Experiments in Fluids (2023) 64:191
https://doi.org/10.1007/s00348-023-03736-2

RESEARCH ARTICLE

Data-driven estimation of scalar quantities from planar velocity measurements by deep learning applied to temperature in thermal convection

Philipp Teutsch[1] · Theo Käufer[2] · Patrick Mäder[1,3] · Christian Cierpka[2,4]

Received: 29 August 2023 / Revised: 16 October 2023 / Accepted: 26 October 2023 / Published online: 20 November 2023

Abstract

The measurement of the transport of scalar quantities within flows is oftentimes laborious, difficult or even unfeasible. On the other hand, velocity measurement techniques are very advanced and give high-resolution, high-fidelity experimental data. Hence, we explore the capabilities of a deep learning model to predict the scalar quantity, in our case temperature, from measured velocity data. Our method is purely data-driven and based on the u-net architecture and, therefore, well-suited for planar experimental data. We demonstrate the applicability of the u-net on experimental temperature and velocity data, measured in large aspect ratio Rayleigh–Bénard convection at $\mathrm{Pr} = 7.1$ and $\mathrm{Ra} = 2 \times 10^5, 4 \times 10^5, 7 \times 10^5$. We conduct a hyper-parameter optimization and ablation study to ensure appropriate training convergence and test different architectural variations for the u-net. We test two application scenarios that are of interest to experimentalists. One, in which the u-net is trained with data of the same experimental run and one in which the u-net is trained on data of different Ra. Our analysis shows that the u-net can predict temperature fields similar to the measurement data and preserves typical spatial structure sizes. Moreover, the analysis of the heat transfer associated with the temperature showed good agreement when the u-net is trained with data of the same experimental run. The relative difference between measured and reconstructed local heat transfer of the system characterized by the Nusselt number Nu is between 0.3 and 14.1% depending on Ra. We conclude that deep learning has the potential to supplement measurements and can partially alleviate the expense of additional measurement of the scalar quantity.

1 Introduction

In many cases, flows are associated with the transport of scalar quantities, e.g., concentration or temperature. One prominent example is thermal convection that drives many astrophysical and geophysical flows (Schumacher and Sreenivasan 2020; Guervilly et al. 2019; Marshall and Schott 1999; Mapes and Houze 1993). Beyond that, thermal convection plays an important role in various fields of engineering, e.g., the cooling of electronic components (Bessaih and Kadja 2000) or inherent temperature-driven flows in large-scale thermal energy storages (Otto et al. 2023). While measuring the flow itself can be oftentimes challenging (Cierpka et al. 2019), the complexity increases further when the scalar is also measured, e.g., due to deterioration

Philipp Teutsch and Theo Käufer have contributed equally to this work.

Christian Cierpka: Visiting scholar.

✉ Philipp Teutsch
philipp.teutsch@tu-ilmenau.de

✉ Theo Käufer
theo.kaeufer@tu-ilmenau.de

Patrick Mäder
patrick.maeder@tu-ilmenau.de

Christian Cierpka
christian.cierpka@tu-ilmenau.de

[1] Institute of Practical Computer Science and Media Informatics, Technische Universität Ilmenau, Max-Planck-Ring 14, 98684 Ilmenau, Thuringia, Germany

[2] Institute of Thermodynamics and Fluid Mechanics, Technische Universität Ilmenau, Am Helmholtzring 1, 98684 Ilmenau, Thuringia, Germany

[3] Faculty of Biological Sciences, Friedrich-Schiller-Universität, Fürstengraben 1, 07743 Jena, Thuringia, Germany

[4] Department of Biomedical Engineering, Lund University, Ole Römers väg 3, 22100 Lund, Scania, Sweden

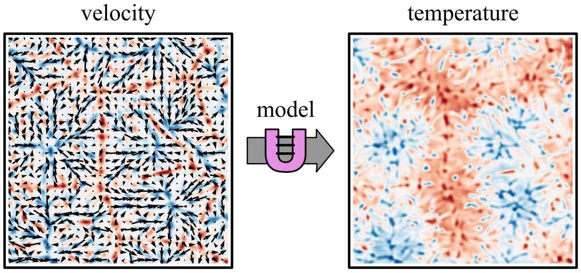

Fig. 1 Conceptual sketch of our goal to predict the temperature by using three velocity components in the horizontal mid-plane of a large aspect ratio RBC experiment

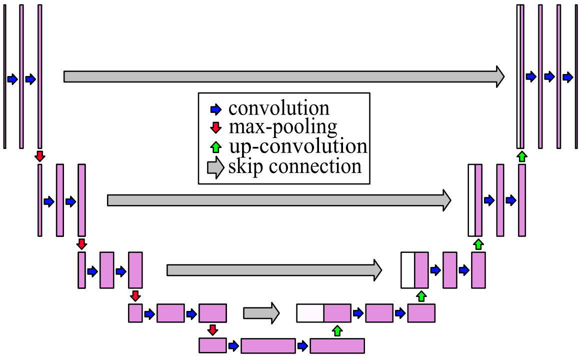

Fig. 2 Basic u-net architecture (based on Ronneberger et al. (2015))

of the fluorescent dye (Sakakibara and Adrian 1999) and increased hardware requirements (Moller et al. 2021). Meanwhile, optical velocity measurements techniques like particle image velocimetry (PIV) are well-established and sophisticated tools in experimental fluid mechanics (Kähler et al. 2016). Hence, novel methods that would assist with the scalar measurement are highly desirable.

Recently, machine learning and deep learning-based methods have emerged as well-suited tools in fluid mechanics (Ling et al. 2016; Mendez et al. 2023; Brunton et al. 2020; Brunton and Kutz 2022; Raissi et al. 2019; Liu et al. 2020; Yu et al. 2023) with and increasing number of applications in flow measurement techniques (Rabault et al. 2017; König et al. 2020; Moller et al. 2020; Sachs et al. 2023), data reduction (Mendez 2022), forecast (Ghazijahani et al. 2022; Heyder and Schumacher 2021; Pandey et al. 2022) and super-resolution (Fukami et al. 2021; Gao et al. 2021). Deep neural networks turned out to be a powerful tool, and effort is spent to make their predictions consistent with physical laws by incorporating the governing equations, making them "physics-informed" (Raissi et al. 2020; Cai et al. 2021a, b). In many cases, solving the governing equations requires knowledge of all gradients, hence demanding fully volumetric measurements and knowledge of the thermal boundary conditions, which still remain the exception (Schiepel et al. 2021; Kashanj and Nobes 2023). Therefore, a purely data-driven model that processes far more common planar data as input would be of great use. In this manuscript, we present a u-net-based model (cp. Figure 2) which is well-suited to process multidimensional data in different fields (Jansson et al. 2017; Zhang et al. 2018; Fonda et al. 2019; Schonfeld et al. 2020) and thus provides a profound studied and versatile basis. The objective of the u-net is to predict the temperature field $\tilde{T}$ for the given velocity fields u_x, u_y and u_z, as conceptualized in Fig. 1. The u-net is trained and tested with experimental temperature and velocity data obtained from joined stereoscopic particle image velocimetry and particle image velocimetry measurements (PIT) in the horizontal mid-plane of a large aspect ratio Rayleigh–Bénard convection (RBC) experiment. The Rayleigh–Bénard setup is a well-studied, simplified model experiment for natural convection and hence ideal to evaluate the u-net performance (Ahlers et al. 2009; Chillà and Schumacher 2012). RBC usually consists of fluid confined by adiabatic side walls, which is heated from below and cooled from above. The resulting dynamical system is governed by the Rayleigh number $\mathrm{Ra} = g\alpha\Delta T H^3/(\nu\kappa)$ and the Prandtl number $\mathrm{Pr} = \nu/\kappa$ which are defined by acceleration due to gravity g, the thermal expansion coefficient α, the temperature difference between heating and cooling plate ΔT, the domain height H, the kinematic viscosity ν and the thermal diffusivity κ of the fluid. Additionally, the aspect ratio $\Gamma = W/H$ as the ratio of domain width W and height H and the container's shape affects the flow (Shishkina 2021; Ahlers et al. 2022). In the present large aspect ratio experiment, so-called turbulent superstructures emerge (Pandey et al. 2018; Stevens Richard et al. 2018; Moller et al. 2022; Käufer et al. 2023).

The remainder of the paper is structured as follows. Section 2 describes the architecture of the u-net together with applied modifications. In Sect. 3, we introduce the data sets as well as the experiment and methods used for their generation. Thereupon, in Sect. 4, we discuss the results of the hyper-parameter study and define two real-world application scenarios in Sect. 5. We subsequently analyze and interpret the prediction of both scenarios in Sect. 6 and conclude with a summary and future research perspectives in Sect. 7.

2 Deep learning model architecture

The u-net proposed by Ronneberger et al. (2015) is an autoencoder-like architecture that consists of two parts, the encoder and the decoder. Depending on the task, the encoder consists of one or multiple fully connected or convolutional layers (LeCun et al. 1989) where each layer provides fewer output values (neurons) than the previous. The decoder is typically constructed as an inverted version of the encoder; thus, the layers inflate their input instead of reducing it. Ronneberger et al. originally proposed u-net for biomedical image segmentation. Its architecture is enriched with additional residual connections that give the decoder additional vision on the different encoder layer outputs. In contrast to conventional residuals, here, the parallel branches are not added together but are appended along the channel dimension. The residuals clearly nullify the original autoencoder idea since the decoder is not able to expand the data solely from its reduced representation. However, this does not affect its applicability for non-reconstruction tasks. This can be tasks where the model performs a local feature extraction instead.

There are already several studies based on u-net, some of which propose improved and more complex u-net architectures (Çiçek et al. 2016; Zhou et al. 2019; Oktay et al. 2018; Wang et al. 2019; Alom et al. 2019; Zhang et al. 2020; Huang et al. 2020). A detailed overview of u-net and its variants is given by Siddique et al. (2021). Most applications here are still related to medical image segmentation/processing. Apart from that, the u-net was successfully utilized in other scientific fields like the analysis of turbulent Rayleigh–Bénard convection (Fonda et al. 2019; Wang et al. 2020; Esmaeilzadeh et al. 2020; Pandey et al. 2020). With each encoder layer, the model captures more abstract visual features of the input snapshot and passes them directly to the decoder using residual connections. This way, the decoder has access to features of different level of detail when step-wise constructing the output scalar. Therefore, we choose to use u-net as the basis for our deep learning experiments.

Figure 2 shows the basic u-net architecture (Ronneberger et al. 2015). It consist of three different types of building blocks convolutional, max-pooling and up-convolution layers. An encoder layer is a sequence of 3 convolutional layers with an increasing number of 3×3 filters with a rectified linear unit (ReLU) activation. Here, the path splits, and one branch forms a skip connection to the decoder and the other goes through a max-pooling layer following the U-shape. A corresponding decoder block concatenates data directly from the encoder via the skip connection with data from the previous layer that is passed through an up-convolution which is composed of an up-sampling layer and a convolutional layer with 2×2 filters. The concatenated data goes through a sequence of three convolutional layers with a decreasing number of 3×3 filters and ReLU activation. The final output is passed through an additional 1×1 convolutional layer. Further details regarding the architecture can be found in the original publication by Ronneberger et al. (2015).

