of the fields via automatic differentiation [24]. Using sine activation functions, the initialization of the weights has a non-negligible impact on the accuracy and convergence speed of the network. As proposed by Sitzmann et al. [25], the weights are randomly initialized by a uniform distribution in the range $\boldsymbol{w} \sim \mathcal{U}\left(-\sqrt{6/N_{\mathrm{N}}^{\mathrm{L}-1}}, \sqrt{6/N_{\mathrm{N}}^{\mathrm{L}-1}}\right)$. All layers are treated equally here, since the introduction of higher frequencies in the first layer, as suggested by [25], does not significantly affect the results (cf. [20]).

3.2. Loss functions and sampling

PINNs make use of a combined loss function that is empirically evaluated on a set of points, including known data points and selected collocation points where physical information is evaluated. Our loss function is a weighted sum including data, PDEs and boundary conditions

$$\begin{aligned} \mathcal{L} =& \lambda_{\mathrm{d}}\mathcal{L}_{\mathrm{data}} + \mathcal{L}_{\mathrm{PDE}} + \mathcal{L}_{\mathrm{BC}}, \\ &\text{with } \mathcal{L}_{\mathrm{PDE}} = \lambda_{\mathrm{m}}\mathcal{L}_{\mathrm{mom}} + \lambda_{\mathrm{c}}\mathcal{L}_{\mathrm{cont}} + \lambda_{\mathrm{e}}\mathcal{L}_{\mathrm{energy}}, \\ &\text{and } \mathcal{L}_{\mathrm{BC}} = \lambda_{\mathrm{t}}\mathcal{L}_{\mathrm{BC}(T)} + \lambda_{\mathrm{v}}\mathcal{L}_{\mathrm{BC}(u,v,w)}, \end{aligned} \tag{4}$$

where $\lambda_{[\cdot]}$ are static weights. An example of visualization of the sampling of the three loss components at a given time is shown in Fig. 4. The data loss (left) is the mean squared error between the velocity data points and the predictions, using the full spatial DNS resolution on plane A

$$\mathcal{L}_{\mathrm{data}} = \frac{1}{3N_b}\sum_{i=1}^{N_b}\left\|\left(\hat{\mathbf{u}}_i - \mathbf{u}_i\right)^2\right\|_1, \tag{5}$$

where $\hat{\mathbf{u}}$ are the true values, $\mathbf{u}$ are the reconstructed values and N_b is the batch size. Since temperature and velocity values are often known in well-controlled experiments – such as no-slip velocities at the walls and imposed temperatures at the top and bottom – we account for this information in the form of an additional synthetic set of boundary points. These points are located both in-plane and off-plane ($z \in \{0, 1\} \times x \times y$) in a domain of thickness δ_x around x_0 (Fig. 4, right). The PDE loss consists of the residuals of Eqs. (1) to (3)

$$\mathcal{L}_{\mathrm{mom}} = \frac{1}{3N_b}\sum_{i=1}^{N_b}\left\|\left(\frac{\partial \mathbf{u}_i}{\partial t} + \mathbf{u}_i \cdot \nabla\mathbf{u}_i + \nabla p_i - \sqrt{\mathrm{Pr}/\mathrm{Ra}}\,\nabla^2\mathbf{u}_i - T_i\mathbf{e}_z\right)^2\right\|_1, \tag{6}$$

$$\mathcal{L}_{\mathrm{cont}} = \frac{1}{N_b}\sum_{i=1}^{N_b}\left(\nabla \cdot \mathbf{u}_i\right)^2, \tag{7}$$

$$\mathcal{L}_{\mathrm{energy}} = \frac{1}{N_b}\sum_{i=1}^{N_b}\left(\frac{\partial T_i}{\partial t} + \mathbf{u}_i \cdot \nabla T_i - \sqrt{1/(\mathrm{PrRa})}\,\nabla^2 T_i\right)^2, \tag{8}$$

and the corresponding weights are labeled λ_{m}, λ_{c} and λ_{e}. The collocation points, where the PDE loss is calculated, are generated in order to help predict the temperature field in the plane A where the labeled velocity data reside. Previous studies have demonstrated that the performance of PINNs can be improved by strategically placing collocation points in critical regions (e.g., [26,27]). Building on this, we discuss below our sampling strategies to deal with the lack of information in the x-direction.

We place the collocation points in a domain around the plane $\mathcal{D} = [x_0 - {}^{\delta_x}\!/_2, x_0 + {}^{\delta_x}\!/_2] \times [0, 1] \times [0, 1]$ with δ_x being the thickness (cf. Fig. 4 center). Given the data at x_0, the collocation points should be strategically placed to guide the network toward learning the physical solution consistent with the provided data. The collocation points in x-direction are therefore randomly distributed according to a normal distribution with $\mu = x_0$, which is adapted to characteristic physical length scales (see Fig. 5(a)). The so-called collocation layer with width δ_x is defined to be the $\pm 2\sigma$-interval of the normal distribution shown in Fig. 5(a), thus containing statistically 95.4% of the collocation points. The boundaries are the cell walls at $x = 0$ and $x = 1$. This method uses a weighted approach, prioritizing regions with available data while ensuring that all other regions contribute to the accurate reconstruction of the temperature field within A.

Since high vertical temperature gradients reside close to the bottom and top walls, where buoyancy increases due to heating, leading to rising hot plumes (bottom plate) and buoyancy decreases, causing sinking cold plumes (top plate), we optimize the positioning of the collocation points in the vertical z-direction, to correctly capture these gradients. Since PINNs generally have difficulty resolving high gradients, a higher density of collocation points is allocated in the boundary layers in

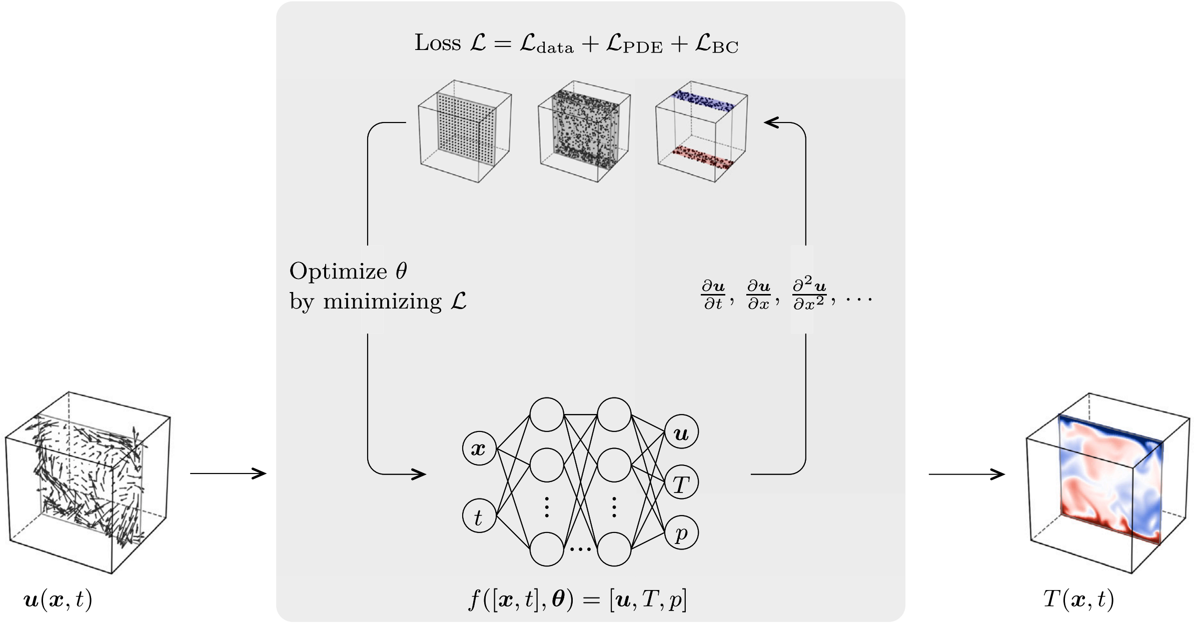

Fig. 3. Schematic of the PINN architecture. The network is a fully connected MLP that takes spatial and temporal coordinates (x, y, z, t) as input and predicts physical fields (u, v, w, T, p) as output. Partial derivatives are obtained via automatic differentiation and used to compute PDE residuals, which are minimized alongside data and boundary condition losses.

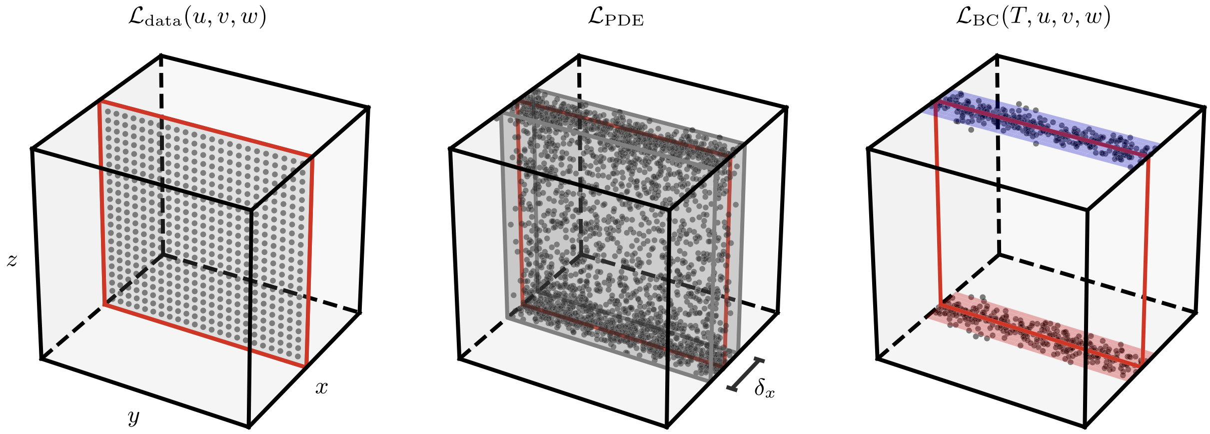

Fig. 4. Schematic of the positions of the loss points. Labeled data points for velocity are given only on the vertical plane A (left), whereas the residuals of the PDEs are minimized in a 3D collocation layer with thickness δ_x (center). Additional Dirichlet data points for velocity and temperature are placed on the top and bottom plates for the boundary condition loss term (right).

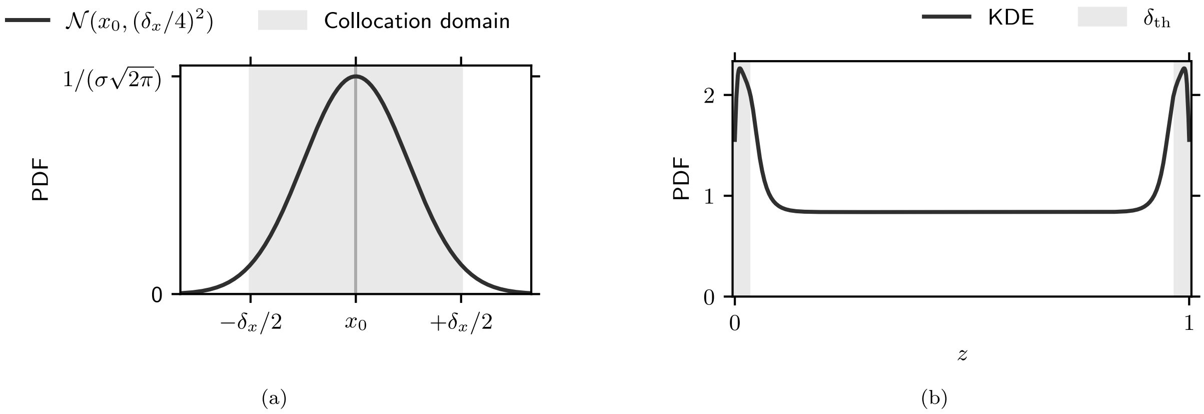

Fig. 5. (a) Collocation points in x-direction are sampled using a Normal distribution centered at $\mu = x_0$. The thickness of the collocation layer δ_x is defined to be the $\pm 2\sigma$-interval of the distribution. (b) The collocation points in the vertical direction are refined in thin layers above and below the heating and cooling plate respectively, by making use of the refinement of the DNS grid shown in Fig. 2 to resolve the boundary layers, which are characterized by high velocity and temperature gradients.

comparison to the bulk. This density is determined by a distribution following the physically motivated distribution of the non-equidistant distribution of the DNS grid points introduced in Section 2. We use kernel density estimation (KDE) [28], where the kernel function is a Gaussian distribution with zero mean and unit variance $K(z) = \mathcal{N}(z, 1)$. The kernel function is evaluated at each of the N_z grid points with a constant kernel bandwidth h and is summed up to form the KDE

$$\mathrm{KDE}(z) = \frac{1}{N_z h} \sum_{i=1}^{N_z} K\left(\frac{z - z_i}{h}\right). \tag{9}$$

The bandwidth is defined as the mean grid spacing $h = \langle \Delta z \rangle$, resulting in a smooth distribution that is higher near the top and bottom plates and uniform in the bulk (cf. Fig. 5(b)). Furthermore, the collocation points are uniformly distributed in time and y-direction.