For our study, we utilize the u-net architecture as model backbone but altered and added concepts that we deemed more appropriate for the problem of temperature field prediction. These concepts are batch normalization, sub-pixel convolutions instead of up-sampling layers and several alternative ReLU-like activations. We discuss each concept in more detail below.

2.1 Activation function

The original u-net architecture uses ReLU activation after its convolutional layers (except the output layer). Although the ReLU (Eq. 10) activation function, which was first proposed by Nair and Hinton (2010), is still widely and successfully applied, more sophisticated successors were proposed in recent years. A well-known downside of the original ReLU is the "dying ReLU" problem. This problem appears when the model enters a state (weight configuration) where all inputs of the ReLU are non-positive and thus produce zero gradients. Leaky ReLU is a variant that ensures nonzero gradients in the whole domain, which often makes it the superior choice (Xu et al. 2020). Clevert et al. (2015) proposed the exponential linear unit (ELU). ELU allows negative activations and moves their mean toward zero, which alleviates the bias shift effect and speeds up training. A similar improvement inspired by the ideas of ReLU, dropout (Hinton et al. 2012) and Krueger et al. (2016) is the Gaussian error linear unit (GELU) activation. It uses the cumulative normal distribution function to imitate the stochastic effect of dropout and zoneout depending on the input o by applying the Gaussian cumulative distribution function Φ and have shown to improve classification results for MNIST and CIFAR-10/100 by the inventors (Hendrycks and Gimpel 2016). Along with the introduction of self-normalizing neural networks (SNNs) came another ReLU-like activation function scaled exponential linear unit (SELU) (Klambauer et al. 2017). The authors proved the self-normalizing properties of SELU for $\lambda_{01} \approx 1.0507$ and $\alpha_{01} \approx 1.6733$, which stabilize training of deeper neural networks. They show that SELU preserves approximately zero mean and unit variance through multiple layers.

All these variants were proposed to improve training behavior and their utilization can have regularizing and stabilizing effects or improve training speed and the overall prediction performance of the model. However, their actual usefulness cannot be taken for granted for any case. Hence, we test different activation functions for our setup to elaborate which one is most suitable for our needs.

In the supplementary material, Fig. 18 displays the above activation functions.

2.2 Batch normalization

Batch normalization layers (Ioffe and Szegedy 2015) are typically added before the layer activation (e.g., ReLU) to process each batch of inputs o as follows:

$$\hat{o} = \gamma \frac{o - \mu_o}{\sqrt{\sigma_o^2 + \epsilon}} + \beta, \tag{1}$$

where μ_o and σ_o are the mean and standard deviation of the moving average batch, respectively. ϵ is added for numerical stability and γ and β are trainable shift and scale parameters. Batch normalization ensures that the input o is scaled and shifted to have β mean and γ variance, thereby helping the model to reduce the internal covariate shift of layer inputs, increasing training speed and preventing exploding gradients for deeper networks and larger learning rates, resulting in overall better model performance (Santurkar et al. 2018; Bjorck et al. 2018).

2.3 Sub-pixel convolution

Shi et al. (2016) proposed sub-pixel convolutions instead of conventional bi-linear or bi-cubic up-sampling to improve the reconstruction and super-resolution (SR) quality of deep neural networks. They were frequently used for state-of-the-art deep learning approaches to perform image super-resolution tasks (Wang et al. 2020). The sub-pixel layer consists of two operations. To scale two-dimensional input feature maps c_{in} by a factor r a sub-pixel layer first applies a convolution operation with $c_{\mathrm{out}} = r^2 c_{\mathrm{in}}$ filters. After that, a pixel-shuffle operation is applied. It re-arranges the output feature maps of the convolution in a deterministic way, as shown in Fig. 3. The number of feature maps can also be referred to as the number of channels, e.g., an RGB image as input means three input channels, where each channel provides the information for either red, green or blue. The figure displays an exemplary application of a sup-pixel convolution. On the left side, the low-resolution (LR) layer input is shown with only a single channel. Four convolutional filters are applied, and their outputs are shown in the middle of the figure. Finally, in the outer right area, the re-arranged convolution outputs that form the super-resolution (SR) version of the layer input can be seen. Here, the small squares stand for pixels and their colors indicate the corresponding convolutional filter. During the re-arrangement, the most upper-left pixel of each filter's output are combined to form the four most upper-left pixels of the SR output. The same pattern applies to all other pixel positions. This behavior persists during inference as well as during training, such that the convolutional filters are trained best to predict their specific sub-pixel value. Instead of RGB pixels, our data consists of the different velocity components and temperature, which are also spatially dependent. This makes our data likewise suitable to be processed by super-resolution layers.

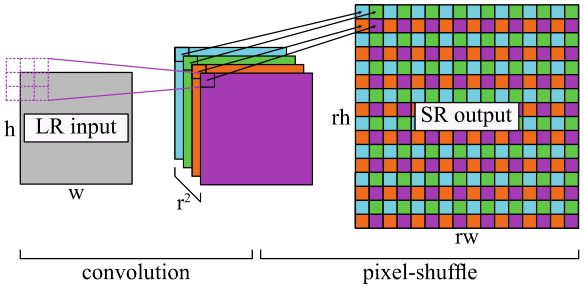

Fig. 3 Operation of a sub-pixel layer that takes a single low-resolution (LR) input feature map (just $c_{\mathrm{in}} = 1$ channel), with width w and height h, and creates a corresponding (super-resolution) SR output feature map with scaling factor r

3 Experiment

The training, testing, and validation data used in this paper were obtained from experiments in a cuboid aspect ratio $\Gamma = 25$ RBC cell with a lateral size $W = 700$ mm and a height $H = 28$ mm. A schematic view of the experiment is shown in Fig. 4.

The working fluid inside the cell is water at a mean temperature $T_{\mathrm{ref}} \approx 19.5\,^{\circ}$C, which has a Prandtl number Pr=7.1 The fluid is confined by glass sidewalls, an aluminum heating plate at the bottom and a cooling plate assembly made from glass at the top. The cooling plate assembly consists of two horizontally oriented, slightly separated glass sheets. This arrangement allows the adjustment of the cooling plate's temperature by cooling water while maintaining optical transparency. The temperature of both plates can be precisely and independently controlled by adjusting the temperature of the through-flowing water. The transparent cooling plate enables the application of optical measurement techniques for spatially and temporally resolved velocity and temperature measurements of a large part of the flow domain, which otherwise would not be possible due to the small height of the experiment. To visualize the flow, polymer-encapsulated thermochromic liquid crystals (TLCs) are added to the fluid. When illuminated by a continuous wavelength spectrum, the color of light reflected by the particles is temperature-dependent, which is leveraged to estimate the temperature

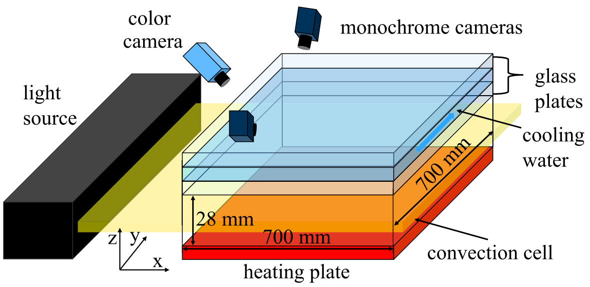

Fig. 4 Sketch of the experiment and the measurement arrangement

of the fluid by particle image thermometry (PIT) (Dabiri 2009). Beyond the temperature, the color is also influenced by other factors, most importantly the observation angle ϕ, which is the angle between illumination and observation, and the illumination spectrum (Moller et al. 2019). During the experiments, a slice of approximately 3 mm thickness around the horizontal mid-plane was illuminated by a custom white light LED array with integrated light sheet optics. The flow in the measurement plane was then observed by two monochrome cameras under an oblique viewing angle of $\approx 55°$, which recorded the data used for the stereoscopic PIV processing (Prasad 2000; Raffel et al. 2018). An additional color camera was positioned with an observation angle $\phi \approx 65°$, which was used to capture the particle color for the subsequent PIT analysis. The observation angle of $\phi \approx 65°$ was chosen as a trade-off to achieve a large field of view with high sensitivity in the desired temperature range. For this study, a processing scheme based on the local calibration curves of the hue was used. Therefore, the color images were sliced into interrogation windows, and the color values were averaged within the windows and then transformed into the hue, saturation, and value (HSV) color space. Since the hue is sufficient to describe the color change, saturation and value are neglected. During the calibration, a known temperature was adjusted in the domain, and color images were recorded for several temperature steps across the temperature range of the experiment. Thereby the relation between the temperature and the hue was obtained for each interrogation window individually. This local calibration effectively eliminates the influence of the observation angle on the reflected color. During the processing of the convection data, the hue calibration curves were then used to derive the temperatures from the measured hue values. Further details on the technique can be found in Moller et al. (2019, 2020).

Utilizing the aforementioned experiment and methods, three data sets at different $\mathrm{Ra} \in \{2\times10^5, 4\times10^5, 7\times10^5\}$ of spatially and temporally resolved, simultaneously measured planar temperature and velocity data were obtained. The data sets consist of long-time data of all three velocity components $\mathbf{u} = (u_x, u_y, u_z)$ and the temperature T to catch the reorganization of the superstructures (Moller et al. 2022). Since the focus of the experimental setup is set on the investigation of the large-scale structures, neither the Batchelor scale nor the Kolmogorov scale are resolved. For the highest Rayleigh number, the Kolmogorov scale and the Batchelor scale are estimated to be 1.8 and 0.7 mm, which is smaller than the interrogation window size of 3.2 mm. Due to the otherwise enormous amount of data, a special recording scheme, visualized in Fig. 5, was applied.

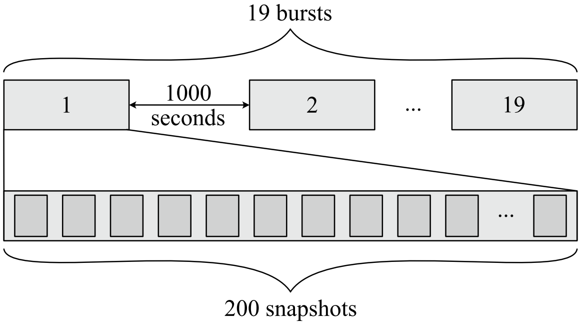

Fig. 5 Structure of the data sets and bursts

Table 1 Overview of the data sets at different Rayleigh numbers Ra and the most important parameters. Data adapted from Moller (2022)

Ra	T_h (°C)	T_c (°C)	$\tilde{x} \times \tilde{y}$	t_f (s)	$\tilde{t}_\mathrm{total}$
2×10^5	19.78	19.08	16.1 × 16.6	4.51	5.39×10^3
4×10^5	20.22	18.76	16.1 × 16.6	3.12	7.79×10^3
7×10^5	20.87	18.43	16.1 × 16.6	2.40	1.01×10^4

T_h and T_c denote the heating and cooling plate temperatures, $\tilde{x} \times \tilde{y}$ the size of the measurement plane in multiples of the cell height H, t_f the free-fall time in seconds, and t_total the total measurement duration in free-fall times

Instead of continuously recording, the data were recorded in bursts of 200 corresponding to at least 44 free-fall times t_f (Eq. 2) seconds with gaps of approximately 1000 s or at least 221 t_f in between. Hence, the data provide sufficient variety within the burst and between the bursts to be used as training data. From each burst, 200 snapshots were obtained. For each data set, a total of 19 bursts were recorded and processed, totaling a number of 5700 snapshots. The initial processing of the data were performed on different grids, which were slightly coarser for $\mathrm{Ra} \in \{2\times10^5, 4\times10^5\}$. To train the u-net, the data were interpolated on a common grid, resulting in a slight up-sampling for the lower two Ra . For the analysis, the data were transferred into their non-dimensional representation denoted by ~ according to

$$\tilde{t} = t/t_\mathrm{f} = \frac{t}{\sqrt{H/\alpha g(T_\mathrm{h} - T_\mathrm{c})}}, \tag{2}$$

$$\tilde{T} = \frac{T - T_\mathrm{c}}{T_\mathrm{h} - T_\mathrm{c}}, \tag{3}$$

$$\tilde{\mathbf{u}} = (\tilde{u}_x, \tilde{u}_y, \tilde{u}_z) = \frac{\mathbf{u}}{\sqrt{H\alpha g(T_\mathrm{h} - T_\mathrm{c})}}, \tag{4}$$

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z}) = \frac{\mathbf{x}}{H}. \tag{5}$$

The most important parameters of the experimental run are shown in Table 1; further details can be found in Moller (2022) and Käufer et al. (2023).

4 Model training and optimization

For our network training procedure, we split the data into a training, a validation and a test subset. We split the dataset between the bursts to avoid highly similar snapshots in different subsets. Thus, the snapshots of any burst are all within the same subset. Similar to the representation of a color image, we consider each snapshot as a two-dimensional tensor of size 295 × 287 and four channels that contain the velocity and temperature. We consider the velocity fields $\tilde{u}$ as input and the temperature field $\tilde{T}$ in the remaining channel as the target or ground truth for our model. We want to point out that ground truth refers to the measured temperature, which itself differs from the exact fluid temperature due to measurement uncertainties. The training was structured into epochs, and each epoch into steps. Within each epoch, the model sees each training snapshot exactly once. The number of epochs is determined by early stopping (Prechelt 2012). Thus, we watch the validation loss and stop the training when it did not improve for 10 patience epochs. This is a best practice value high enough to ensure the convergence of the model while keeping the training duration reasonable. If the number of patience epochs is chosen too small, it may result in an under-fitting of the model. The processing of one batch within an epoch is followed by a stochastic gradient descent back-propagation step to adjust the model weights. Other hyper-parameters are the learning rate η and the channel factor ρ. The latter is bound to the model architecture and determines the number of channels used in each layer. Therefore, the l-th encoder layer has $c_e^{(l)} = \rho 2^{l-1}$ channels. Due to u-net's symmetric architecture, the l-th decoder layer has $c_d^{(l)} = \rho 2^{L-l}$ channels, where L is the number of encoder respectively decoder layers. During the hyper-parameter tuning of η and ρ, we use a fixed batch size of 64 snapshots to ensure a stable and efficient training behavior. A too small batch size leads to increased training duration and less general validity of each gradient descent step. A too large batch size may exceed the memory limits. It is important to note that the hyper-parameter tuning in Sect. 4.1 is only valid for this batch size.

For transparency and reproducibility, we provide configurations for η and ρ for all tests. Since we train our model to solve a regression task, we use the mean squared error (MSE) (6) as loss function for all trained models. The MSE of a single dimensionless temperature snapshot $\tilde{T}_{\mathrm{GT}}$ and its prediction $\tilde{T}_{\mathrm{P}}$ with $d = \|\tilde{T}_{\mathrm{GT}}\| = \|\hat{T}_{\mathrm{P}}\|$ values each is formalized as:

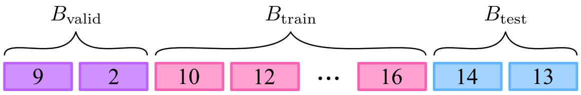

Fig. 6 Data bursts split into subsets for the P0 scenario

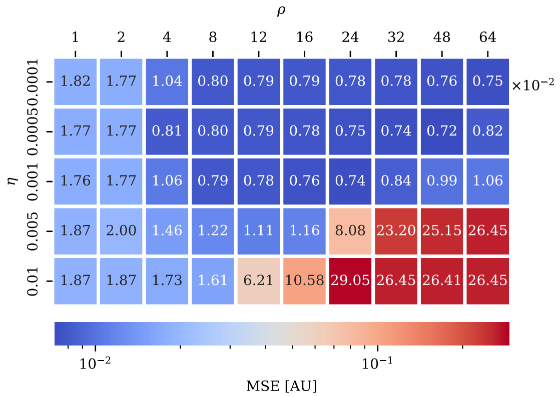

Fig. 7 Mean of the validation MSEs for all tested η-ρ-combinations as heat map

$$\mathrm{MSE}(\tilde{T}_{\mathrm{GT}}, \tilde{T}_{\mathrm{P}}) = \frac{1}{d} \sum_{i=1}^{d} (\tilde{T}_{\mathrm{GT},i} - \tilde{T}_{\mathrm{P},i})^2. \tag{6}$$

In the consideration of validation losses as in Fig. 7, we report the average MSE across all validation snapshots. We performed all experiments on an NVIDIA A40 GPU with 40GB VRAM.

4.1 Hyper-parameter optimization

Before we begin to train our model, we structure the data as sketched in Fig. 6. We shuffle the whole dataset B on burst level. Then, we divide it into a training set B_{train}, a validation set B_{valid} and a testing set B_{test}. These subsets hold $b_{\mathrm{train}} := \|B_{\mathrm{train}}\| = 15$ bursts for training and $b_{\mathrm{valid}} := \|B_{\mathrm{valid}}\| = b_{\mathrm{test}} := \|B_{\mathrm{test}}\| = 2$ bursts for validation and test phase respectively, which leaves us with an approximate 80 : 10 : 10 split. We refer to this training scenario as **P0**.

Before moving on to more complex scenarios, we determine appropriate values for η and ρ. Therefore, we perform a full grid search for these hyper-parameters with a basic u-net model and P0 conditions (see Fig. 6) with $\mathrm{Ra} = 2 \times 10^5$. We test all combinations of $\eta \in \{0.01, 0.005, 0.001, 0.0005, 0.00001\}$ and $\rho \in \{8, 12, 16, 24, 32, 48, 64\}$. Each combination is replicated 5 times with a different random seed $s \in \{0, 1, 2, 3, 4\}$.

Table 2 Results of the ablation study

Up-sampling	Activation	Batch-norm	MAE [AU]: $\mu \pm \sigma$
Nearest	ReLU	No	0.06676 ± 0.00166
Nearest	leaky ReLU	No	0.08064 ± 0.01382
Nearest	ELU	No	0.07316 ± 0.00917
Nearest	GELU	No	0.08911 ± 0.01113
Nearest	SELU	No	0.08191 ± 0.01331
Sub-pixel	ReLU	No	0.07228 ± 0.00923
Nearest	ReLU	Yes	$\mathbf{0.06320 \pm 0.00044}$

Figure 7 provides the MSE averaged over all replications of each tested hyper-parameter combination. It indicates that the predictions tend to worsen for increasing η and ρ. It also appears that smaller $\rho \leq 8$ are more robust against larger η. It also seems that for learning rates $\eta \leq 0.0001$, it is beneficial to increase the channel factor. Though, the performance gain is rather small, given the memory effort. Therefore, we set our limit to the capacity of a single GPU. For our setup, this is the case for a $\rho = 64$. Here, we receive the most promising validation MSE for $\rho = 48$ and a learning rate of $\eta = 0.0005$.

For the ablation study, we want to add the different architectural variations described in Sect. 2 separately to the model architecture and observe the effect on performance. Table 2 shows the different architecture configurations and the resulting mean absolute error (MAE) (7) that is calculated as follows:

$$\mathrm{MAE}(\tilde{T}_{\mathrm{GT}}, \tilde{T}_{\mathrm{P}}) = \frac{1}{d} \sum_{i=1}^{d} \|\tilde{T}_{\mathrm{GT},i} - \tilde{T}_{\mathrm{P},i}\|. \tag{7}$$

The smallest and therefore best MAE result is printed in bold. As mentioned above, we run each experiment with five different initializations. Therefore, the reported MAE is actually the average of these runs together with the standard deviation. Due to the preprocessing of our data, the target temperature values lie within 0 and 1. We ensure the same limits for the predicted temperature with a sigmoid activation at the last decoder layer. This is due to the fact that the sigmoid activation also projects to the interval [0, 1]. A ReLU-like activation, as used for the other layers, would allow unbounded positive output values, which is not desirable for our task. It also implies that the MAE is within the same limits and is given in arbitrary units (AU). The ablation study shows that only the batch normalization improves the performance of the base model. Hence, we add batch normalization layers to all our subsequent models. These also increase the resilience of our model against larger learning rates and thus allow for faster training convergence. In addition, the dependence on the random seed is less prominent (as indicated by the low standard deviation) which is important if the model will be applied for other tasks in the future.

5 Application scenarios

So far, we shuffled our data before splitting. This, however, does not resample many possible applications. Since the u-net is trained on the measurement data, these must be either generated during the experimental run in temporal succession, e.g., at the beginning or taken from a different run. Hence, we consider two different scenarios. First, we investigate the neural network's performance when trained on the data of the same experimental run, one model for each Ra. We call this scenario **P1**. This scenario could help to further expand the accessible measurement time when the measurement of the scalar quantity is not possible anymore. For example, due to the degeneration of the tracer or dye to visualize the scalar quantity due to photobleaching for LIF (Sakakibara and Adrian 1999) or luminescent two-color tracer particle measurements (Massing et al. 2016) limits the time in which scalar measurements can be performed while velocity measurements are still viable. Another reason could be the increased computer memory requirements for the scalar measurement, e.g., the combination of two-color LIF and planar time-series PIV increases the amount of data by 200 % compared to simple PIV setup (Sakakibara and Adrian 1999).

For this scenario, it is essential to know how much training data are needed for reliable results, or in other words, for how long simultaneous scalar and velocity measurements are required until the u-net can replace the scalar measurements. Therefore, we systematically investigated the influence of B_{train} on the loss, as conceptualized in Fig. 8. It shows out of which bursts the training B_{train} (pink), validation B_{valid} (purple) and test subset B_{train} (blue) are composed for different amounts of training data b_{train} in scenario P1. Here, B_{valid} and B_{test} are fixed and consist of burst 1 and 2 respectively burst 18 and 19 in all P1 experiments.