To enhance the network's convergence, the domain is optimally filled by applying the Latin Hypercube Sampling scheme (LHS) [29] together with the aforementioned individual distributions in all four directions. The use of LHS with a custom distribution ensures that the collocation points follow the desired probability density more effectively, leading to a better representation of the important regions in the domain and thus to a more efficient training of the PINN. See Appendix C for pseudo-code of the sampling.

The described method is used to generate a new random set of collocation points at each iteration, where an iteration corresponds to processing a mini-batch of the data. This effectively covers the physical space and improves generalization by preventing the model from overfitting to a fixed set of points.

3.3. *Training optimization and loss weighting*

For an effective training, careful consideration is required when assigning loss weights. The confidence level of the loss terms is different and we want to follow the most reliable information. Thus, at this stage, since we are using DNS data for the training, systematic or random measurement errors are not taken into account; consequently, the data loss term is assigned the highest weight $\lambda_\mathrm{d} = 1$. Furthermore, since the velocity data is provided, we weight the Navier–Stokes momentum equation highest from the PDEs ($\lambda_\mathrm{m} = 1 \times 10^{-1}$) and the energy equation is weighted less by a factor of ten ($\lambda_\mathrm{e} = 1 \times 10^{-2}$). Since relaxing the constraint of the continuity equation has been shown to improve the training results [21], this term is weighted even less ($\lambda_\mathrm{c} = 1 \times 10^{-3}$). These weights have proven effective for temperature reconstruction, as reported in [20]. In this particular setup, however, where the data is given only in a 2D plane, the boundary condition term must be handled carefully to avoid overloading the boundary information which would lead to the trivial solutions of $\mathbf{u} = \mathbf{0}$ and the heat conduction state $T(z) = 0.5 - z$. Consequently, the latter term carries the smallest weight compared to all other loss terms ($\lambda_\mathrm{t} = \lambda_\mathrm{v} = 5 \times 10^{-6}$). Despite the extremely small weights, the boundary conditions are still effectively optimized, as shown by an example training history in Fig. B.22.

The Adam optimizer with its default configuration is employed for training [30]. A learning rate scheduler monitors the total loss function and gradually reduces the learning rate by a factor of 0.8 from an initial value of $\eta_\mathrm{max} = 1 \times 10^{-3}$ to a minimum of $\eta_\mathrm{min} = 1 \times 10^{-4}$. In addition, the batch size for both the data and the generation of collocation points is set to $N_\mathrm{b} = 4096$.

4. Evaluation

Unless otherwise specified, a random spatial and temporal subset of 5% of the data points is omitted from the training and used to monitor the training process and for evaluation. An example of the metrics plotted against the training epochs is provided in Appendix B. The evaluation metrics used in this study include the mean absolute error (MAE), the coefficient of determination (R^2), the relative L_2 error, and the relative pointwise error (ϵ_{L_2}), which is used for plotting

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|, \tag{10}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\hat{y}_i - \bar{\hat{y}})^2}, \tag{11}$$

$$L_2^\mathrm{rel} = \frac{\sqrt{\sum_{i=1}^{n} \left(\hat{y}_i - y_i\right)^2}}{\sqrt{\sum_{i=1}^{n} \hat{y}_i^2}}, \tag{12}$$

$$\epsilon_{L_2,i} = \frac{|\hat{y}_i - y_i|}{\sqrt{\sum_{i=1}^{n} \hat{y}_i^2}}, \tag{13}$$

where $\hat{y}$ are the ground truth values (DNS), y are the reconstructed values and $\bar{\hat{y}}$ is the mean of the ground truth values. The default dataset used for training spans a time period of $I_t = 5$, which corresponds approximately to half on an LSC turnover time, and is divided into $N_t = 68$ equidistant snapshots. The default PINN configuration consists of $N_\mathrm{L} = 10$ densely connected layers with $N_\mathrm{N} = 256$ sine activated neurons each, which is the result of a parameter study (see Appendix A). Furthermore, all cases are trained for 10^4 epochs for comparison. The final results of the PINN are presented in Section 4.3, while the following two sections focus on identifying the critical training parameters.

4.1. *Spatial training domain*

Since data and collocation points are only placed in a 2D space, the PINN does not have out-of-plane information and is therefore not capable of accurately predicting the $\partial/\partial x$ derivatives of all physical fields except $\partial u/\partial x$, which can be obtained from the continuity equation (Eq. (2)). Therefore, for the reconstruction of the in-plane temperature field, it is crucial to determine an appropriate collocation layer thickness δ_x. Fig. 6 shows a comparison of planar temperature fields $T\big|_{x_0}$ and temperature gradient fields $\partial T/\partial x\big|_{x_0}$ from the ground truth data (left) and two different PINN reconstructions with $\delta_x = 0$ (middle) and $\delta_x = 0.15$ (right), i.e. a 2D and 3D collocation domain. It can be seen that the PINN trained with $\delta_x = 0$ is not able to predict the structures of the ground truth $\partial T/\partial x$ field. Some key structures in the temperature field are still recognizable, but the overall field deviates significantly from the ground truth; whereas for $\delta_x = 0.15$ the field closely resembles the ground truth field. Fig. 7 shows R^2 and MAE for the predicted temperature field in dependence of the thickness δ_x. The training time increases for larger training domains because the spatial collocation point density remains the same for each case. Increasing δ_x leads to increasing R^2 scores and decreasing MAE values until $\delta_x \approx 0.15$. Further expanding the collocation layer does not improve the results. For the largest tested values $\delta_x \geq 0.4$, there is even a decrease in the R^2 score and increase in the MAE values. In regions far from x_0, where no data is available, the PINN generates its own physically meaningful field. However, this field may not fully correspond to the ground truth in these regions, potentially introducing an unwanted influence on the solution at x_0. Therefore, a good choice for the thickness of the training domain is the length scale where the velocity field is still statistically correlated with the field itself at x_0. Following Thacker et al. [31], we compute the horizontal integral length scales of the velocity components averaged over 73 3D snapshots equidistantly distributed over a large time span of $t = 500$. The average of all components in both horizontal directions is $\Lambda = 0.13$, which agrees well with the optimal collocation layer thickness $\delta_x = 0.15$.

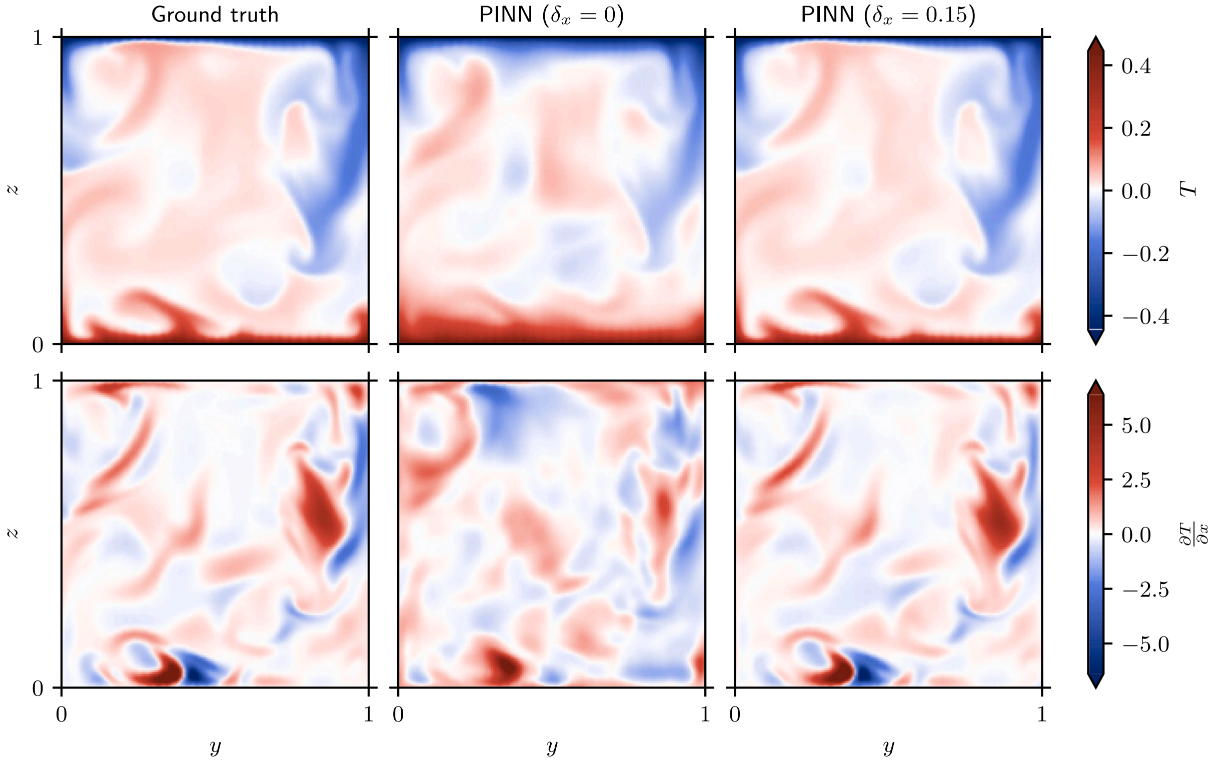

Fig. 6. Comparison of planar distribution of temperature fields $T\big|_{x_0}$ (top) and temperature gradient $\partial T/\partial x\big|_{x_0}$ fields (bottom) at a given instant; ground truth data (left), the predictions from PINN configurations with $\delta_x = 0$ (center) and $\delta_x = 0.15$ (right).

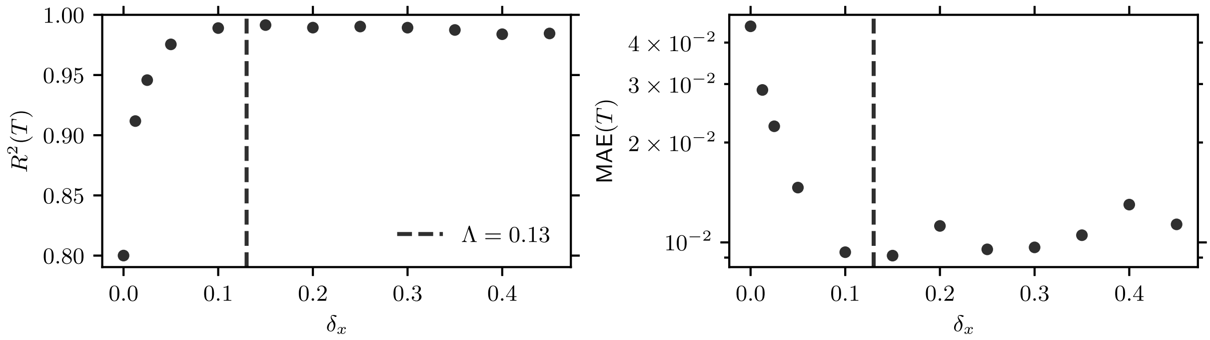

Fig. 7. Coefficient of determination R^2 (left) and MAE values (right) of the in-plane temperature reconstruction vs. the thickness δ_x of the collocation layer. The lowest-performing result for the temperature prediction is obtained using only a 2D collocation domain ($\delta_x = 0$). The results reach a plateau with increasing δ_x, as training into regions too distant from the data points provides no benefit.

4.2. *Temporal training domain*

Since, unlike other studies, we do not specify initial conditions for the training, the result of the training depends on the time span of the available data. Therefore, in this section, we analyze how the time span and the temporal resolution of the given data snapshots influence the accuracy of the reconstructed temperature field.