The first P1 results are presented in Fig. 9, which shows the MAE on the test subset for different b_{train}. We observe that the MAE decreases for increased b_{train}. However, considering the value of the MAE, the improvement of additional training data is limited, especially when three or more bursts are used for training. Hence, we considered $b_{\mathrm{train}} = 3$, which

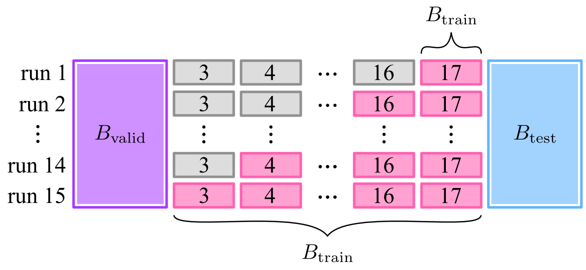

Fig. 8 Data bursts split into subsets to produce results for different b_{train} in the P1 scenario

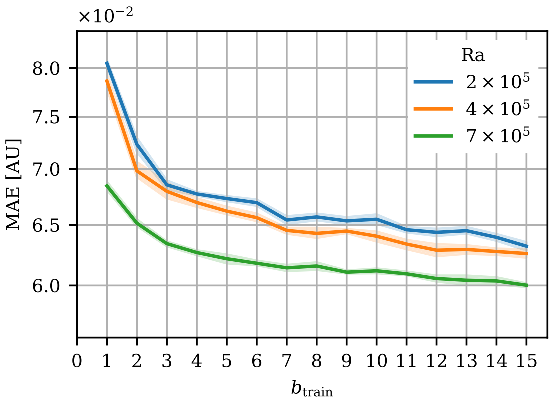

Fig. 9 MAE for different b_{train} in the P1 scenario

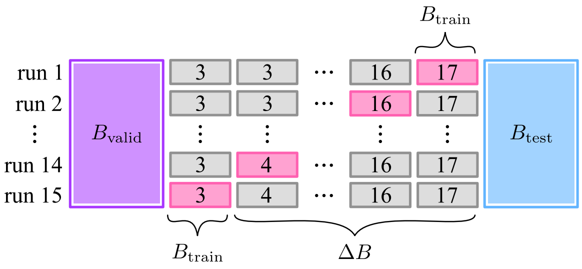

Fig. 10 Data bursts split into subsets to produce results for different $\Delta b = \|\Delta B\|$ in the P1 scenario

corresponds to a measurement time of approximately an hour, as an appropriate trade-off between the u-net performance and effort on generating the training data for further experiments. At first glance, it seems surprising that MAE for a fixed B_{train} decreases with increasing Ra considering that the flow becomes increasingly turbulent but likewise the free-fall time t_f as the characteristic time scale of the flow decreases. Hence, a burst at higher Ra contains more temporal information. Furthermore, the turbulent superstructures that are characteristic of the flow are more distinct for lower Ra (Käufer et al. 2023).

Considering these results is insufficient to ensure that the u-net generalizes well. It additionally requires testing whether the neural network learns the characteristics of the flow and does not rely upon temporal information. Therefore, we systematically investigate the effect of temporal distance, measured in the number of bursts Δb between the last training burst and the first testing burst. We refer to these bursts as ΔB as shown in Fig. 10.

To analyze the effect, we plotted the MAE in dependence of Δb in Fig. 11.

The plot shows some fluctuations, but overall, no systematic dependence of the MAE on ΔB for any Ra, which indicates that our deep learning model does not rely on temporal similarities between testing and training data which decreases with increasing ΔB.

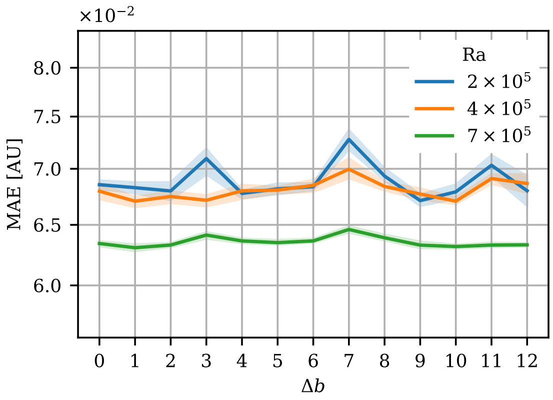

Fig. 11 MAE for different Δb and fix $b_{\mathrm{train}} = 3$

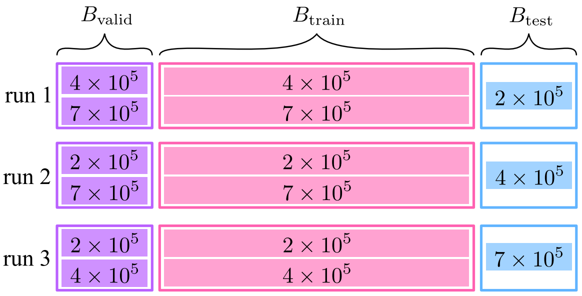

Fig. 12 Data bursts split into subsets for the P2 scenario

In a nutshell, our systematic investigation showed that the u-net requires relatively little training data and produces time-independent predictions.

The second scenario we consider is training the u-net on data from a different experimental run with a slightly different Ra, which we call scenario **P2**. When previously obtained data from similar experimental conditions are available, new measurements of the scalar quantity are not feasible anymore. For this scenario, we train the u-net on data of two Rayleigh numbers and apply the model on data of the remaining Ra experiment as sketched in Fig. 12, which means for one case interpolation and extrapolation for the others.

In total, 30 bursts of two different Ra are used for training. Since the training and prediction are performed on different experimental runs, the data are time-independent by default.

6 Results

In Sects. 4 and 5, we showed that the u-net can predict temperature from velocity data after an initial training phase and identified a channel factor $\rho = 48$ and a learning rate

Table 3 Test results of the P1 models (same Ra)

Ra	MAE		
	P1 [K]: $\mu \pm \sigma$	P1 [AU]: $\mu \pm \sigma$	Naive [AU]
2×10^5	0.04424 ± 0.00031	0.06320 ± 0.00044	0.1093
4×10^5	0.09138 ± 0.00069	0.06259 ± 0.00047	0.1108
7×10^5	0.14645 ± 0.00046	0.06002 ± 0.00019	0.1121

Table 4 Test results of the P2 models (varying Ra)

Ra	MAE		
	P2 [K]: $\mu \pm \sigma$	P2 [AU]: $\mu \pm \sigma$	Naive [AU]
2×10^5	0.05877 ± 0.00181	0.08396 ± 0.00259	0.1022
4×10^5	0.10810 ± 0.00350	0.07404 ± 0.00240	0.1097
7×10^5	0.18366 ± 0.00503	0.07527 ± 0.00206	0.1288

$\eta = 0.0005$ as optimal hyper-parameter. We defined two different scenarios P1 and P2 as potential use cases and determined three bursts of training data as optimum between model performance and data generation effort. To fully evaluate the u-net's performance on the given task, a rigorous physical interpretation and comparison of the results is required. The subsequent analysis is performed only on the test data, which the u-net has never seen during training.

6.1 Temperature prediction

We start by comparing the MAEs (reported in Kelvin and arbitrary unit) of the final trained model u-net for scenario P1 (Table 3) and P2 (Table 4) and of a respective naive approach that simply shifts and scales the vertical velocity field to approximate temperature. This way, the predicted temperature field is composed as $\tilde{T}_{\text{naive}} = \sigma_T \frac{\tilde{\mathbf{u}}_z - \mu_z}{\sigma_z} + \tilde{T}_{\text{mean}}$. Here, μ_z and σ_z are the average vertical velocity and its standard deviation, while σ_T corresponds to the standard deviation of the temperature. $\tilde{T}_{\text{mean}} = 0.5$ denotes the mean temperature in the horizontal mid-plane which would emerge under the idealized Boussinesq assumption.

While a fluid mechanical interpretation of the MAE is not straightforward, it nevertheless is a clear indicator of the model's performance. In Tables 3 and 4, the MAE is reported for the dimensionless temperature fields and lies within 0 to 1, just as in Table 2.

When comparing the different MAE values, we clearly see that the u-net in any scenario outperforms the naive approach by an order of magnitude in MAE, demonstrating the usefulness of the u-net. Furthermore, the u-net trained in the P1 scenario achieves better prediction results compared to the u-net trained in the P2 scenario. This is expected since the training data in the P1 scenario is generated from the same experimental run at the same Ra and thermal boundary conditions and, therefore, inheres a better representation of distinct flow compared to the u-net trained in the P2 scenario. With the single exception of the Ra $= 4 \times 10^5$ case in the P2 scenario, the MAE of the u-net predictions decreases with Ra. While this, at first glance, seems to be counter-intuitive since flows get more complex with increasing Ra, it is a consequence of the measurement technique and turbulent superstructures. On the one hand, the experiment and the measurements were designed and performed to investigate the development of large-scale turbulent superstructures, and due to the inherent trade-off between the field of view and spatial resolution, the measurements do neither resolve the smallest temperature nor velocity structures, and these are averaged out. On the other hand, the turbulent superstructures are more distinct and pronounced at lower Ra, and the temperature fields appear smoother at higher Ra (Moller et al. 2022), which tends to be beneficial for the u-net. The lower MAE for the P2 u-net predictions at Ra $= 4 \times 10^5$ is a consequence of the training data, which only for this Ra incorporates a lower and higher Ra. Hence, the u-net performs an "interpolation" compared to the "extrapolation" of the other P2 cases.

To gain more physical insights, we continue with comparing exemplary instantaneous snapshots of the temperature fields for Ra $= 4 \times 10^5$ shown in Fig. 13.

Here, we contrast the ground truth temperature $\tilde{T}_{\text{GT}}$ with the temperature predicted by the u-net trained with scenario P1 $\tilde{T}_{\text{P1}}$ and scenario P2 $\tilde{T}_{\text{P2}}$. We observe that the dominant structures in the temperature field, the so-called turbulent superstructures, are clearly distinguishable in both predicted temperature fields. However, small-scale features appear to be smoothed out in the predictions. This observation also persists in the fields of the temperature differences $\tilde{T}_{\text{GT}} - \tilde{T}_{\text{naive,P1}}$, which resembles the difference of the naive approach, $\tilde{T}_{\text{GT}} - \tilde{T}_{\text{P1}}$ and $\tilde{T}_{\text{GT}} - \tilde{T}_{\text{P2}}$. Looking at the $\tilde{T}_{\text{GT}} - \tilde{T}_{\text{naive,P1}}$ field, we observe differences significantly larger than those of any u-net prediction. Hence, the naive approach is not suitable at all. For the difference $\tilde{T}_{\text{GT}} - \tilde{T}_{\text{P1}}$, we note that for large parts of the field, the absolute value of the deviation is below 0.1. However, there are localized spots, especially for $\tilde{T}_{\text{GT}} - \tilde{T}_{\text{P1}} > 0$, where the deviations are slightly larger. Nevertheless, the absolute value for 90% of the differences is below 0.15, and larger outliers are unlikely since the temperature measurement is itself associated with measurement uncertainty. Likewise, we observe localized spots of higher deviations in the difference field $\tilde{T}_{\text{GT}} - \tilde{T}_{\text{P2}}$, but additionally, it seems to be somewhat biased toward $\tilde{T}_{\text{GT}} - \tilde{T}_{\text{P2}} < 0$ which indicates a slight overestimation of the temperature predicted by the u-net trained in scenario P2. Still, the absolute value for 90% of the differences is below 0.15. To further quantify how well ground truth and prediction align, we computed the Pearson correlation coefficient $C(\tilde{T}_{\text{GT}}, \tilde{T}_{\text{P}})$ (8) for each $\tilde{T}_{\text{GT}}$ the corresponding $\tilde{T}_{\text{P}}$ as:

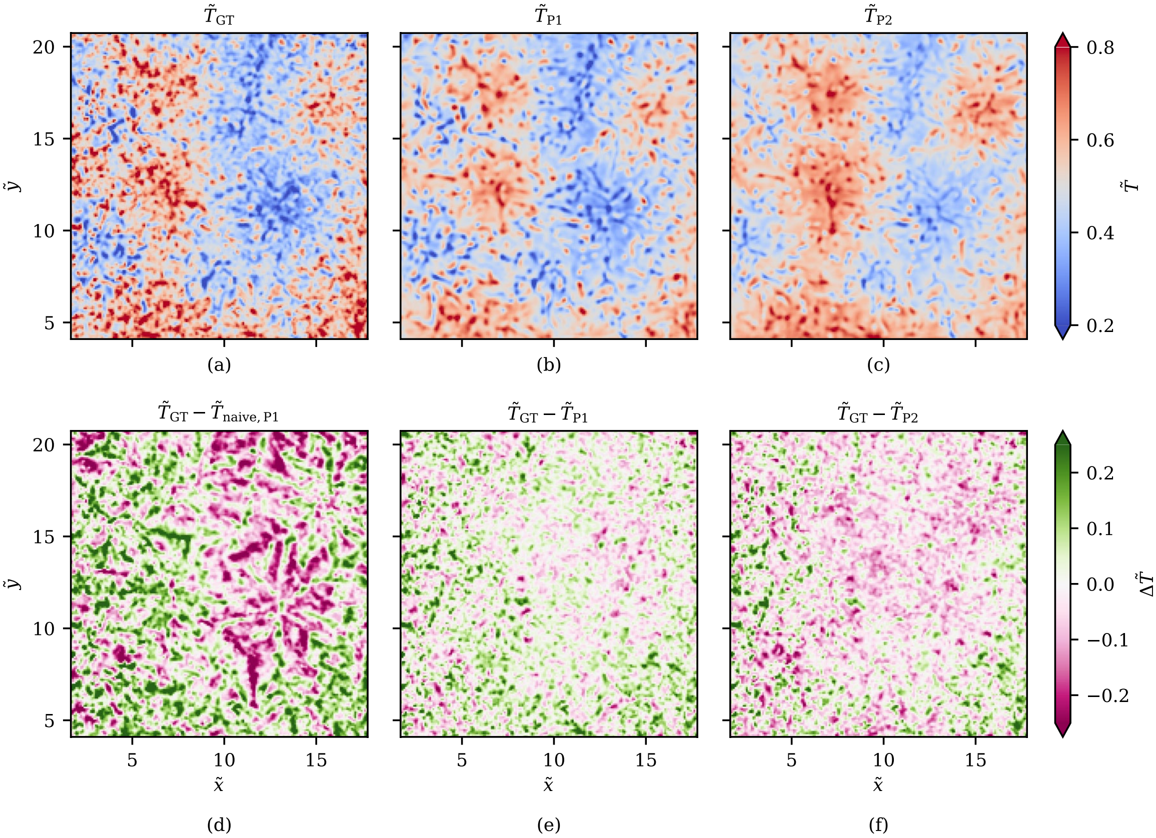

Fig. 13 Comparison of exemplary measured and predicted temperature fields $\tilde{T}_{\mathrm{GT}}$ (**a**), $\tilde{T}_{\mathrm{P1}}$ (**b**), $\tilde{T}_{\mathrm{P2}}$ (**c**) and the difference fields $\tilde{T}_{GT} - \tilde{T}_{naive,P1}$ (**d**), $\tilde{T}_{\mathrm{GT}} - \tilde{T}_{\mathrm{P1}}$ (**e**) and $\tilde{T}_{\mathrm{GT}} - \tilde{T}_{\mathrm{P2}}$ (**f**) at $\mathrm{Ra} = 4 \times 10^5$

Table 5 Overview average Pearson correlation coefficient C for various Ra and scenarios

	P1 (u-net)	P2 (u-net)	Naive
$\mathrm{Ra} = 2 \times 10^5$	0.78	0.72	0.52
$\mathrm{Ra} = 4 \times 10^5$	0.77	0.76	0.46
$\mathrm{Ra} = 7 \times 10^5$	0.74	0.70	0.37

$$C(\tilde{T}_{\mathrm{GT}}, \tilde{T}_{\mathrm{P}}) = \frac{\mathrm{cov}(\tilde{T}_{\mathrm{GT}}, \tilde{T}_{\mathrm{P}})}{\sigma_{\tilde{T}_{\mathrm{GT}}} \sigma_{\tilde{T}_{\mathrm{P}}}}, \tag{8}$$

where $\mathrm{cov}(\tilde{T}_{\mathrm{GT}}, \tilde{T}_{\mathrm{P}})$ is the covariance between ground truth and prediction and $\sigma_{\tilde{T}_{\mathrm{GT}}}$ and $\sigma_{\tilde{T}_{\mathrm{P}}}$ are their standard deviations. The resulting mean correlation coefficient C for all Ra and scenarios is shown in Table 5.

Looking at the table, we observe a high degree of correlation for the u-net predictions for all cases with a correlation coefficient C of 0.7 or higher. The naive approach for both scenarios results in the same correlation coefficient, which is at least 0.2 lower compared to the u-net prediction and decreases with Ra. This shows that temperature $\tilde{T}$ and vertical velocity $\tilde{u}_z$ are not highly correlated, especially at high Ra and, hence, the simple re-scaling approach of the vertical velocity field is unsuitable. Similar to the MAE, we see that for the P2 scenario, the u-net performs best for $\mathrm{Ra} = 4 \times 10^5$. In the next step, we computed the probability density functions (PDFs) of the temperatures $\tilde{T}$ from all snapshots in the test data set to understand further the deviation between measured and predicted temperature. Contrasting the PDFs shown in Fig. 14, we observe a good agreement for the lowest Rayleigh number $\mathrm{Ra} = 2 \times 10^5$ especially for the P1 case, albeit the PDFs of the predicted temperature show a lower probability of extreme temperature events and specifically high temperatures, which is in line with the smooth appearance of the predicted temperature fields.

This trend continues and increases with the Rayleigh number. For $\mathrm{Ra} = 4 \times 10^5$, we furthermore observe that the PDFs of the predictions for $\tilde{T} < 0.5$ no longer collapse on each other. While the PDF of the P1 u-net predictions agrees rather well with the PDF of the measurement data, the P2 u-net underestimates the probability of $\tilde{T} < 0.4$ events and overestimates the probability of $\tilde{T} \approx 0.6$ events. This coincides with the bias we observed in the temperature difference

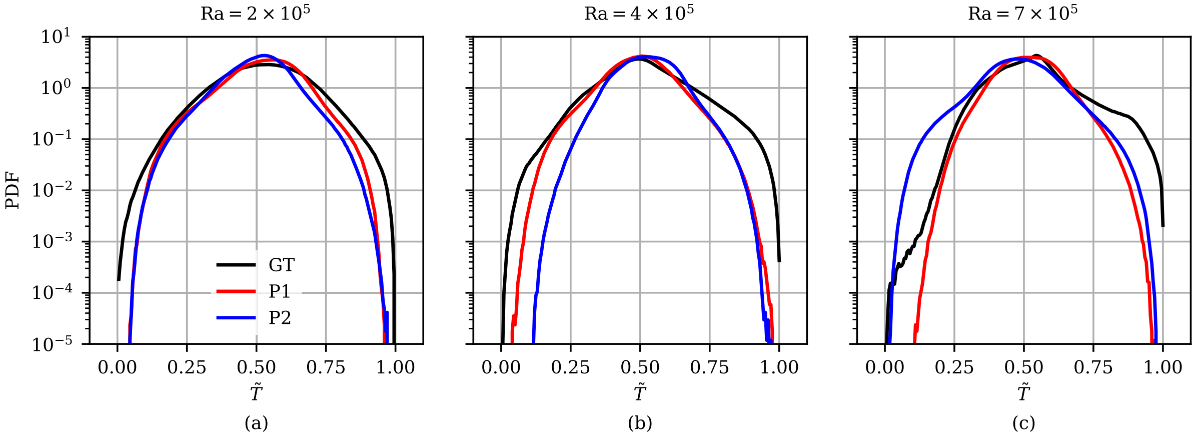

Fig. 14 PDFs of the temperature $\tilde{T}$ for various Ra

field $\tilde{T}_{\mathrm{GT}} - \tilde{T}_{\mathrm{P2}}$. For the highest Ra most dominantly but also for the lower Ra, we observe that the statistics of the temperature PDFs are not symmetric. This is mostly an effect of the asymmetric boundary conditions of the experiment, which resembles almost perfect isothermal boundary conditions at the bottom plate in contrast to the cooling plate made from glass where the conductive heat transfer within the plates is comparable to the convective heat transfer at the surface of the plate. This results in an increased probability of inverted heat transfer and limits the global heat transfer of the system. Further details can be found in Käufer et al. (2023). Nevertheless, we see a PDF of the P1 predictions and the measured temperature in the range $0.2 < \tilde{T} < 0.65$ almost match. Unlike the P1 predictions, the PDF of the P2 predictions shows a significant overestimation of cold temperatures. This is expected since the probability density function of the test and training cases differs significantly.

As the final step of our analysis of the predicted temperature data, we compare the azimuthally averaged power spectra of the temperature. These spectra indicate how the temperature is distributed over the different spatial scales or wavelengths. They are also commonly used to determine the size of the turbulent superstructures (Pandey et al. 2018; Moller et al. 2022). To determine the azimuthally averaged spectra, we compute the temperature field's two-dimensional discrete Fourier transform (DFT). We then compute the power spectrum and average the data azimuthally. We apply a spectral filter where we discard wavelengths that are larger than the field of view and therefore have no physical meaning. Additionally, we zero-pad the temperature fields to increase the number of spectral bins. Further details on the procedure can be found elsewhere (Moller 2022). The computed dimensionless wavelength $\tilde{\lambda}$ is normalized by H.

Figure 15 shows the power spectra calculated from a single exemplary instantaneous temperature snapshot (black, red, and blue) and the corresponding vertical velocity snapshot (green). Looking at the spectra, we observe excellent agreement between the spectra of the measured temperature data and their respective predicted counterparts in all cases, especially for wavelength $\tilde{\lambda} > 0.5$. For smaller scales, we see an increased power spectral density $E(\tilde{\lambda})$ for the measurement data as a consequence of the smoother appearance of the predictions. Of high interest in large aspect ratio, RBC are the turbulent superstructures whose size is commonly determined by the maximum of the power spectrum. Thus, a magnification of the peak region is shown in the bottom right of each plot. The dashed, gray line indicates the wavelength $\tilde{\lambda}$ corresponding to the peak in the power spectrum, which is additionally written in the inset. For Ra $= 4 \times 10^5$ and Ra $= 7 \times 10^5$, we observe a good agreement between the measured and predicted temperatures, especially for Ra $= 7 \times 10^5$ where all peaks virtually collapse. In contrast, for Ra $= 2 \times 10^5$, the peak of the P2 prediction shows a reduced power spectral density compared to the peak obtained from the measurement data. Nevertheless, the shapes of the peaks still agree well.

Beyond that and even most importantly, the plots show that for all cases, the wavelength $\tilde{\lambda}$ belonging to the peak is the same for the predictions and the measurements, which underlines that the u-net in both scenarios is a suitable tool capable of correctly predicting the size of the temperature superstructures.