The PINN presumably needs to be trained over a sufficient time span to learn the dynamics of the flow and also the $\partial/\partial t$ derivatives of the fields, analogous to the spatial domain requirement discussed in the previous Section 4.1. We trained the PINN with different time spans of data $I_t \in \{1.25, 2.5, 5, 10\}$. The longest run covers approximately a full turnover of the LSC. Fig. 8 shows the R^2 and MAE metrics for the temperature reconstruction with data sampled at time intervals of $\Delta t = 0.06$. For a fair comparison, the runs have processed the same number of collocation points per time unit during their training. The evaluation metrics are computed using PINN predictions on plane A (full resolution). It can be observed that, in all cases, the results are worst at the beginning of the time span. Only the two longest runs are long enough for R^2 to reach a plateau, while towards the end of the time span, the results slightly deteriorate again (cf. $I_t = 10$). The best result for a single snapshot is $R^2 = 0.997$ and is obtained with $I_t = 5$ at $t = 3.90$. The results are poorest at the boundaries, but particularly at early time instances, indicating that earlier snapshots aid the PINN in reconstructing subsequent fields, but this effect is not as strong the other way round. From this we conclude, that the performance of the PINN depends on the direction of the time. This phenomenon is related to the irreversible nature of turbulent flows, driven by dissipation. For comparison, we compute the integral time scale from the auto-correlation of time series of all velocity components. Averaging over a total of 648 signals, each with a duration of $I_t = 30$, distributed over 27 positions in the cell, we obtain $\mathcal{T} = 1.5$. As visible in Fig. 8, the PINN requires approximately an integral time scale at the beginning of the temporal training domain until the R^2 result reaches a plateau, which is an analogous observation to the minimum spatial domain size requirement in Section 4.1, which was found to be one integral length

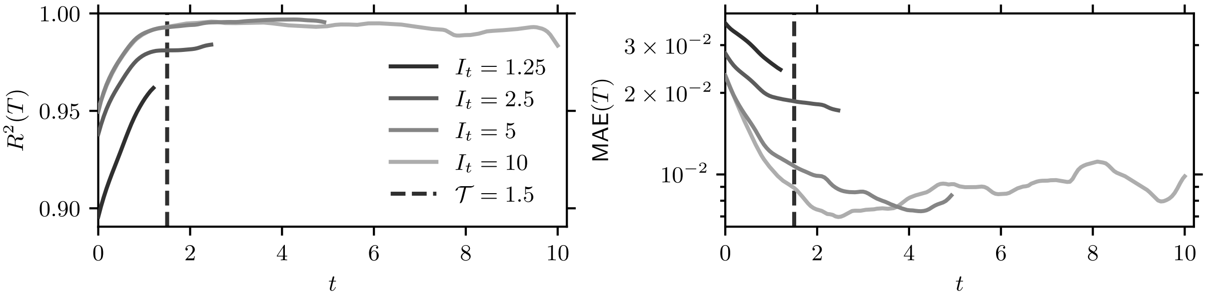

Fig. 8. Comparison of R^2 (left) and MAE (right) values of temperature reconstruction for PINNs trained over different time spans, with a snapshot interval of $\Delta t = 0.06$.

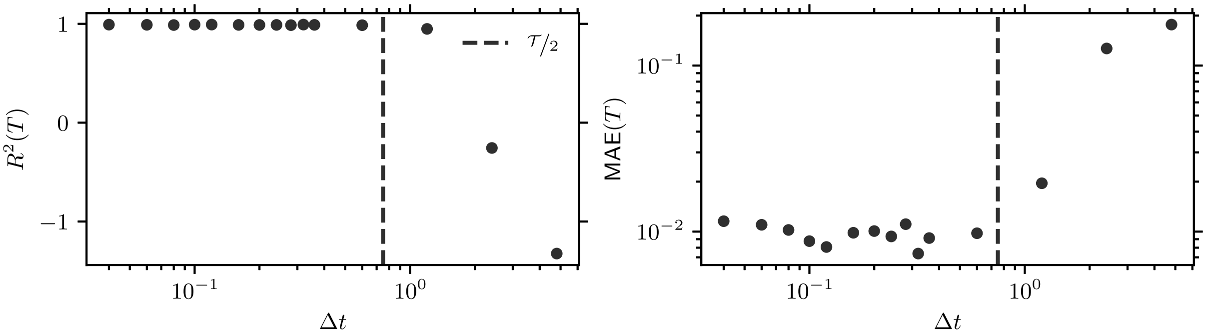

Fig. 9. Variation of the time increment Δt between the snapshots using a time span $I_t = 5$.

scale of the flow. This suggests that, in the absence of initial conditions, the PINN requires a characteristic time scale to constrain its degrees of freedom and accurately reconstruct the physical fields. However, to determine whether a general principle holds, it would be necessary to test RBC cases with varying parameters.

Because it is expensive to achieve full temporal resolution in PIV experiments, we also investigate the influence of the data snapshot acquisition frequency (here reported as the time increment between two consecutive snapshots). Fig. 9 shows the R^2 and MAE metrics of the temperature reconstruction using a time span $I_t = 5$ and different time increments $\Delta t \in [0.04, 4.8]$. We observe that $R^2(T)$ and MAE(T) remain consistent as Δt increases over a wide range of values. However, for a step size larger than half the integral time scale, both metric results collapse. In other words, the time span of an integral time scale must be sampled at least at twice the frequency, to be able to reconstruct the time evolution of the flow structure with the PINN. This is consistent with the Nyquist–Shannon sampling theorem [32], meaning that the intermediate time scales, such as the integral time scale, must be resolved to allow the PINN to reconstruct the temporal evolution in between. This is an interesting result that is probably not specific to our setup, but rather applicable to any reconstruction tasks involving time-dependent flow structures using PINNs.

An overview of the results of the parametric study with different time spans and time step sizes is shown in Fig. 10. The R^2 metric is computed using the full DNS resolution of snapshots with $\Delta t = 0.06$ over the entire temporal training domain ($t > \mathcal{T}$, for cases with sufficiently long time spans). Regardless of Δt, it can be seen that the time spans $I_t = 1.25$ and 2.5 are not long enough and therefore produce the worst results. The best results are obtained for $I_t = 5$, which corresponds to about half of an LSC turnover time, with no significant difference when using either $N_t = 9$, 16 or 42 snapshots.

4.3. Physical analysis of the PINN-reconstructed fields

In this section, we analyze the physical properties of the reconstructed temperature fields. All results presented below are based on a temporal training domain of $I_t = 5$ with a snapshot interval of $\Delta t = 0.075$. For the statistical analysis, twelve cases are considered, each starting at a different time point of the simulation. As observed in the previous Section 4.2, the results within the range of one integral time scale at the beginning of the temporal training domain are the worst. Therefore, we exclude this range from the evaluation.

4.3.1. Temperature reconstruction

Fig. 11 displays the ground truth fields (top row), the reconstructed fields (center row) and the relative pointwise error fields ϵ_{L_2} (bottom row) of all five physical quantities on plane A in the middle of the temporal training domain ($t = I_t/2$). The absolute pressure values cannot be determined because the PDE (Eq. (1)) contains only the pressure gradient ∇p. Therefore, for better comparison, the respective field averages are subtracted from the pressure ground truth and the reconstructed field. Since velocity data is provided for training, the network successfully resolves all visible structures, which is also reflected in the small-scale error fields (Fig. 11, bottom). There is no remarkable difference in the ϵ_{L_2} fields of the out-of-plane component u and the in-plane horizontal component v. Furthermore, the visible structures of the temperature and pressure fields are reconstructed in all details by the PINN (Fig. 11, middle row). All plumes, corner eddies and the boundary layers are successfully reconstructed. However, for the reconstructed temperature and pressure fields, the error field scales are larger and partially coincide with high gradient regions of the true field. This indicates that the unknown quantities inferred by the PINN have the highest uncertainties, particularly in regions with steep gradients.

To further investigate the reconstruction of thermal boundary layers, Fig. 12 shows the temperature profile, averaged over t and y, as a function of the height $\langle T\rangle_{t,y}(z)$ and its variance $\langle T^2\rangle_{t,y}(z) - \langle T\rangle^2_{t,y}(z)$, computed with true and reconstructed fields on plane A from a total time span of $I_t^{\mathrm{tot}} = 42$. The resulting temperature profile has the characteristic shape of the RBC, with the reconstructed mean temperature profile closely matching the ground truth. Slight differences are noticeable in the variance profile within the bulk of the cell. From the peaks of the variance, we can determine the thickness of the thermal boundary layer. Averaging over the upper and lower layer thickness, we obtain the same value of $\delta_{\mathrm{th}}^{\mathrm{DNS}} = \delta_{\mathrm{th}}^{\mathrm{PINN}} = 0.028(3)$. Using the Nusselt

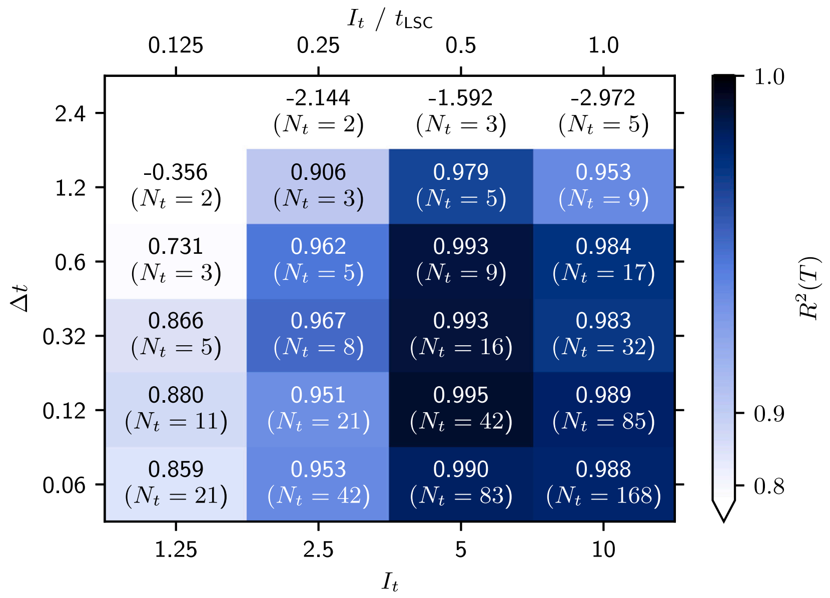

Fig. 10. Variation of the time span I_t and the time increment Δt between the snapshots. N_t is the number of data snapshots used. The average time of an LSC turnover is approximately $t_{\text{LSC}} \approx 10$. The plot has a power-normal color scale with a gamma value of 15, which preserves details in high-scoring areas.

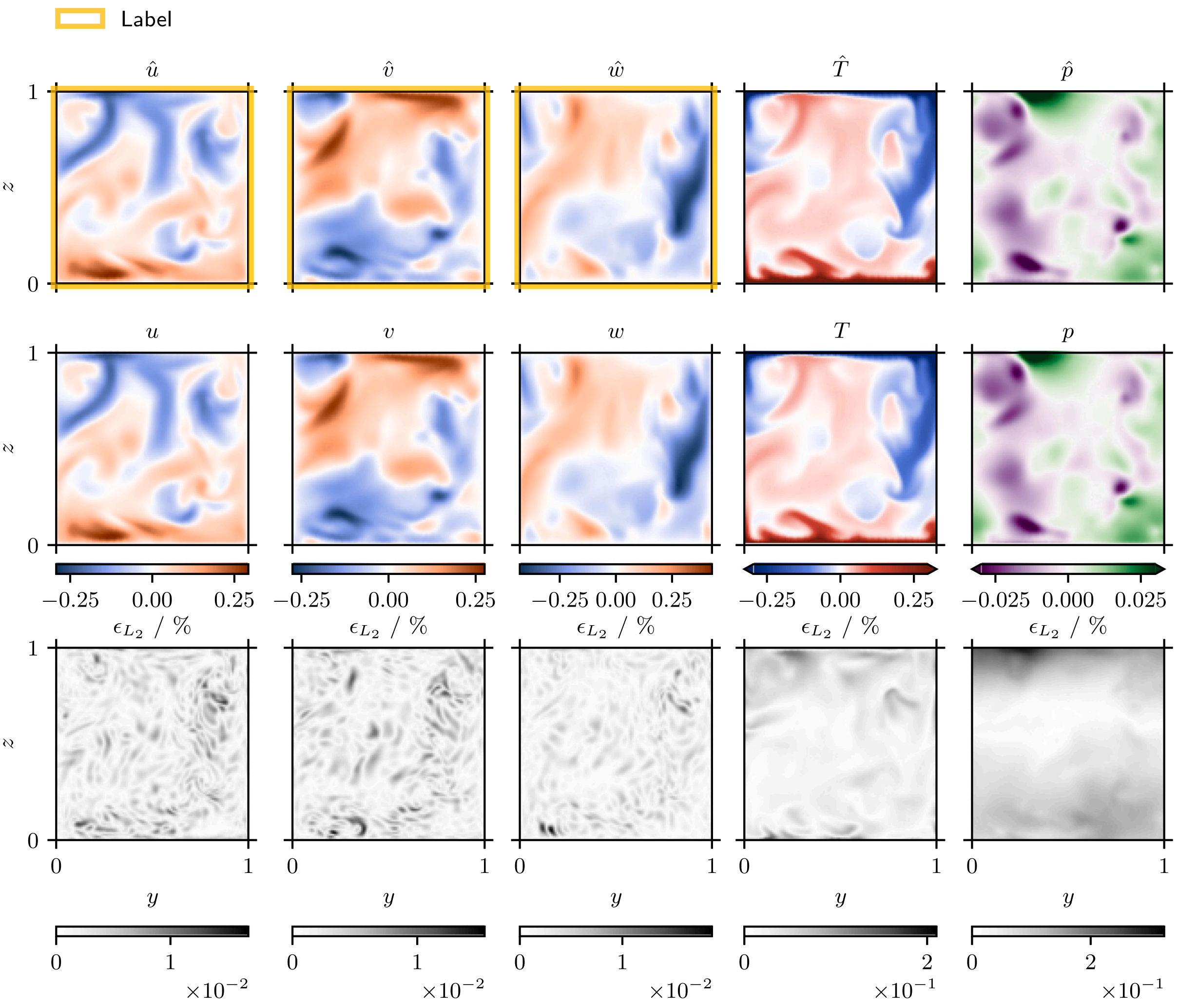

Fig. 11. Comparison of ground truth ($\hat{\cdot}$, top row) and reconstructed fields (center row) on plane A and their relative pointwise error field ϵ_{L_2} (bottom row) at a time instant $t = I_t/2$.

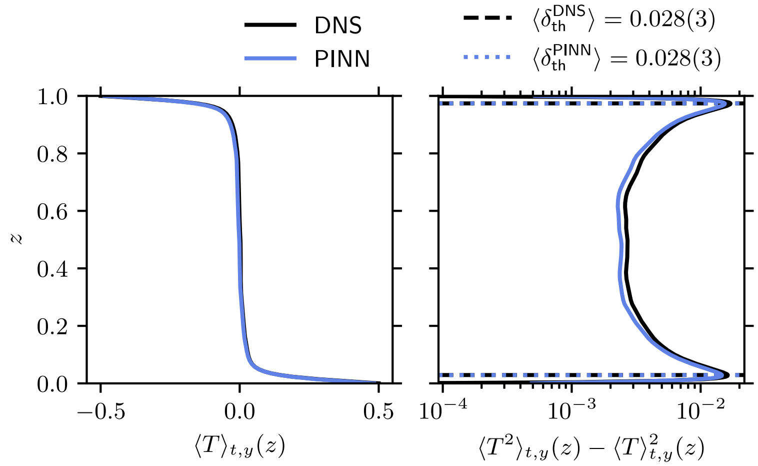

Fig. 12. The temperature field in plane A, averaged over t and y (left), along with its variance (right) as a function of the height z from DNS and PINN reconstructions over a total time span of $I_t = 42$.

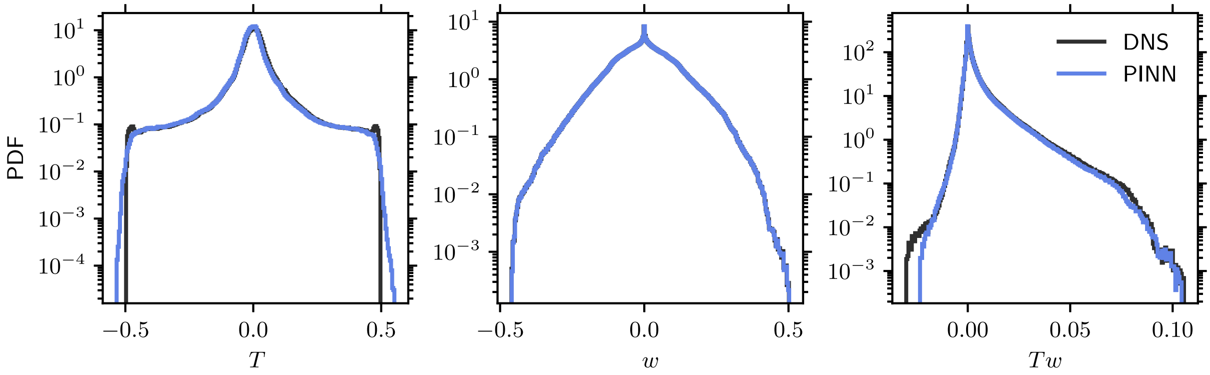

Fig. 13. Probability density functions of the true and reconstructed fields of temperature (left), vertical velocity component (center) and vertical convective heat flux (right) over a total time span of $I_t = 42$.

number Nu = 16.2 derived from the entire cubic cell in [33], we obtain the boundary layer thickness $\delta_{\rm th} = 1/(2\mathrm{Nu}) = 0.031$, which is consistent with the values we obtain in Fig. 12.

To examine the reconstruction of the temperature in more detail, we plot the probability density functions (PDF) of the temperature (left), the vertical velocity component w (middle), and the vertical convective heat flux Tw (right) in Fig. 13. The reconstructed temperature is generally consistent with the DNS histogram (Fig. 13, left); however, for the extreme values, instead of showing a peak, the values $T_h = 0.5$ and $T_c = -0.5$ are exceeded. This is possible due to the linear activation in the last layer. While no deviations are observed for w in Fig. 13 (middle), there is generally good agreement between the DNS and reconstructed fields in the Tw histogram (Fig. 13, right). However, in the left tail, the PINN histogram exhibits a steeper slope and an earlier cut-off. This indicates difficulties in resolving regions where positive temperature is not associated with positive velocity, and vice versa.

Since the training is performed in a 3D domain, we can also examine the reconstructions out-of-plane. Fig. 14 shows a section of the vertical plane at $y = 0.5$. At this view angle, data for training is given only at the vertical center line ($x = x_0$) and training is accomplished in the area $[x_0 - \delta_x/2, x_0 + \delta_x/2]$ marked by the dashed vertical lines. The top row shows the ground truth fields, followed by the reconstructed fields in the middle row, and the relative pointwise error fields ϵ_{L_2} at the bottom. It is noteworthy, that the PINN fully reconstructs the out-of-plane velocity fields within the δ_x domain. Even outside of the domain, where 5 % of the collocation points are placed, the fields highly resemble the ground truth. This is clearly visible in the u field at $x \approx 0.4$ near the bottom (Fig. 14, far left column), where the PINN fully reconstructs a rightward-streaming region. Since only a small fraction of the collocation points are placed outside the collocation layer and the distance to the given data increases, the velocity error fields exhibit the highest relative errors at the left and right edges of the image. Furthermore, also the structures of the temperature and pressure fields are completely captured inside the collocation layer. Notable is the reconstruction of a hot plume at $x \approx 0.45$, which then blurs to the left outside the collocation domain. The same location in the pressure field shows a low pressure region, which is also successfully captured by the PINN.

4.3.2. Viscous and thermal dissipation rates

A major advantage of using a PINN is that the automatic differentiation provides access to all partial derivatives of the five physical fields. This includes derivatives in the out-of-plane direction, even though the data is given in a plane only. As a result, we are able to analyze the viscous (ϵ_u) and thermal (ϵ_T) dissipation rates, which can be derived from Eqs. (1) to (3) (see e.g. [34]), and which are

$$\epsilon_u = \sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}} \sum_{i,j} \left(\frac{\partial u_i}{\partial x_j} \right)^2, \tag{14}$$

$$\epsilon_T = \frac{1}{\sqrt{\mathrm{RaPr}}} \sum_{i} \left(\frac{\partial T}{\partial x_i} \right)^2, \tag{15}$$

where the indices i and j range from 1 to 3, corresponding to the three spatial dimensions. Fig. 15 shows the viscous (top row) and thermal (bottom row) dissipation rates in plane A at a given time instant. The left column shows the ground truth fields $(\hat{\cdot})$ obtained by

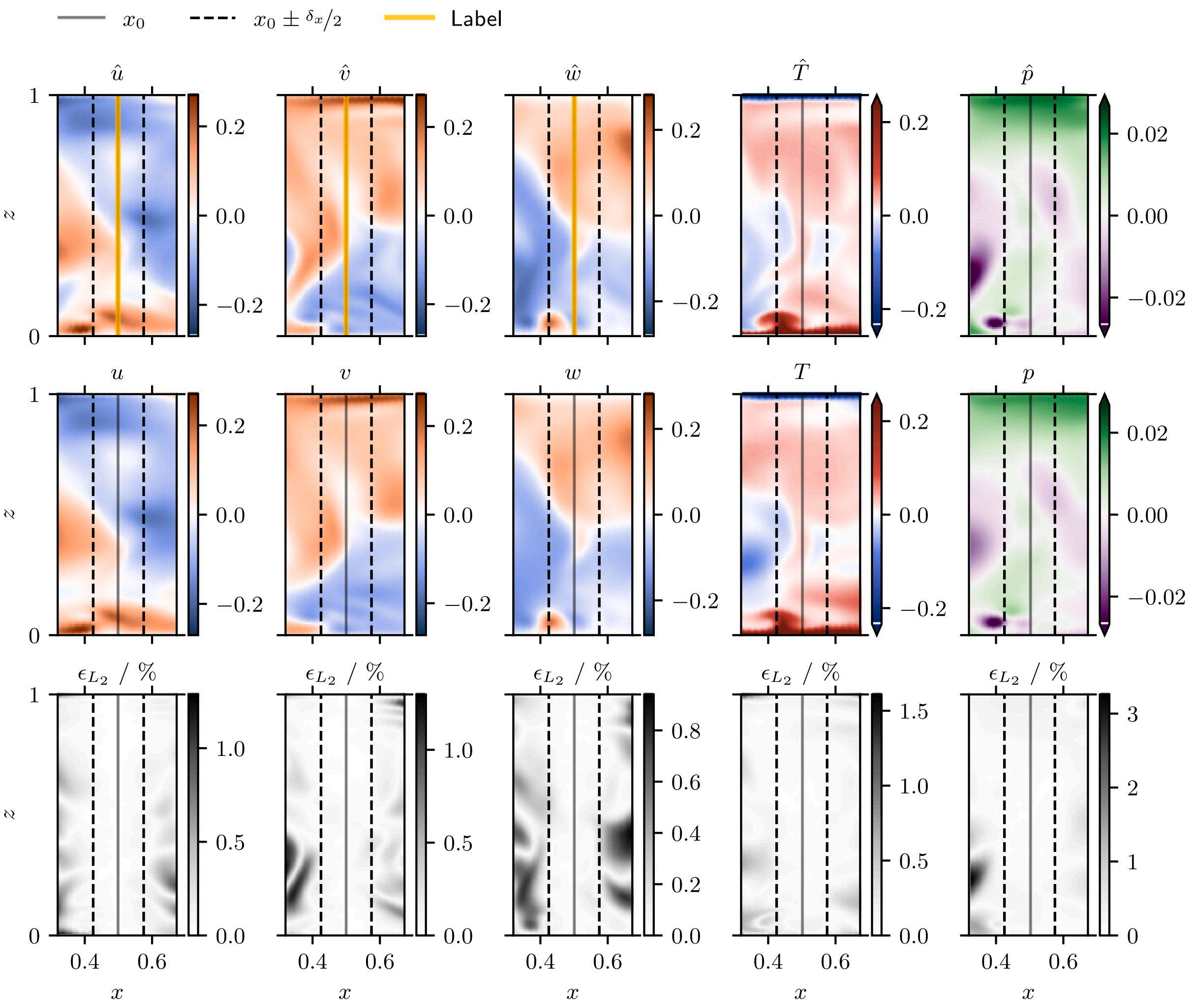

Fig. 14. Comparison of DNS (top row) and reconstructed out-of-plane fields (middle row) and their relative pointwise error fields ϵ_{L_2} (bottom) at $y = 0.5$ and $t = {}^{I_t}/_2$. Velocity data is given at x_0 and the width of the training domain is $\delta_x = 0.15$.

central differences from the DNS fields, the middle column shows the dissipation fields obtained by automatic differentiation from the PINN, and the right column shows the relative pointwise error fields (ϵ_{L_2}). According to the scaling theory proposed by Grossmann and Lohse [35, 36], four different main scaling regimes are obtained, depending on the boundary layer and bulk contributions of the global dissipation rates. Our parameters $Pr = 0.7$ and $Ra = 10^7$ are assigned to the regime, where ϵ_u is dominated by the bulk and ϵ_T is dominated by the boundary layers, which is reflected in Fig. 15. The structures in both reconstructed (viscous and thermal) dissipation fields (Fig. 15, middle column) closely resemble the ground truth (Fig. 15, left column). The error plots (Fig. 15, right column) give a better insight into the location of the uncertainties. For the viscous dissipation, higher error regions are distributed throughout the cell, while for the thermal dissipation, the highest error values are obtained near the top and bottom walls, where the highest dissipation rates are found. Since velocity data is provided for training, we expect the overall error for viscous dissipation to be smaller than for thermal dissipation. The relative L_2 errors computed with data on plane A from the time instant shown in Fig. 15 are $L_2^{\mathrm{rel}}(\epsilon_u) = 9.3\%$ for the viscous dissipation, and $L_2^{\mathrm{rel}}(\epsilon_T) = 19.7\%$ for the thermal dissipation.