Comparing the temperature spectra with the vertical velocity spectra, we can see a similar trend; however, the vertical velocity spectra are offset from the temperature and have a less pronounced peak at a large wavelength, which indicates the superstructure size. Especially for Ra $= 7 \times 10^5$, we see a significant difference between the temperature and vertical velocity spectra since the spectral peak of the vertical velocity is shifted toward a smaller wavelength. This indicates that the u-net learns to transfer

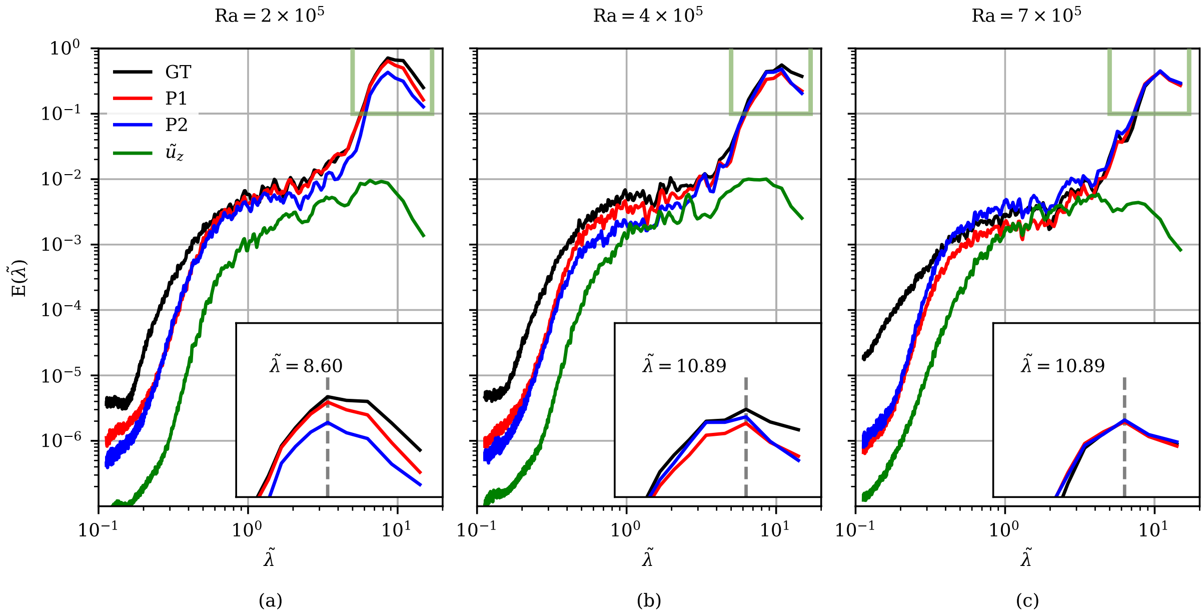

Fig. 15 Power spectra of instantaneous temperature (black, red, and blue) and vertical velocity fields (green) for various Ra

spatial scales rather than just augmenting the vertical velocity field.

So far, we have investigated the performance of the u-net models of both scenarios P1 and P2 by means of fields, probability density functions and power spectra of the temperature. We saw that for both scenarios P1 and P2, the u-net is able to accurately predict the temperature and, specifically, the large-scale structures in the temperature field. Overall the predicted temperature data are smoother, containing fewer extreme temperature events. This aligns with our expectations since the smallest-scale structures are lost in the convolutional layer of the contraction branch of the u-net despite the skip connections. On the one hand, this means the u-net does not reproduce the smallest-scale features. On the other hand, it makes it robust against measurement noise and acts like a filter. In fact, both autoencoders and u-nets were also successfully used for denoising tasks in the past (Vincent et al. 2010; Bao et al. 2020).

6.2 Heat transfer prediction

An important quantity of any RBC setup is the system's heat transfer, which is described by the Nusselt number Nu. Since we obtain temperature and vertical velocity data at the same point in space and time, we are able to determine the local Nusselt number

$$\mathrm{Nu}_{\mathrm{loc}} = \sqrt{\mathrm{RaPr}}\,\tilde{u}_z(\tilde{\mathbf{x}},\tilde{t})\tilde{\Theta}(\tilde{\mathbf{x}},\tilde{t}). \tag{9}$$

Here, $\tilde{\Theta}$ denotes the temperature fluctuation, which we obtain by decomposing $\tilde{T}$ into the linear conduction profile $\tilde{T}_{\mathrm{lin}}$ and the fluctuations $\tilde{\Theta}$ according to $\tilde{T}(\tilde{\mathbf{x}},\tilde{t}) = \tilde{T}_{\mathrm{lin}}(\tilde{z}) + \tilde{\Theta}(\tilde{\mathbf{x}},\tilde{t})$ with $\tilde{T}_{\mathrm{lin}} = 1 - \tilde{z}$. Thus $\tilde{T}_{\mathrm{lin}} = 0.5$ at the horizontal mid-plane where the data were measured. Further details on the derivation of the $\mathrm{Nu}_{\mathrm{loc}}$ can be found in Käufer et al. (2023).

Figure 16 shows exemplary snapshots of the $\mathrm{Nu}_{\mathrm{loc}}$ field computed from $\tilde{u}_z$ and $\tilde{T}_{\mathrm{GT}}$ (a), $\tilde{u}_z$ and $\tilde{T}_{\mathrm{P1}}$ (b), and $\tilde{u}_z$ and $\tilde{T}_{\mathrm{P2}}$ (c), respectively.

Comparing all three fields, we can easily detect the same patterns, albeit the fields of the ground truth $\mathrm{Nu}_{\mathrm{loc}}$ number and $\mathrm{Nu}_{\mathrm{loc}}$ calculated from the P1 predictions are substantially more similar. Yet, most strikingly, is the fact that we observe negative $\mathrm{Nu}_{\mathrm{loc}}$ events in the u-net predictions independent of the training scenario. In general, high temperature and upward velocity, as well as low temperature and downward velocity, are strongly correlated since temperature-induced local changes in density drive the flow. Hence, the neural network could achieve low MAE values by predicting high temperatures where upward velocities occur and vice versa. In practice, we also observe events $\mathrm{Nu}_{\mathrm{loc}} < 0$ where heat is transferred from top to bottom, even though with a much lower probability. The occurrence of negative $\mathrm{Nu}_{\mathrm{loc}}$ events in the $\mathrm{Nu}_{\mathrm{loc}}$ field obtained from both predictions, furthermore, at the same locations as in the ground truth fields, proves that the u-net learns a representation of the flow that goes far beyond the simple correlation of velocity and temperature. To further investigate this intriguing insight, we turn our attention to the PDFs of $\mathrm{Nu}_{\mathrm{loc}}$ in Fig. 17.

We note that PDFs agree well, especially in the range $0 < \mathrm{Nu}_{\mathrm{loc}} < 50$ for all Ra. With the exception of the P2 case for $\mathrm{Ra} = 7 \times 10^5$, $\mathrm{Nu}_{\mathrm{loc}} > 50$ events seem to be slightly underrepresented in the $\mathrm{Nu}_{\mathrm{loc}}$ calculated from predicted temperature data. Again this can be linked to the smoother

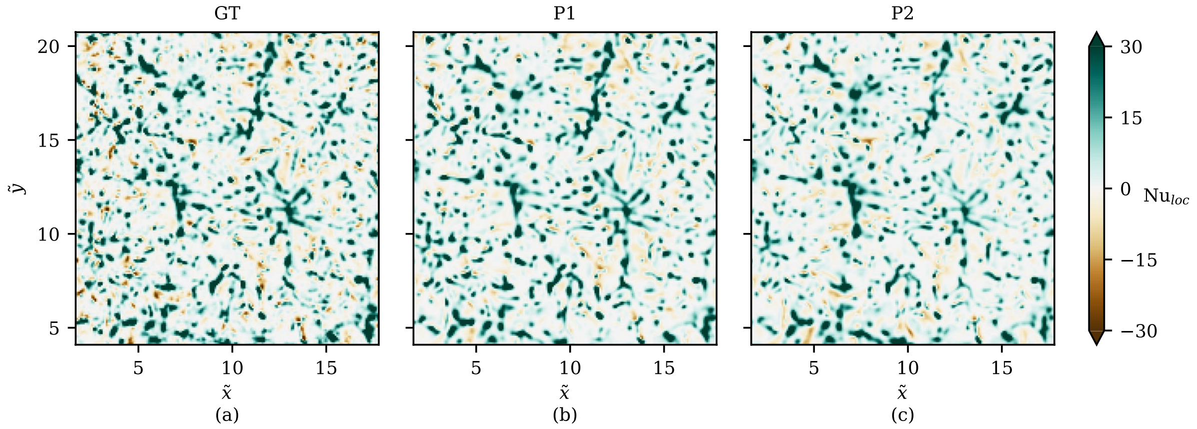

Fig. 16 Comparison of exemplary local Nusselt number fields at Ra = 4×10^5 calculated from purely measurement data (a) and from measured velocity and predicted temperature (b, c)

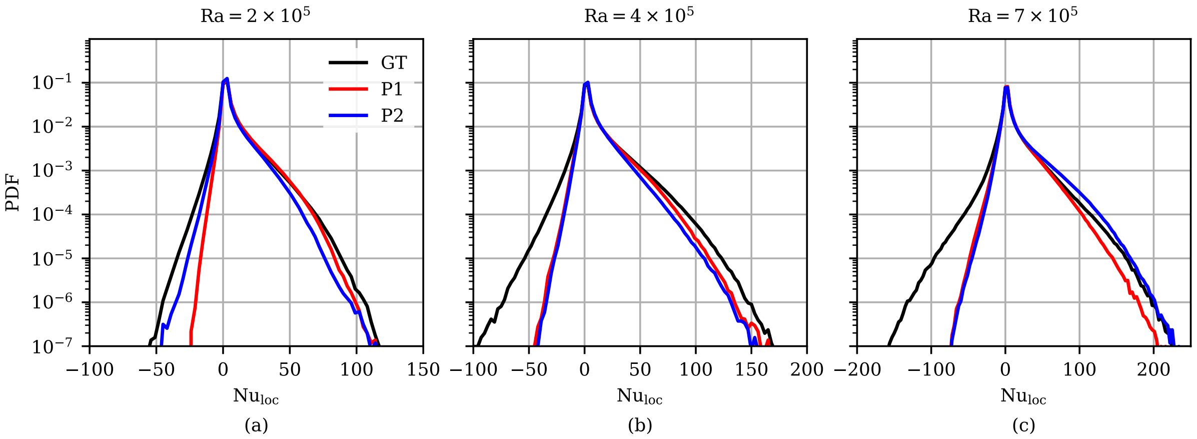

Fig. 17 PDFs of the local Nusselt number Nu_{loc} for various Ra

temperature field obtained from the predictions. Furthermore, the absolute probability of these events is low.

Looking at the probability for $Nu_{loc} < 0$, we see that the data obtained from the u-net predictions correctly represents the probability of low-intensity reversed heat transfer events, but the negative far-tail events are underrepresented. As mentioned above, those events are the toughest to predict due to the preferential correlation between temperature and velocity. Considering the low value of the absolute probability of these events, those might be altered by the measurement uncertainty. An exception is the PDFs obtained from the P2 data at Ra = 2×10^5. Solely based on the PDFs, for this Ra, the P2 model seems to outperform the P1 model since it better matches the PDF of the measured ground truth data, even for extreme $Nu_{loc} < 0$ events. At first glance, this seems surprising, but recalling that the P2 model for this Ra was trained only with data of higher Ra, which have a broader distribution of $Nu_{loc} < 0$ events, it is unsurprising that these are embedded into the trained model.