4.4. Robustness towards noisy data and missing labels

To assess the potential applicability of our method to more realistic PIV data, we evaluate the robustness of the model to noisy training data. We superimpose Gaussian white noise $\mathcal{N}(0, \sigma^2)$ on the velocity fields with standard deviations σ of different levels: 0 %, 10 %, 20 % and 30 % relative to the maximum amplitude of each velocity component. Generally, in stereo PIV measurements, the out-of-plane velocity component is the one more sensitive to measurement uncertainty; here, for the sake of simplicity, all velocity components are treated equally. So far, we have only used DNS data, which is why the data loss term was weighted the highest in the PINN. However, with error-prone data, the weight of the data loss term (λ_{d}) must be reduced relative to the PDE loss weights, to allow the PINN more flexibility in reconstructing the velocity fields. We tested weights in increments of powers of 10. Fig. 16 shows the vertical velocity field with different levels of Gaussian noise (top), the reconstructed velocity field (center) and the reconstructed temperature field (bottom). The relationship between the weight of the data loss term (λ_{d}) and the weight of the momentum conservation equation (λ_{m}) is shown, since the balance between the individual PDE loss weights remains unchanged (cf. Section 3.3). To better visualize the differences, only one quadrant of plane A is shown. The assimilated velocity fields are very similar to the ground truth data (Fig. 16, second row), although there is a slight loss of amplitude at higher σ. This effect is even more pronounced in the temperature field (Fig. 16, bottom). While the overall structures are generally well captured, their intensity decreases slightly at higher σ, as seen, for example, in the corner plume in the lower right of the temperature fields (Fig. 16, bottom). Fig. 17 shows the corresponding R^2 values for the temperature reconstruction (left) and the velocity reconstruction (right). The R^2 value of the

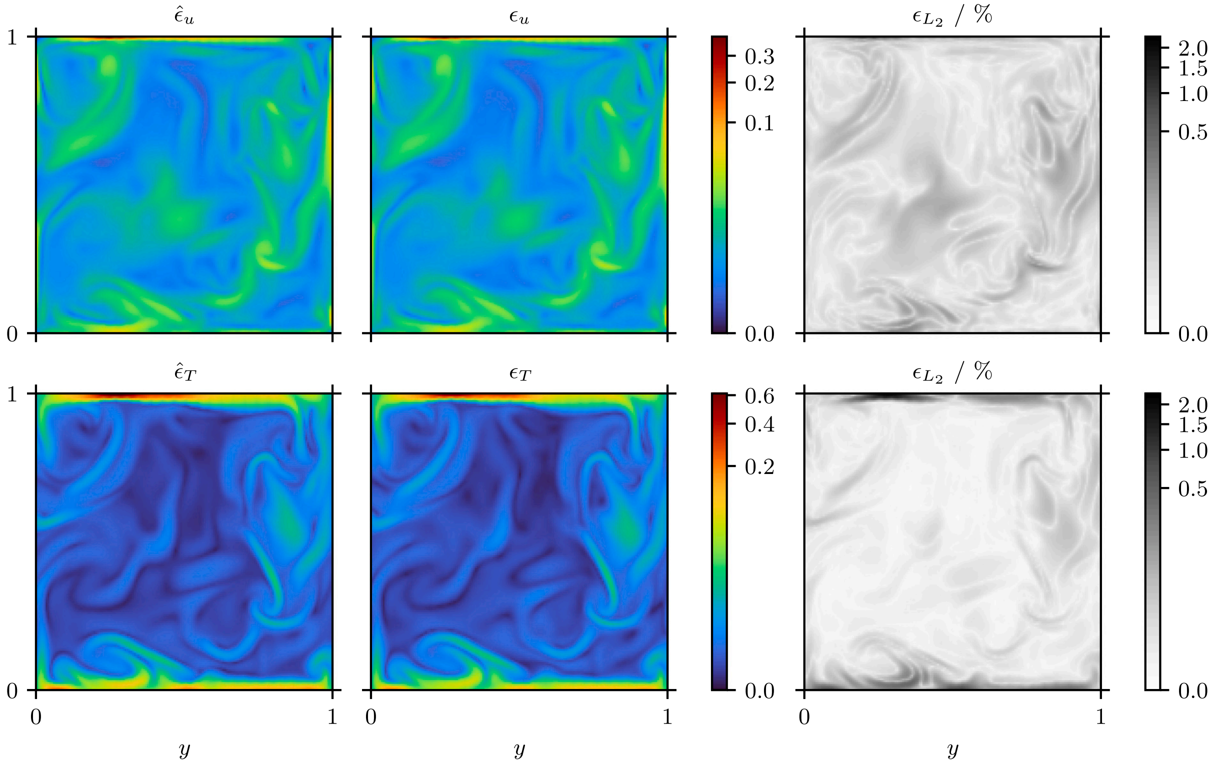

Fig. 15. Viscous (top row) and thermal (bottom row) dissipation rates at plane A at a given time instant ($t = I_t/2$). The ground truth fields $(\hat{\cdot})$ are shown on the left, the PINN predicted fields obtained by automatic differentiation are shown in the middle, and the relative error fields are shown on the right.

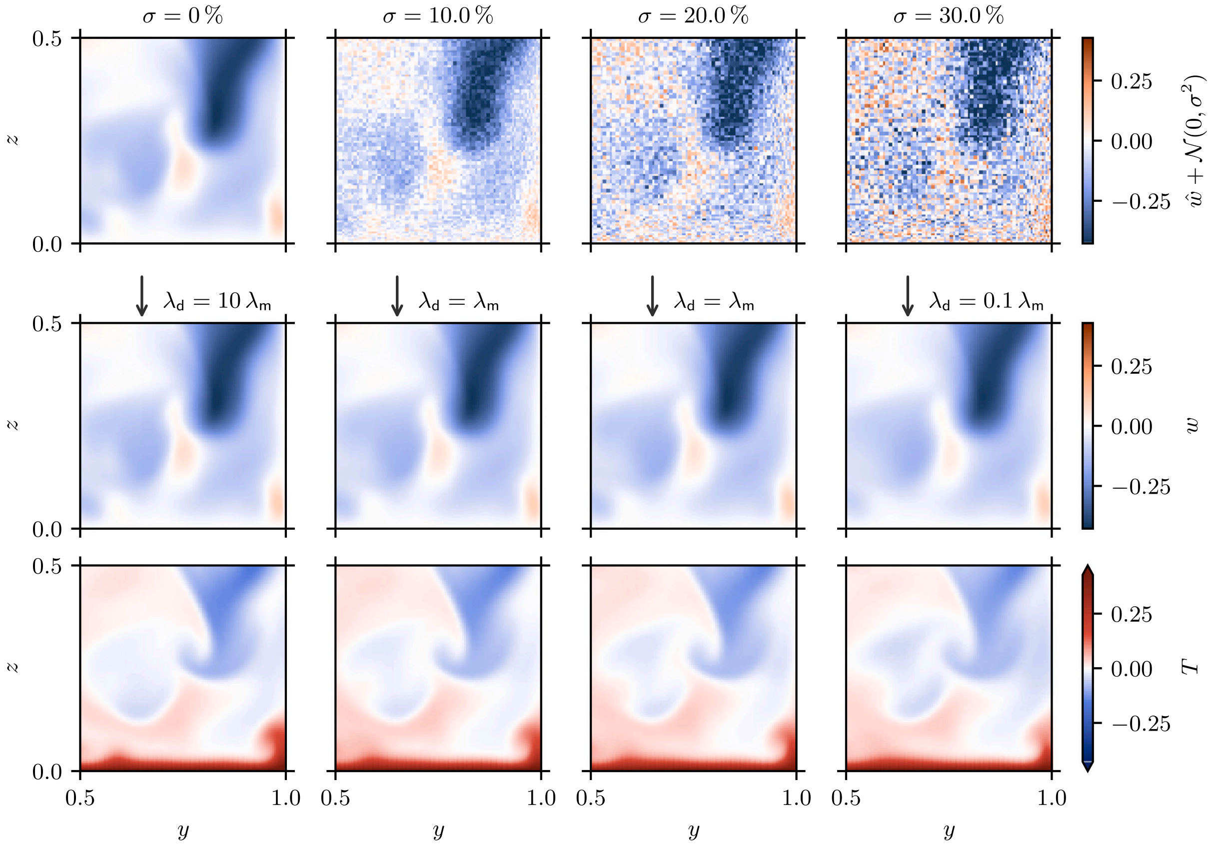

Fig. 16. Ground-truth vertical velocity fields in a quadrant of plane A with varying intensities of Gaussian noise $\mathcal{N}(0, \sigma^2)$ relative to the maximum amplitude of the velocity component (top row), reconstructed vertical velocity fields (middle row) and reconstructed temperature fields (bottom row). The weight of the data loss term (λ_d) is reduced compared to the weights of the PDE loss terms for higher σ.

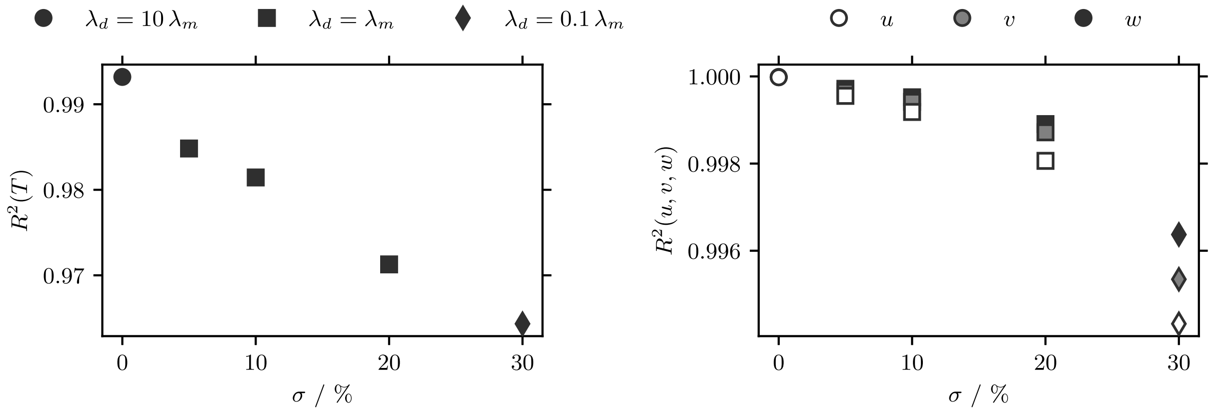

Fig. 17. Comparison of temperature (left) and velocity (right) reconstruction results at different intensities of Gaussian noise $\mathcal{N}(0, \sigma^2)$, expressed as a percentage of the maximum velocity component amplitude. At higher noise ratios, the weighting of the data loss term $\lambda_{\rm d}$ has to be decreased compared to the PDE loss terms.

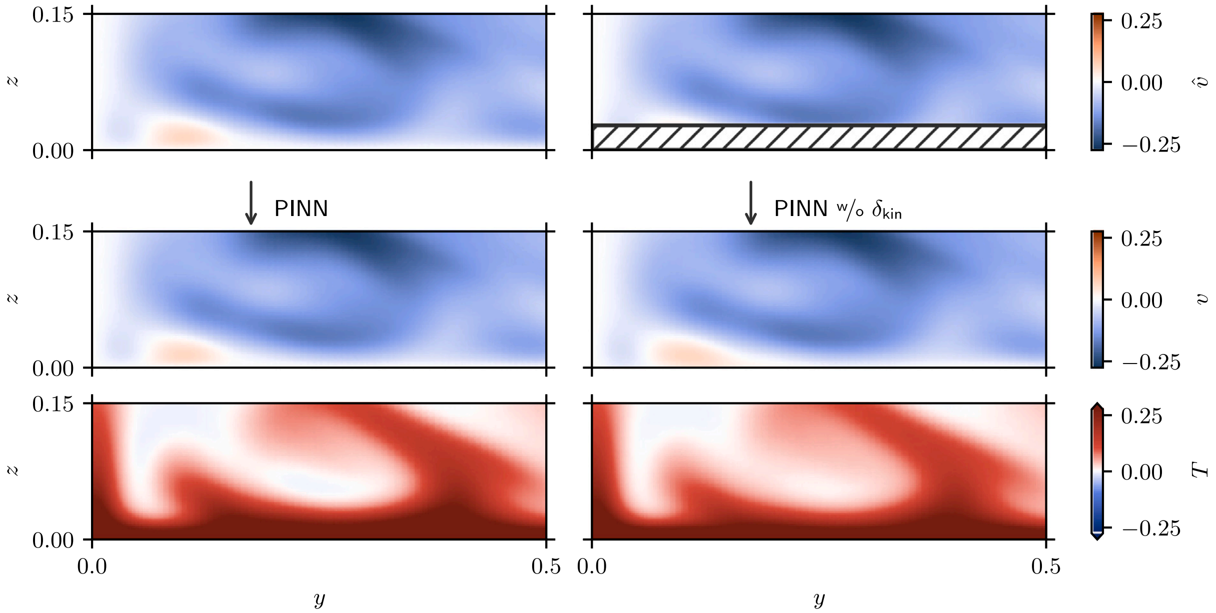

Fig. 18. Comparison of the PINN trained with the spatial DNS resolution on slice A (left) and when omitting the velocity data in the kinematic boundary layer (right). The ground truth horizontal velocity field $\hat{v}$ is shown in the top row, the reconstructed velocity field v in the middle row and the reconstructed temperature field in the bottom row. In order to focus on the boundary layers, only a subset of A is shown.

temperature reconstruction decreases with increasing noise ratio from $R^2(T, 0\%) > 0.99$ to $R^2(T, 30\%) \approx 0.96$. The same applies for the velocity; however, in this case, we still consistently obtain values of $R^2 > 0.99$. When comparing the different velocity components, it is noticeable that the result for u is the worst throughout.