Table 6 Overview of the global Nusselt Nu for all combinations of Ra and temperature determination methods

	GT	P1	P2
Ra = 2×10^5	3.62	4.13	2.96
Ra = 4×10^5	5.10	5.12	4.41
Ra = 7×10^5	5.38	5.80	7.94

Lastly, we want to quantify and compare the global heat transfer characterized by the global Nusselt number $Nu = \langle Nu_{loc} \rangle_{\tilde{A},\tilde{t}}$ which we obtain by averaging Nu_{loc} over the field of view $\tilde{A}$ and time $\tilde{t}$.

Looking at the resulting values in Table 6, we observe that the ground truth Nu and the Nu calculated from the P1 agree well with relative differences, which are defined as the absolute value of the difference between reconstructed and measured global Nusselt number divided by the measured global Nusselt number, between 14.1% and 0.3%. As a rule of thumb, Nu obtained from the P1 predictions is slightly higher. In contrast, the deviations for Nusselt numbers obtained from the P2 predictions—which undoubtedly are the more difficult predictions—are significantly larger with relative values between 47.6% and 13.5%. Contributing to this large deviation is that in the experimental setup, a change in Ra also affects the thermal boundary conditions, additionally influencing the flow physics. Nevertheless, we see qualitative agreement. In both cases, the relative deviation is the lowest for $\mathrm{Ra} = 4 \times 10^5$ which is reasonable in the P2 case but unexpected for the P1 case where the MAE of the temperature prediction is lowest for $\mathrm{Ra} = 7 \times 10^5$.

To summarize, we have shown that the u-net predictions allow us to compute physically plausible $\mathrm{Nu}_{\mathrm{loc}}$ fields, albeit underestimating the probability of extreme $\mathrm{Nu}_{\mathrm{loc}} < 0$ events. Comparing the global values of Nu, we observed good quantitative agreement between ground truth and the P1 case but only qualitative agreement for the P2 case.

7 Conclusion and perspectives

In this paper, we proposed a purely data-driven, supervised, deep learning-based method to derive the distribution of a scalar quantity from experimental velocity data. We demonstrate it for the example of large aspect ratio RBC and extract temperature data from stereoscopic PIV velocity data. We used data from experiments at $\mathrm{Ra} = 2 \times 10^5, 4 \times 10^5, 7 \times 10^5$ and chose the u-net architecture as a baseline. Starting from this point, we investigated the influence of several modifications, namely the choice of the activation function, the up-sampling and the batch normalization. We observed that only batch normalization had a positive effect on our task. Thereupon, we identified the optimal values for the channel factor $\rho = 48$ and the learning rate $\eta = 0.0005$ in an extensive grid search. We observed that u-net exhibits stable training behavior, yielding only minor deviation in performance when trained with different random initial weights. The chosen parameter combination shows a low standard deviation of the validation MSE and the grid search heat map a clear convergence, indicating the robustness of the parameter selection. Subsequently, we defined two real-world application scenarios, P1 and P2. In scenario P1, we trained the u-net on data of the same experimental run. We studied the influence of the amount of training data on the MAE and selected the number of three training bursts as the best compromise between model performance and training data generation effort. Furthermore, we proved that the predictions are independent of the temporal distance between training and prediction. The MAE, which is like the MSE a common loss metric in the field of machine learning, for all models in scenario P1 is below 0.065. For scenario P2, we trained the u-net on the data of two Ra and predicted the temperature for the remaining third Ra. In this scenario, all models achieve MAE values below 0.085. We demonstrated that the u-net prediction in any scenario significantly outperforms the naive assumption of $\tilde{T}_{\mathrm{naive}}$.

We rigorously analyzed the performance of the models in both scenarios by comparing the temperature fields, PDFs, and power spectra with the ground truth data. We observed that the characteristic superstructures are clearly recognizable in the predictions of both scenarios, albeit smoothed. The signs of smoother predictions are also remnant in the PDFs and the power spectra, which show a lower probability for extreme temperature events and lower power spectral density on smaller scales, respectively. Even though the P2 predictions tend to be biased, the size of the temperature superstructure can be correctly determined from the spectra in all cases. Our comparison of the heat transfer associated with the temperature predictions unveiled similarities in the field of ground truth $\mathrm{Nu}_{\mathrm{loc}}$ field and the $\mathrm{Nu}_{\mathrm{loc}}$ fields obtained from the predicted temperature. Remarkably, the $\mathrm{Nu}_{\mathrm{loc}}$ fields obtained from the u-net predictions feature the occurrence and location of $\mathrm{Nu}_{\mathrm{loc}} < 0$ correctly, especially for the P1 scenario. The PDFs of $\mathrm{Nu}_{\mathrm{loc}}$ show a good agreement with ground truth data for $\mathrm{Nu}_{\mathrm{loc}} > 0$. However, extreme $\mathrm{Nu}_{\mathrm{loc}} < 0$ events are underrepresented. The comparison of the global Nu displayed quantitative agreement between measurement and P1 scenario data but only qualitative agreement for the P2 scenario.

Our study showed that the u-net has proven to be a suitable and robust tool. When trained on data of the same experimental run, it is capable of physically consistent predictions from noisy measurement data. In the future, we want to incorporate information about the heat transfer into the training of the u-net. Therefore, we want to add an additional loss term determined from the difference between the PDF of $\mathrm{Nu}_{\mathrm{loc}}$ obtained from measured temperature and vertical velocity and the PDF of $\mathrm{Nu}_{\mathrm{loc}}$ obtained from predicted temperature and measured vertical velocity. Thereby, we expect the model to better estimate the negative tails of the $\mathrm{Nu}_{\mathrm{loc}}$ PDFs.

Appendix 1 Activation functions

See Fig. 18.

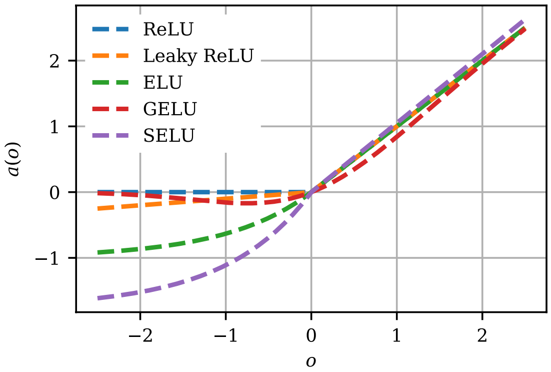

Fig. 18 An overview over different activation functions a: ReLU, Leaky ReLU ($\alpha = 0.01$), ELU ($\alpha = 1$), GELU ($\mu = 0$, $\sigma = 1$) and SELU ($\alpha = \alpha_{01}$, $\lambda = \lambda_{01}$) for layer outputs o

$$\mathrm{ReLU}(o) = \begin{cases} o & \text{if } o > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

$$\mathrm{LeakyReLU}(o) = \begin{cases} o & \text{if } o > 0, \\ \alpha o & \text{otherwise,} \end{cases} \tag{11}$$

$$\mathrm{ELU}(o) = \begin{cases} o & \text{if } o > 0, \\ a(e^{o} - 1) & \text{otherwise,} \end{cases} \tag{12}$$

$$\mathrm{GELU}(o) = o\Phi(o), \tag{13}$$

$$\mathrm{selu}(o) = \lambda \begin{cases} o & \text{if } o > 0, \\ \alpha(e^{o} - 1) & \text{otherwise.} \end{cases} \tag{14}$$

Appendix 2 Experimental data

See Fig. 19.

Appendix 3 Hyper-parameter study

See Fig. 20.

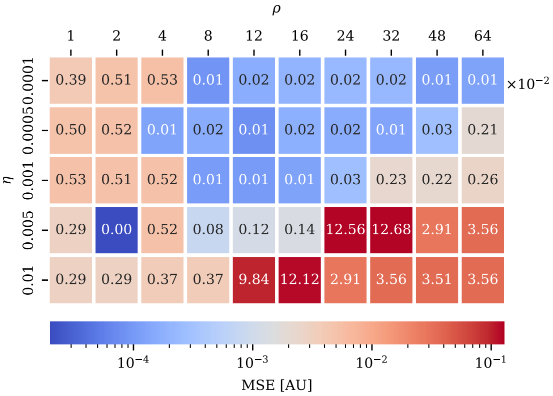

Fig. 20 Standard deviation of the validation MSEs for all tested η-ρ -combinations as heat map

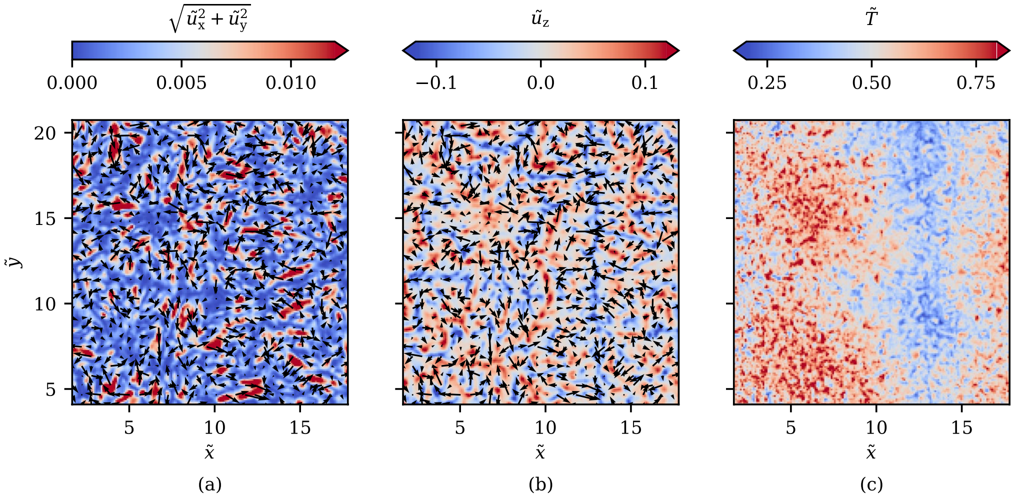

Fig. 19 The input (**a**, **b**) and target (**c**) of the neural network. Exemplary snapshots at $\mathrm{Ra} = 7 \times 10^{5}$

Acknowledgements The authors gratefully acknowledge Sebastian Moller for providing the experimental data.

Author contributions P. T. developed the deep learning model and conducted all related experiments. T. K. carried out the fluid mechanical analysis. P. T. and T. K. wrote the manuscript and designed the deep learning strategy and were supervised by P. M. and C. C. All authors reviewed the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was supported by the Carl Zeiss Foundation within project no. P2018-02-001 "Deep Turb - Deep Learning in and of Turbulence." The work of T. K. was partially funded by the DFG Priority Programme SPP 1881 on "Turbulent Superstructures" under project no. 429328691.

Data availability The experimental data can be made available by the authors on request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

References

Ahlers G, Grossmann S, Lohse D (2009) Heat transfer and large scale dynamics in turbulent Rayleigh–Bénard convection. Rev Mod Phys 81:503–537

Ahlers G, Bodenschatz E, Hartmann R, He X, Lohse D, Reiter P, Stevens RJ, Verzicco R, Wedi M, Weiss S et al (2022) Aspect ratio dependence of heat transfer in a cylindrical Rayleigh–Bénard cell. Phys Rev Lett 128:084501

Alom MZ, Yakopcic C, Hasan M, Taha TM, Asari VK (2019) Recurrent residual U-Net for medical image segmentation. J Med Imaging 6:014006

Bao L, Yang Z, Wang S, Bai D, Lee J (2020) Real image denoising based on multi-scale residual dense block and cascaded U-Net with block-connection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp 448–449

Bessaih R, Kadja M (2000) Turbulent natural convection cooling of electronic components mounted on a vertical channel. Appl Therm Eng 20(2):141–154

Bjorck N, Gomes CP, Selman B, Weinberger KQ (2018) Understanding batch normalization. Adv Neural Inf Process Syst 31

Brunton SL, Kutz JN (2022) Data-driven science and engineering: machine learning, dynamical systems, and control. Cambridge University Press, Cambridge

Brunton SL, Noack BR, Koumoutsakos P (2020) Machine learning for fluid mechanics. Annu Rev Fluid Mech 52:477–508

Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE (2021) Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mech 37:1727–1738

Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis GE (2021) Physics-informed neural networks for heat transfer problems. J Heat Transfer 143:060801

Chillà F, Schumacher J (2012) New perspectives in turbulent Rayleigh–Bénard convection. Eur Phys J E 35:58

Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O (2016) 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 424–432

Cierpka C, Kästner C, Resagk C, Schumacher J (2019) On the challenges for reliable measurements of convection in large aspect ratio Rayleigh–Bénard cells in air and sulfur-hexafluoride. Exp Therm Fluid Sci 109:109841

Clevert D-A, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (ELUs). ArXiv preprint. arXiv:1511.07289

Dabiri D (2009) Digital particle image thermometry/velocimetry: a review. Exp Fluids 46:191–241

Esmaeilzadeh S, Azizzadenesheli K, Kashinath K, Mustafa M, Tchelepi HA, Marcus P, Prabhat M, Anandkumar A et al (2020) Meshfreeflownet: a physics-constrained deep continuous space-time super-resolution framework. In: SC20: International conference for high performance computing, networking, storage and analysis. IEEE, pp 1–15

Fonda E, Pandey A, Schumacher J, Sreenivasan KR (2019) Deep learning in turbulent convection networks. Proc Natl Acad Sci 116:8667–8672

Fukami K, Fukagata K, Taira K (2021) Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. J Fluid Mech 909:9

Gao H, Sun L, Wang J-X (2021) Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. Phys Fluids 33:073603

Ghazijahani MS, Heyder F, Schumacher J, Cierpka C (2022) On the benefits and limitations of echo state networks for turbulent flow prediction. Meas Sci Technol 34:014002

Guervilly C, Cardin P, Schaeffer N (2019) Turbulent convective length scale in planetary cores. Nature 570:368–371

Hendrycks D, Gimpel K (2016) Gaussian error linear units (Gelus). ArXiv preprint. arXiv:1606.08415

Heyder F, Schumacher J (2021) Echo state network for two-dimensional turbulent moist Rayleigh–Bénard convection. Phys Rev E 103:053107

Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. ArXiv preprint. arXiv:1207.0580

Huang H, Lin L, Tong R, Hu H, Zhang Q, Iwamoto Y, Han X, Chen Y-W, Wu J (2020) UNet 3+: a full-scale connected UNet for medical image segmentation. In: ICASSP 2020-2020 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1055–1059

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR, pp 448–456

Jansson A, Humphrey E, Montecchio N, Bittner R, Kumar A, Weyde T (2017) Singing voice separation with deep U-net convolutional networks

Kähler CJ, Astarita T, Vlachos PP, Sakakibara J, Hain R, Discetti S, La Foy R, Cierpka C (2016) Main results of the 4th international PIV challenge. Exp Fluids 57:1–71

Kashanj S, Nobes DS (2023) Application of 4D two-colour LIF to explore the temperature field of laterally confined turbulent Rayleigh–Bénard convection. Exp Fluids 64:46

Käufer T, Vieweg PP, Schumacher J, Cierpka C (2023) Thermal boundary condition studies in large aspect ratio Rayleigh–bénard convection. ArXiv preprint. arXiv:2302.13738

Klambauer G, Unterthiner T, Mayr A, Hochreiter S (2017) Self-normalizing neural networks. Adv Neural Inf Process Syst 30

König J, Chen M, Rösing W, Boho D, Mäder P, Cierpka C (2020) On the use of a cascaded convolutional neural network for three-dimensional flow measurements using astigmatic PTV. Meas Sci Technol 31:074015

Krueger D, Maharaj T, Kramár J, Pezeshki M, Ballas N, Ke NR, Goyal A, Bengio Y, Courville A, Pal C (2016) Zoneout: regularizing RNNs by randomly preserving hidden activations. ArXiv preprint. arXiv:1606.01305

LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551

Ling J, Kurzawski A, Templeton J (2016) Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J Fluid Mech 807:155–166

Liu B, Tang J, Huang H, Lu X-Y (2020) Deep learning methods for super-resolution reconstruction of turbulent flows. Phys Fluids 32(2):025105

Mapes BE, Houze RA Jr (1993) Cloud clusters and superclusters over the oceanic warm pool. Mon Weather Rev 121:1398–1416

Marshall J, Schott F (1999) Open-ocean convection: observations, theory, and models. Rev Geophys 37:1–64

Massing J, Kaden D, Kähler C, Cierpka C (2016) Luminescent two-color tracer particles for simultaneous velocity and temperature measurements in microfluidics. Meas Sci Technol 27(11):115301

Mendez MA (2022) Linear and nonlinear dimensionality reduction from fluid mechanics to machine learning. Measurement Science and Technology

Mendez MA, Ianiro A, Noack BR, Brunton SL (2023) Data-driven fluid mechanics: combining first principles and machine learning. Cambridge University Press, Cambridge

Moller S (2022) Experimental characterization of turbulent superstructures in large aspect ratio Rayleigh–Bénard convection. Dissertation, TU Ilmenau

Moller S, König J, Resagk C, Cierpka C (2019) Influence of the illumination spectrum and observation angle on temperature measurements using thermochromic liquid crystals. Meas Sci Technol 30:084006

Moller S, Resagk C, Cierpka C (2020) On the application of neural networks for temperature field measurements using thermochromic liquid crystals. Exp Fluids 61:1–21

Moller S, Resagk C, Cierpka C (2021) Long-time experimental investigation of turbulent superstructures in Rayleigh–Bénard convection by noninvasive simultaneous measurements of temperature and velocity fields. Exp Fluids 62:1–18

Moller S, Käufer T, Pandey A, Schumacher J, Cierpka C (2022) Combined particle image velocimetry and thermometry of turbulent superstructures in thermal convection. J Fluid Mech 945:A22

Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: ICML

Oktay O, Schlemper J, Folgoc LL, Lee M, Heinrich M, Misawa K, Mori K, McDonagh S, Hammerla NY, Kainz B et al (2018) Attention U-Net: learning where to look for the pancreas. ArXiv preprint. arXiv:1804.03999

Otto H, Naumann C, Odenthal C, Cierpka C (2023) Unsteady inherent convective mixing in thermal-energy-storage systems during standby periods. PRX Energy 2(4):043001

Pandey A, Scheel JD, Schumacher J (2018) Turbulent superstructures in Rayleigh–Bénard convection. Nat Commun 9:2118

Pandey S, Schumacher J, Sreenivasan KR (2020) A perspective on machine learning in turbulent flows. J Turbul 21:567–584

Pandey S, Teutsch P, Mäder P, Schumacher J (2022) Direct data-driven forecast of local turbulent heat flux in Rayleigh–Bénard convection. Phys Fluids 34:045106

Prasad AK (2000) Stereoscopic particle image velocimetry. Exp Fluids 29:103–116

Prechelt L (2012) Early stopping—But when?. In: Neural networks: tricks of the trade, 2nd edn., pp 53–67

Rabault J, Kolaas J, Jensen A (2017) Performing particle image velocimetry using artificial neural networks: a proof-of-concept. Meas Sci Technol 28:125301

Raffel M, Willert CE, Scarano F, Kähler CJ, Wereley ST, Kompenhans J (2018) Particle image velocimetry. Springer International Publishing, Berlin

Raissi M, Wang Z, Triantafyllou MS, Karniadakis GE (2019) Deep learning of vortex-induced vibrations. J Fluid Mech 861:119–137

Raissi M, Yazdani A, Karniadakis GE (2020) Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. Science 367:1026–1030

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 234–241

Sachs S, Ratz M, Mäder P, König J, Cierpka C (2023) Particle detection and size recognition based on defocused particle images: a comparison of a deterministic algorithm and a deep neural network. Exp Fluids 64:21

Sakakibara J, Adrian RJ (1999) Whole field measurement of temperature in water using two-color laser induced fluorescence. Exp Fluids 26(1–2):7–15

Santurkar S, Tsipras D, Ilyas A, Madry A (2018) How does batch normalization help optimization? Adv Neural Inf Process Syst 31

Schiepel D, Schmeling D, Wagner C (2021) Simultaneous tomographic particle image velocimetry and thermometry of turbulent Rayleigh–Bénard convection. Meas Sci Technol 32:095201

Schonfeld E, Schiele B, Khoreva A (2020) A U-net based discriminator for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8207–8216

Schumacher J, Sreenivasan KR (2020) Colloquium: unusual dynamics of convection in the sun. Rev Mod Phys 92:041001

Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1874–1883

Shishkina O (2021) Rayleigh–Bénard convection: the container shape matters. Phys Rev Fluids 6:090502

Siddique N, Paheding S, Elkin CP, Devabhaktuni V (2021) U-Net and its variants for medical image segmentation: a review of theory and applications. IEEE Access 9:82031–82057

Stevens Richard JAM, Blass A, Zhu X, Verzicco R, Lohse D (2018) Turbulent thermal superstructures in Rayleigh–Bénard convection. Phys Rev Fluids 3:041501

Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A, Bottou L (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408

Wang C, Zhao Z, Ren Q, Xu Y, Yu Y (2019) Dense U-net based on patch-based learning for retinal vessel segmentation. Entropy 21:168

Wang Z, Chen J, Hoi SC (2020) Deep learning for image super-resolution: a survey. IEEE Trans Pattern Anal Mach Intell 43:3365–3387

Wang R, Kashinath K, Mustafa M, Albert A, Yu R (2020) Towards physics-informed deep learning for turbulent flow prediction. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1457–1466

Xu J, Li Z, Du B, Zhang M, Liu J (2020) Reluplex made more practical: Leaky ReLU. In: 2020 IEEE Symposium on computers and communications (ISCC). IEEE, pp 1–7

Yu C, Bi X, Fan Y (2023) Deep learning for fluid velocity field estimation: a review. Ocean Eng 271:113693

Zhang Z, Liu Q, Wang Y (2018) Road extraction by deep residual u-net. IEEE Geosci Remote Sens Lett 15(5):749–753

Zhang Z, Wu C, Coleman S, Kerr D (2020) DENSE-INception U-net for medical image segmentation. Comput Methods Progr Biomed 192:105395

Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J (2019) Unet++: redesigning skip connections to exploit multiscale features in image segmentation. IEEE Trans Med Imaging 39:1856–1867

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.