While in DNS the mesh resolution is usually refined in the boundary layers, in experimental setups it is difficult for the tracer particles to reach the boundary layers at all. In addition, in the RBC cell, there may be reflections from the top and bottom plates, making it difficult to measure velocity fields in the vicinity. The kinematic boundary layer thickness for $\mathrm{Ra} = 10^7$ and $\mathrm{Pr} = 0.7$ is estimated to be $\delta_{\rm kin} \approx 0.027$ [23]. Therefore, to further emulate a real PIV dataset, the data points within the kinematic boundary layer are omitted for training. The resulting fields of velocity assimilation and temperature reconstruction on a subarea of plane A are plotted in Fig. 18. The left side shows the case where the PINN is trained with the spatially fully resolved DNS velocity field, and the right side shows the case, where the velocity data within the kinematic boundary layers is omitted. The fields of the ground truth velocity component v are plotted at the top, the reconstructed velocity fields are plotted in the middle, and the reconstructed temperature fields are plotted at the bottom. Both reconstructed velocity fields (Fig. 18, center row) do not show differences in the reconstruction of the bulk flow region. The PINN also manages to reconstruct the velocity field in the area where no data is given. This is particularly evident in the reconstructed velocity field with omitted data labels (Fig. 18, center row, right), at $y \approx 0.1$ near the bottom plate, where a rightward-streaming area is reconstructed despite its absence in the data. The two reconstructed temperature fields also differ only slightly (Fig. 18, bottom row). It can be seen that the PINN manages to reconstruct the thermal boundary layer structure despite the absence of data in this region ($\delta_{\rm th}$ and $\delta_{\rm kin}$ are close for $\mathrm{Pr} = 0.7$); however the thermal boundary layer on the right side of Fig. 18 seems to be thinner.

To examine whether the physical properties are preserved, the temperature profiles near the heating plate $\langle T \rangle_{t,y}(z)$, as a function of the height and averaged over a time span of $I_t = 17.5$ and y, are plotted in Fig. 19 (left) for both cases and the DNS. Indeed, we observe slight differences in the temperature profile (Fig. 19, left) and in the corresponding variance profile $\langle T^2 \rangle_{t,y}(z) - \langle T \rangle^2_{t,y}(z)$ (Fig. 19, right), resulting in a thinner boundary layer thickness of $\delta_{\rm th}^{\mathrm{PINN}\,w/o\,\delta_{\rm kin}} = 0.019(3)$ compared to $\delta_{\rm th}^{\rm PINN} = \delta_{\rm th}^{\rm DNS} = 0.026(3)$.

Since the determination of dissipation rates is of great interest but experimentally challenging (see e.g. the experimental measurement of

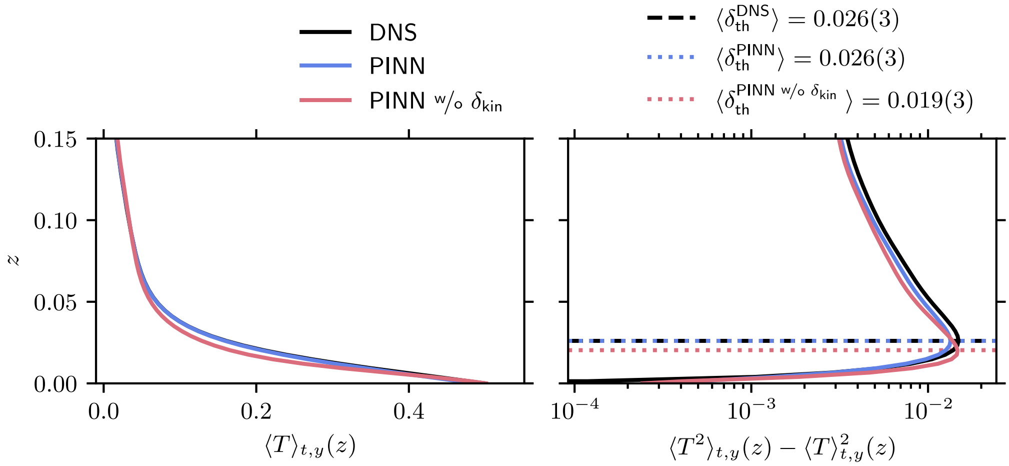

Fig. 19. The temperature field on plane A, averaged over t and y, and as a function of z (left), along with its variance (right) over a total time span of $I_t = 17.5$; comparison of the DNS temperature profile, the reconstructed profile and the reconstructed profile when the PINN is trained without access to the data inside the kinematic boundary layers.

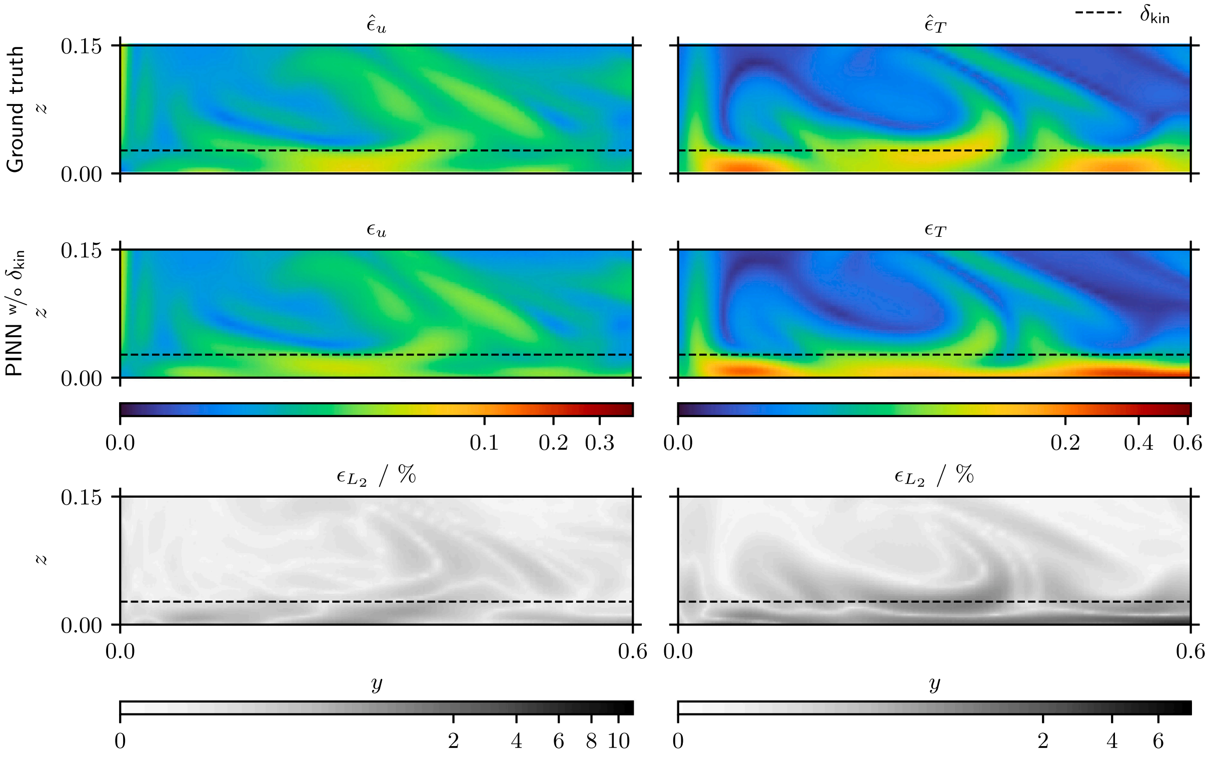

Fig. 20. Viscous (left column) and thermal (right column) dissipation rates on a subset of the plane A. The ground truth fields ($\hat{\cdot}$) are shown at the top, the PINN predicted fields obtained by automatic differentiation are shown in the middle row, and the relative error fields are shown at the bottom. The PINN model is trained on velocity data, excluding kinematic boundary layer data.

the viscous dissipation rate in Xu et al. [37]), we want to study the viscous (ϵ_u, Eq. (14)) and thermal (ϵ_T, Eq. (15)) dissipation rates, when reconstructed from the PINN model trained only on velocity data outside the kinematic boundary layers. Fig. 20 shows the ground truth dissipation fields ($\hat{\cdot}$, top row), the PINN reconstructed fields obtained by automatic differentiation (middle row), and the respective relative error fields (bottom row). The kinematic dissipation rate is shown on the left and the thermal dissipation rate on the right. Despite the lack of data in δ_{kin}, the overall structures of both reconstructed dissipation rates are very similar to the ground truth. Compared to the PINN trained on velocity data in the whole plane (Section 4.3.2), we observe here the highest errors for both (viscous and thermal) dissipation rates inside the boundary layers. While for the viscous dissipation the reconstructed fields inside the boundary layer (Fig. 20, middle row, left) slightly lack intensity, the reconstructed thermal dissipation rate (Fig. 20, middle row, right) shows higher values. This is probably due to the imposed boundary conditions, which result in reduced velocities and increased temperature values. The relative L_2 errors computed with data on plane A from the time instant shown in Fig. 15 are $L_2^{\text{rel}}(\epsilon_u) = 41.9\%$ for the viscous dissipation and $L_2^{\text{rel}}(\epsilon_T) = 68.3\%$ for the thermal dissipation. To distinguish between the error contributions from the boundary layers and the bulk, we compute the relative L_2 errors using only points that are either in the boundary layers or in the bulk. For the viscous dissipation we get $L_2^{\text{rel}}(\epsilon_{u,\text{BL}}) = 50.5\%$ for the boundary layers and $L_2^{\text{rel}}(\epsilon_{u,\text{bulk}}) = 9.3\%$ for the bulk. For the thermal dissipation we get $L_2^{\text{rel}}(\epsilon_{T,\text{BL}}) = 68.7\%$ and $L_2^{\text{rel}}(\epsilon_{T,\text{bulk}}) = 42.1\%$. Table 1 shows the comparison of the L_2^{rel} errors with the PINN trained on the velocity data in the whole plane (Section 4.3.2). We observe that

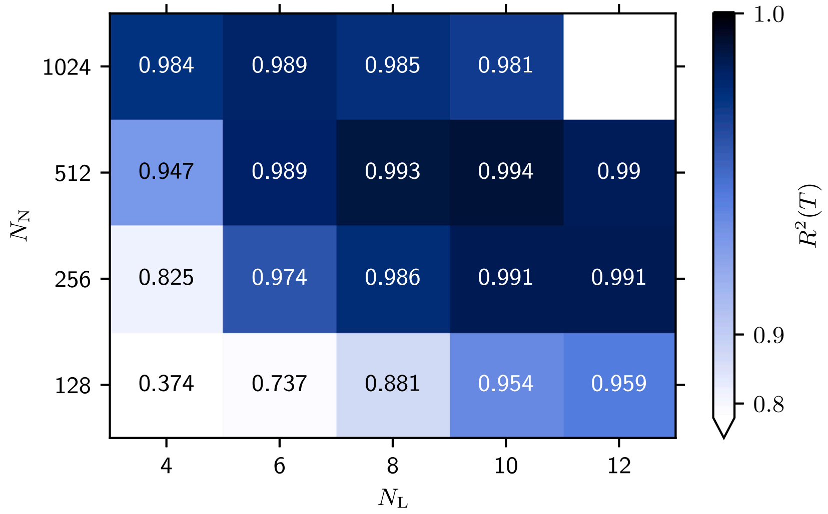

Fig. A.21. Results of the temperature reconstruction (R^2) when varying the number of hidden layers (N_L) and neurons per layer (N_N).

Table 1
Comparison of L_2^{rel} errors when the PINN is trained with velocity data on plane A and when it is trained with velocity data only outside the kinematic boundary layers ($^w/_o$ δ_{kin}). We distinguish between boundary layer (BL) and bulk contributions to the error.

		PINN	PINN $^w/_o$ δ_{kin}
$L_2^{\text{rel}}(\epsilon_u)/\%$	A	9.3	41.9
	BL	9.2	50.5
	Bulk	9.5	9.3
$L_2^{\text{rel}}(\epsilon_T)/\%$	A	19.7	68.3
	BL	19.6	68.7
	Bulk	24.0	42.1

missing data within the kinematic boundary layers affects the accuracy of the dissipation rates in these regions. However, the relative L_2 error for viscous dissipation in the bulk remains unaffected, and the relative L_2 error for thermal dissipation in the bulk is affected, but less than in the boundary layers.

5. Conclusion and outlook

Assuming that 2D3C (two-dimensional, three-component) planar PIV velocity fields are available, we generate the missing temperature field using a PINN and DNS data as the ground truth. We show, that training the PINN with three-component spatially (but not temporally) fully resolved velocity snapshots from DNS and the governing PDEs in an enclosing 3D layer yields promising results with the coefficient of determination exceeding $R^2(T) > 0.995$. It turns out that careful considerations have to be done concerning the spatial training domain. Having access to only planar data, the PINNs training domain in out-of-plane direction (δ_x) needs to be larger than the size of the characteristic length of the medium to big eddies in the flow, i.e. that integral length scale, to be able to fully reconstruct the fields. As the training domain becomes thinner, the agreement with the ground truth becomes increasingly poorer. At $\delta_x = 0$, some of the general structures in the temperature field are reconstructed, but it is not possible to obtain the correct out-of-plane derivatives $^\partial/_{\partial x}$. We apply a weighting of the PDE residuals by distributing collocation points in x-direction normally around the PIV plane, i.e. the region where data is provided for training is weighted highest. Furthermore, we incorporate the knowledge of the DNS grid to define a sampling distribution in the vertical z-direction, placing greater emphasis on the boundary layer and its associated high temperature gradients.

A time span study reveals, that the results are worst at the beginning of the temporal training domain. As initial conditions are not provided, the PINN requires a minimum time span exceeding an integral time scale of the flow to obtain reliable results. This result is not only relevant for our specific case but also to other time-dependent assimilation and reconstruction tasks. In our case, a dataset covering a total time span of half an LSC turnover is found to be a suitable choice. In terms of temporal resolution, the PINN remains robust until the snapshot interval exceeds one half of the integral time scale. This implies that the data must provide at least a framework in which all medium-scale time dynamics are resolved. The PINN can then fill the gaps between the snapshots with physically reasonable fields that remain consistent with the given data.

We showed that the suggested approach allows to reconstruct large and small structures of the temperature as well as pressure fields. We mimicked stereo PIV data by using only planar velocity fields. In a common approach, we used Gaussian noise as a surrogate for potential measurement errors. Even with a high noise ratio of $\sigma = 30\%$, the PINN is still able to reconstruct the velocity with $R^2(\{u, v, w\}) > 0.995$. However, experimental data may not contain such an intense white noise. Instead, it is subjected to colored noise or more systematic measurement uncertainties, such as zones with reflections and mirroring. Since the areas at the lower and upper plate are mostly affected by such reflections and moreover only a few tracer particles reach the boundary layers, we also omitted the data labels inside the kinematic boundary layers. It turns out that the PINN has no difficulties in reconstructing the fields if the data does not contain the top and bottom kinematic boundary layers.

In addition to reconstructing the temperature field, the PINN enables access to all partial derivatives of the physical fields through automatic differentiation. This capability allows for the computation of viscous and thermal dissipation rates. Evaluating the solution on plane A, we found that the PINN is fully able to reconstruct all visible structures in the viscous and thermal dissipation rates. Even when trained without data inside the kinematic boundary layers, it still provides an estimate of the dissipation rates within these layers. Moreover, the accuracy of viscous dissipation in the bulk region remains unaffected under these conditions.

Overall, we can state, that PINNs are fully able to reconstruct temperature fields of turbulent RBC, when having access to only planar snapshots of velocity data. The presented approach can be used to obtain not only temperature fields but also in-plane viscous and thermal dissipation rates, opening up new possibilities for experimenters to study the physics of RBC through planar velocity measurements.

In our case, we used velocity data on a vertical plane; however, it would also be interesting to explore whether temperature reconstruction on a horizontal plane in the RBC cell could be achieved with

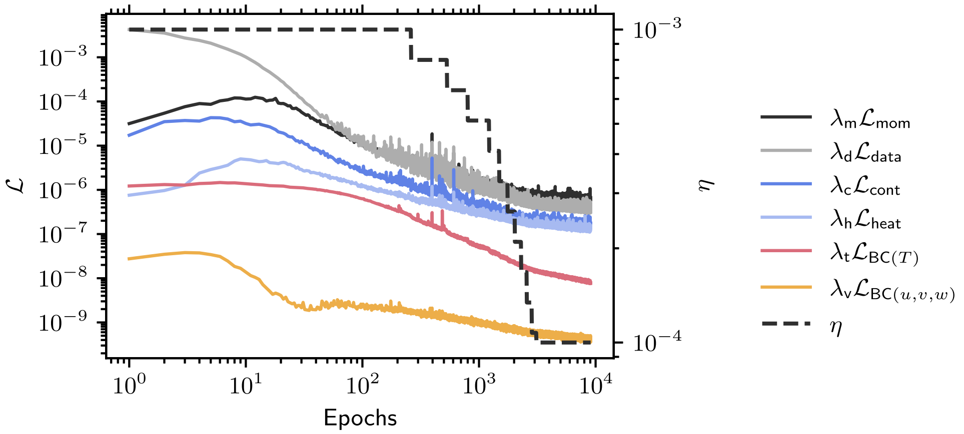

Fig. B.22. Evolution of the weighted training loss components and the learning rate (η) over epochs. For the weight values, refer to Section 3.3.

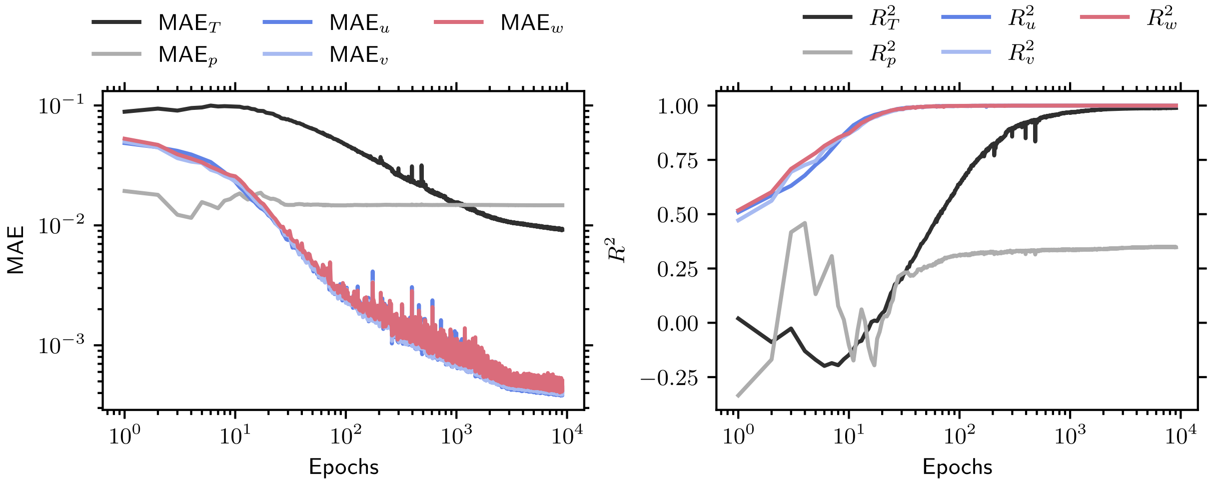

Fig. B.23. Evolution of the validation metrics MAE (left) and R^2 (right) over epochs. 5 % of the dataset is excluded from training and used for validation.

the same level of accuracy. The challenge here is likely to be greater due to the higher out-of-plane velocities, and therefore adjustments to the out-of-plane sampling would be required. Another aspect worth investigating is whether temperature can be reconstructed using only mono-PIV data, i.e. the two in-plane velocity components only. The PINN would likely require additional physical constraints, such as the net mass flux through the plane, to compensate for the lack of out-of-plane velocity information. These aspects will be further investigated and developed in future work.

CRediT authorship contribution statement

Marie-Christine Volk: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Anne Sergent:** Writing – review & editing, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Didier Lucor:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Michael Mommert:** Writing – review & editing, Validation, Supervision, Conceptualization. **Christian Bauer:** Writing – review & editing, Supervision, Resources, Data curation. **Claus Wagner:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Marie-Christine Volk reports financial support was provided by Science and Technology Department of the French Embassy in Germany. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge the financial support of the program Procope-Mobilité of the French Embassy in Germany and the scientific support and HPC resources provided by the German Aerospace Center (DLR). The HPC system CARA is partially funded by “Saxon State Ministry for Economic Affairs, Labour and Transport” and “Federal Ministry for Economic Affairs and Climate Action”. The HPC system CARO is partially funded by “Ministry of Science and Culture of Lower Saxony” and “Federal Ministry for Economic Affairs and Climate Action”.

Appendix A. Layer and neuron variation study

Our default PINN configuration consists of $N_{\mathrm{L}} = 10$ densely connected layers with $N_{\mathrm{N}} = 256$ sine activated neurons each, which is the result of a parameter study. Fig. A.21 shows the resulting $R^2(T)$ values from PINNs with varying numbers of hidden layers and neurons per layer. We observe a trend of increasing performance from the bottom left to the top right, corresponding to the use of more layers and neurons. The smallest tested network of $N_{\mathrm{L}} = 4$ layers and $N_{\mathrm{N}} = 128$ neurons each finds the general pattern of the temperature and pressure fields, but has problems especially in regions with high gradients, leading to the lowest result of $R^2(T) = 0.374$. The expressivity of the network is increased with higher numbers of layers and neurons, leading generally to better results in our case. The best result of $R^2(T) = 0.994$ is obtained using $N_{\mathrm{L}} = 10$ layers and $N_{\mathrm{N}} = 512$ neurons. Increasing the network width to more than $N_{\mathrm{L}} = 10$ layers does not affect the results, and the same holds true for the number of neurons. Good results are already achieved using $N_{\mathrm{N}} = 256$ neurons. Using more neurons is especially beneficial for smaller numbers of layers, but this effect vanishes for $N_{\mathrm{L}} > 10$. Increasing the number of neurons to $N_{\mathrm{N}} = 1024$ leads to a significant increase in computational cost without leading to better $R^2(T)$ values (cf. Fig. A.21).

Appendix B. Training history

An example of the evolution of the several weighted loss functions during training is shown in Fig. B.22 and the corresponding tracking metrics MAE and R^2 are plotted in Fig. B.23. The PINN is trained for 10^4 epochs, starting with a learning rate of $\eta = 1 \times 10^{-3}$, which is gradually reduced to $\eta = 1 \times 10^{-4}$. The loss weight values are given in Section 3.3. For the first ten epochs, only the data loss function $\mathcal{L}_{\mathrm{d}}$ decreases, whereas the PDE loss functions increase ($\mathcal{L}_{\mathrm{mom}}$, $\mathcal{L}_{\mathrm{cont}}$, $\mathcal{L}_{\mathrm{energy}}$) and the boundary condition loss components stagnate ($\mathcal{L}_{\mathrm{BC}(T)}$, $\mathcal{L}_{\mathrm{BC}(u,v,w)}$). After this initial phase, the PINN begins optimizing the PDE and boundary condition losses.

The metric evaluation in Fig. B.23 confirms this observation: during the initial phase, the MAE metric for the velocity fields decreases (left), while the R^2 values of the velocity field increase. After approximately the first ten epochs, the metrics for the temperature field begin to improve.

The metrics for the pressure are generally the worst during the training, as the PDEs involve only the pressure gradient ∇p but not the absolute pressure values.

Appendix C. Sampling strategies

At each training iteration, a new set of points is generated to evaluate the PDE loss. In our approach, we treat all spatiotemporal directions independently. Algorithm 1 outlines the procedure for sampling in the vertical z-direction using a custom distribution. This distribution is constructed using kernel density estimation (KDE) based on the DNS grid points in the z-direction, combined with Latin Hypercube Sampling (LHS). A similar sampling strategy is applied to the remaining coordinates, but without custom distributions: a normal distribution is used in the x-direction, while uniform distributions are employed in the y- and t-directions.

Data availability

Data will be made available on request.

Algorithm 1: Latin Hypercube Sampling from a Kernel Density Estimate (KDE)

```
Input  : $N \in \mathbb{N}$: Number of samples to generate
Input  : $\{z_i\}_{i=1}^{N_z}$: Seed points for KDE (e.g., DNS grid in $[0,1]$)
Input  : $h > 0$: Bandwidth for KDE
Output: $\{b_i\}_{i=1}^{N}$: KDE-based samples via LHS

Step 1: Construct evaluation grid over the data range
$g_j \leftarrow \mathrm{linspace}(\min_i(z_i), \max_i(z_i), L)$

Step 2: Estimate probability density function (PDF) via Gaussian KDE
for $j \leftarrow 1$ to $L$ do
    $f(g_j) \leftarrow \frac{1}{N_z}\sum_{i=1}^{N_z} \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{(g_j - z_i)^2}{2h^2}\right)$

Step 3: Compute and normalize cumulative distribution function (CDF)
$\Delta g \leftarrow g_2 - g_1$
$F(g_j) \leftarrow \sum_{k=1}^{j} f(g_k) \cdot \Delta g$
Normalize: $F(g_j) \leftarrow F(g_j)/F(g_L)$ for all $j$

Step 4: Generate Latin Hypercube Samples (LHS) in $[0,1]$
for $i \leftarrow 1$ to $N$ do
    $s_i \sim \mathcal{U}\left(\frac{i-1}{N}, \frac{i}{N}\right)$
Randomly permute $\{s_i\}_{i=1}^{N}$

Step 5: Map LHS samples to KDE space using inverse CDF
for $i \leftarrow 1$ to $N$ do
    Find $j$ such that $F(g_{j-1}) \le s_i < F(g_j)$
    $b_i \leftarrow g_{j-1} + \frac{s_i - F(g_{j-1})}{F(g_j) - F(g_{j-1}) + \varepsilon} \cdot (g_j - g_{j-1})$
return $\{b_i\}_{i=1}^{N}$
```

References

[1] T. Dehne, P. Lange, D. Schmeling, I. Gores, Experimental evaluation of alternative ceiling-based ventilation systems for long-range passenger aircraft, CEAS Aeronaut. J. 15 (4) (2024) 1031–1050, http://dx.doi.org/10.1007/s13272-024-00739-5.

[2] A. Schröder, D. Schanz, 3D Lagrangian particle tracking in fluid mechanics, Annu. Rev. Fluid Mech. 55 (Volume 55, 2023) (2023) 511–540, http://dx.doi.org/10.1146/annurev-fluid-031822-041721, URL https://www.annualreviews.org/content/journals/10.1146/annurev-fluid-031822-041721, Publisher: Annual Reviews.

[3] R. Barta, C. Bauer, S. Herzog, D. Schiepel, C. Wagner, proPTV: A probability-based particle tracking velocimetry framework, J. Comput. Phys. 514 (2024) 113212, http://dx.doi.org/10.1016/j.jcp.2024.113212, URL https://www.sciencedirect.com/science/article/pii/S0021999124004613.

[4] D. Schiepel, S. Herzog, C. Wagner, Experimental study of turbulent Rayleigh-Bénard convection using large-scale tomo-PIV and high-density PTV, in: M.O. Deville, V. Couaillier, J.-L. Estivalezes, V. Gleize, T. HiêpLê, M. Terracol, S. Vincent (Eds.), Turbulence and Interactions, Springer International Publishing, Cham, 2018, pp. 225–231, http://dx.doi.org/10.1007/978-3-319-60387-2_24.

[5] D. Schmeling, M. Czapp, J. Bosbach, C. Wagner, Development of combined particle image velocimetry and particle image thermography for air flows, in: 2010 14th International Heat Transfer Conference, Washington D.C., USA, 2010, pp. 57–64, URL http://www.asmeconferences.org/ihtc14/, Issue: IHTC14-22774 Number: IHTC14-22774.

[6] D. Schiepel, D. Schmeling, C. Wagner, Simultaneous tomographic particle image velocimetry and thermometry of turbulent Rayleigh–Bénard convection, Meas. Sci. Technol. 32 (9) (2021) 095201, http://dx.doi.org/10.1088/1361-6501/abf095, Publisher: IOP Publishing.

[7] T. Käufer, C. Cierpka, Volumetric Lagrangian temperature and velocity measurements with thermochromic liquid crystals, Meas. Sci. Technol. 35 (3) (2023) 035301, http://dx.doi.org/10.1088/1361-6501/ad16d1, Publisher: IOP Publishing.

[8] D. Schmeling, J. Bosbach, C. Wagner, Simultaneous measurement of temperature and velocity fields in convective air flows, Meas. Sci. Technol. 25 (3)

(2014) 035302, http://dx.doi.org/10.1088/0957-0233/25/3/035302, Publisher: IOP Publishing.

[9] Y. Gasteuil, W.L. Shew, M. Gibert, F. Chillá, B. Castaing, J.-F. Pinton, Lagrangian temperature, velocity, and local heat flux measurement in Rayleigh-Bénard convection, Phys. Rev. Lett. 99 (23) (2007) 234302, http://dx.doi.org/10.1103/PhysRevLett.99.234302, URL https://link.aps.org/doi/10.1103/PhysRevLett.99.234302, Publisher: American Physical Society.

[10] P. Clark Di Leoni, A. Mazzino, L. Biferale, Synchronization to big data: Nudging the Navier-Stokes equations for data assimilation of turbulent flows, Phys. Rev. X 10 (1) (2020) 011023, http://dx.doi.org/10.1103/PhysRevX.10.011023, URL https://link.aps.org/doi/10.1103/PhysRevX.10.011023, Publisher: American Physical Society.

[11] C. Bauer, D. Schiepel, C. Wagner, Assimilation and extension of particle image velocimetry data of turbulent Rayleigh–Bénard convection using direct numerical simulations, Exp. Fluids 63 (1) (2022) 22, http://dx.doi.org/10.1007/s00348-021-03369-3.

[12] S. Weiss, M.S. Emran, J. Bosbach, O. Shishkina, On temperature reconstruction from velocity fields in turbulent Rayleigh–Bénard convection, Int. J. Heat Mass Transfer 242 (2025) 126768, http://dx.doi.org/10.1016/j.ijheatmasstransfer.2025.126768, URL https://www.sciencedirect.com/science/article/pii/S0017931025001097.

[13] 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, http://dx.doi.org/10.1016/j.jcp.2018.10.045, URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[14] S. Wassing, S. Langer, P. Bekemeyer, Physics-informed neural networks for parametric compressible Euler equations, Comput. & Fluids 270 (2024) 106164, http://dx.doi.org/10.1016/j.compfluid.2023.106164, URL https://www.sciencedirect.com/science/article/pii/S0045793023003894.

[15] S. Cai, C. Gray, G.E. Karniadakis, Physics-informed neural networks enhanced particle tracking velocimetry: An example for turbulent jet flow, IEEE Trans. Instrum. Meas. 73 (2024) 1–9, http://dx.doi.org/10.1109/TIM.2024.3398068, URL https://ieeexplore.ieee.org/abstract/document/10522764, Conference Name: IEEE Transactions on Instrumentation and Measurement.

[16] H. Wang, Y. Liu, S. Wang, Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network, Phys. Fluids 34 (1) (2022) 017116, http://dx.doi.org/10.1063/5.0078143.

[17] B. Steinfurth, A. Hassanein, N.A.K. Doan, F. Scarano, Physics-informed neural networks for dense reconstruction of vortex rings from particle tracking velocimetry, Phys. Fluids 36 (9) (2024) 095110, http://dx.doi.org/10.1063/5.0212585.

[18] R. Sundar, D. Majumdar, D. Lucor, S. Sarkar, Physics-informed neural networks modelling for systems with moving immersed boundaries: Application to an unsteady flow past a plunging foil, J. Fluids Struct. 125 (2024) 104066, http://dx.doi.org/10.1016/j.jfluidstructs.2024.104066, URL https://www.sciencedirect.com/science/article/pii/S088997462400001X.

[19] J.D. Toscano, T. Käufer, Z. Wang, M. Maxey, C. Cierpka, G.E. Karniadakis, Inferring turbulent velocity and temperature fields and their statistics from Lagrangian velocity measurements using physics-informed Kolmogorov-Arnold networks, 2024, http://dx.doi.org/10.48550/arXiv.2407.15727, URL http://arxiv.org/abs/2407.15727, arXiv:2407.15727 [physics].

[20] M. Mommert, R. Barta, C. Bauer, M.-C. Volk, C. Wagner, Periodically activated physics-informed neural networks for assimilation tasks for three-dimensional Rayleigh–Bénard convection, Comput. & Fluids 283 (2024) 106419, http://dx.doi.org/10.1016/j.compfluid.2024.106419, URL https://www.sciencedirect.com/science/article/pii/S0045793024002500.

[21] 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, http://dx.doi.org/10.1016/j.jcp.2022.111022, URL https://www.sciencedirect.com/science/article/pii/S0021999122000845.

[22] D. Feldmann, C. Wagner, Direct numerical simulation of fully developed turbulent and oscillatory pipe flows at, J. Turbul. 13 (2012) N32, http://dx.doi.org/10.1080/14685248.2012.708470, Publisher: Taylor & Francis.

[23] O. Shishkina, R.J.A.M. Stevens, S. Grossmann, D. Lohse, Boundary layer structure in turbulent thermal convection and its consequences for the required numerical resolution, New J. Phys. 12 (7) (2010) 075022, http://dx.doi.org/10.1088/1367-2630/12/7/075022.

[24] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (153) (2018) 1–43, URL http://jmlr.org/papers/v18/17-468.html.

[25] V. Sitzmann, J.N.P. Martel, A.W. Bergman, D.B. Lindell, G. Wetzstein, Implicit neural representations with periodic activation functions, 2020, http://dx.doi.org/10.48550/arXiv.2006.09661, URL http://arxiv.org/abs/2006.09661, arXiv:2006.09661 [cs, eess].

[26] T.N.K. Nguyen, T. Dairay, R. Meunier, C. Millet, M. Mougeot, Fixed-budget online adaptive learning for physics-informed neural networks. Towards parameterized problem inference, in: J. Mikyška, C. de Mulatier, M. Paszynski, V.V. Krzhizhanovskaya, J.J. Dongarra, P.M. Sloot (Eds.), Computational Science – ICCS 2023, Springer Nature Switzerland, Cham, 2023, pp. 453–468, http://dx.doi.org/10.1007/978-3-031-36027-5_36.

[27] C. Wu, M. Zhu, Q. Tan, Y. Kartha, L. Lu, A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks, Comput. Methods Appl. Mech. Engrg. 403 (2023) 115671, http://dx.doi.org/10.1016/j.cma.2022.115671, URL https://www.sciencedirect.com/science/article/pii/S0045782522006260.

[28] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (3) (1962) 1065–1076, URL https://www.jstor.org/stable/2237880, Publisher: Institute of Mathematical Statistics.

[29] M. Stein, Large sample properties of simulations using latin hypercube sampling, Technometrics 29 (2) (1987) 143–151, http://dx.doi.org/10.1080/00401706.1987.10488205.

[30] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017, http://dx.doi.org/10.48550/arXiv.1412.6980, URL http://arxiv.org/abs/1412.6980, arXiv:1412.6980 [cs].

[31] A. Thacker, S. Loyer, S. Aubrun, Comparison of turbulence length scales assessed with three measurement systems in increasingly complex turbulent flows, Exp. Therm. Fluid Sci. 34 (5) (2010) 638–645, http://dx.doi.org/10.1016/j.expthermflusci.2009.12.005, URL https://www.sciencedirect.com/science/article/pii/S0894177709002076.

[32] C. Shannon, Communication in the presence of noise, Proc. the IRE 37 (1) (1949) 10–21, http://dx.doi.org/10.1109/JRPROC.1949.232969, URL https://ieeexplore.ieee.org/document/1697831, Conference Name: Proceedings of the IRE.

[33] M. Kaczorowski, K.-L. Chong, K.-Q. Xia, Turbulent flow in the bulk of Rayleigh–Bénard convection: aspect-ratio dependence of the small-scale properties, J. Fluid Mech. 747 (2014) 73–102, http://dx.doi.org/10.1017/jfm.2014.154, URL https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/turbulent-flow-in-the-bulk-of-rayleighbenard-convection-aspectratio-dependence-of-the-smallscale-properties/B9489FD3807D17048C999C12FBDA54CB.

[34] B.I. Shraiman, E.D. Siggia, Heat transport in high-Rayleigh-number convection, Phys. Rev. A 42 (6) (1990) 3650–3653, http://dx.doi.org/10.1103/PhysRevA.42.3650, URL https://link.aps.org/doi/10.1103/PhysRevA.42.3650, Publisher: American Physical Society.

[35] S. Grossmann, D. Lohse, Scaling in thermal convection: a unifying theory, J. Fluid Mech. 407 (2000) 27–56, http://dx.doi.org/10.1017/S0022112099007545, URL https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/scaling-in-thermal-convection-a-unifying-theory/C04F99EF099F794FC23B4939CCDB477F, Publisher: Cambridge University Press.

[36] R.J.A.M. Stevens, E.P.v.d. Poel, S. Grossmann, D. Lohse, The unifying theory of scaling in thermal convection: the updated prefactors, J. Fluid Mech. 730 (2013) 295–308, http://dx.doi.org/10.1017/jfm.2013.298, URL https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/unifying-theory-of-scaling-in-thermal-convection-the-updated-prefactors/71CC88EE08E81AA678985F5CCC1F45A2.

[37] F. Xu, L. Zhang, K.-Q. Xia, Experimental measurement of spatio-temporally resolved energy dissipation rate in turbulent Rayleigh-Bénard convection, J. Fluid Mech. 984 (2024) A8, http://dx.doi.org/10.1017/jfm.2024.164, URL https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/experimental-measurement-of-spatiotemporally-resolved-energy-dissipation-rate-in-turbulent-rayleighbenard-convection/359C262A9092111FCCE5B786C1F3CCA9.