Machine Learning Applications in Convective Turbulence

R. Kräuter, D. Krasnov, A. Pandey, C. Schneide,
K. Padberg-Gehle, D. Giannakis, K. R. Sreenivasan, J.
Schumacher

published in

NIC Symposium 2020
M. Müller, K. Binder, A. Trautmann (Editors)

Forschungszentrum Jülich GmbH,
John von Neumann Institute for Computing (NIC),
Schriften des Forschungszentrums Jülich, NIC Series, Vol. 50,
ISBN 978-3-95806-443-0, pp. 357.
http://hdl.handle.net/2128/24435

Machine Learning Applications in Convective Turbulence

Robert Kräuter[1], Dmitry Krasnov[1], Ambrish Pandey[1], Christiane Schneide[2], Kathrin Padberg-Gehle[2], Dimitrios Giannakis[3], Katepalli R. Sreenivasan[3,4], and Jörg Schumacher[1,4]

[1] Institut für Thermo- und Fluiddynamik,
Technische Universität Ilmenau, 98684 Ilmenau, Germany
E-mail: {ambrish.pandey, robert.kraeuter, dmitry.krasnov, joerg.schumacher}@tu-ilmenau.de

[2] Institut für Mathematik und ihre Didaktik,
Leuphana Universität Lüneburg, 21335 Lüneburg, Germany
E-mail: {christiane.schneide, kathrin.padberg-gehle}@leuphana.de

[3] Courant Institute of Mathematical Sciences,
New York University, New York City, NY 10012, USA
E-mail: dimitris@cims.nyu.edu

[4] Tandon School of Engineering, New York University, New York City, NY 11201, USA
E-mail: katepalli.sreenivsan@nyu.edu

Turbulent convection flows are ubiquitous in natural systems such as in the atmosphere or in stellar interiors as well as in technological applications such as cooling or energy storage devices. Their physical complexity and vast number of degrees of freedom prevents often an access by direct numerical simulations that resolve all flow scales from the smallest to the largest plumes and vortices in the system and requires a simplified modelling of the flow itself and the resulting turbulent transport behaviour. The following article summarises some examples that aim at a reduction of the flow complexity and thus of the number of degrees of freedom of convective turbulence by machine learning approaches. We therefore apply unsupervised and supervised machine learning methods to direct numerical simulation data of a Rayleigh-Bénard convection flow which serves as a paradigm of the examples mentioned at the beginning.

1 Introduction

Machine learning algorithms,[5] particularly in the form of multilayered deep neural networks, have lately found various new applications in the Big Data domain that increasingly affects many fields of our daily life, *e. g.* by speech recognition tools in cellular phones. These machine learning methods are yet about to find an established place in the modelling and analysis of turbulent flows despite a few promising attempts.[2] The comprehensive data records needed for the algorithms come from full-scale supercomputer simulations of the equations of fluid motion or laboratory experiments applying optical measurement techniques. A proper assimilation and processing of this vast amount of data requires a change of paradigms in data processing, structure recognition and subgrid modelling.

Machine learning can be classified into two big categories, supervised and unsupervised machine learning. Supervision refers to a training of the algorithm with labelled input-output examples. In the following, we review briefly our own machine learning analyses with examples from both categories. The flow of interest is turbulent Rayleigh-Bénard convection (RBC) which serves as a paradigm to many turbulent flows in nature and technology that are driven and sustained by temperature differences.[1] On the one hand, we will apply unsupervised machine learning for the reconstruction of the large-scale circulation (LSC) in a cubical closed convection cell by means of data-based Koopman eigenfunctions.[4] Unsupervised machine learning is also applied to study clusters of Lagrangian particle trajectories in a large-aspect-ratio turbulent convection flow.[12] On the other hand, supervised machine learning algorithms in form of deep neural networks are taken to reduce the three-dimensional and fully resolved turbulent superstructures in the temperature field of a Rayleigh-Bénard flow,[9] to a two-dimensional planar temporal network[6] of maximal and minimal temperature ridges with point defects.[3]

In all examples, we solve the three-dimensional Boussinesq equations of turbulent convection by means of direct numerical simulations that resolve all relevant turbulence structures in the flow and provide the data base. They couple the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ with the temperature field $T(\boldsymbol{x}, t)$. The equations are made dimensionless by using the height of the layer or cell H, the free-fall velocity U_f and the imposed temperature difference ΔT between bottom and top. This implies a natural convective time unit, the free-fall time $T_f = H/U_f$. The Boussinesq system contains the three control parameters: the Rayleigh number Ra, the Prandtl number Pr and the aspect ratio $\Gamma = L/H$ with the cell length $L = L_x = L_y$ that is set to values of $\Gamma = 1$, 16 or 25 in this work. The equations are given by

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = -\boldsymbol{\nabla} p + \sqrt{\frac{Pr}{Ra}}\, \boldsymbol{\nabla}^2 \boldsymbol{u} + T \boldsymbol{e}_z \tag{2}$$

$$\frac{\partial T}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})T = \frac{1}{\sqrt{RaPr}} \boldsymbol{\nabla}^2 T \tag{3}$$

No-slip boundary conditions for the fluid are applied at all walls, *i. e.* $u_i = 0$. The side walls are thermally insulated, *i. e.* $\partial T/\partial n = 0$ with n being the normal direction. At the top and bottom a constant dimensionless temperature of $T = 0$ and 1 is maintained, respectively. The equations are numerically solved by the Nek5000 spectral element method package.[8, 11] In addition, N_p massless Lagrangian tracer particles are advanced corresponding to

$$\frac{d\boldsymbol{X}_j}{dt} = \boldsymbol{u}(\boldsymbol{X}_j, t) \tag{4}$$

with $j = 1 \ldots N_p$. The Lagrangian particles are advected by a 3-step explicit Adams-Bashforth scheme. The interpolation of the velocity field to the particle position is done spectrally. The turbulent convection flow can be considered as a nonlinear dynamical system in a high-dimensional phase space $\mathcal{M}$. The state vector of the RBC is given by $\phi_m(t) = (\boldsymbol{u}(\boldsymbol{x}_i, t), T(\boldsymbol{x}_i, t))$ where $\boldsymbol{x}_i$ are the coordinates of the grid points, $i = 1, \ldots, N$. With a total number of grid points of N it follows that $\dim(\mathcal{M}) = 4N$. The dynamical evolution of the turbulent flow describes a trajectory in $\mathcal{M}$ which is determined by $\phi_m(t_{k+1}) = F_m(\phi_n(t_k), t_k)$ where F_m is a nonlinear vector valued function with $m, n = 1, \ldots, 4N$. Here $k = 1, 2, \ldots, N_t$ with the number of time steps N_t.

2 Unsupervised machine learning applications

Unsupervised learning extracts features in (high-dimensional) data sets without preexisting labelled training data. These techniques are partly already well-established and well-known. They comprise for example clustering and dimensionality-reduction techniques, such as the well-known Dynamic Mode Decomposition.[15] We will describe two specific examples more detailed in the following.

Giannakis *et al.*[4] discuss the long-time evolution of a three-dimensional turbulent RBC flow in a closed cubic cell via the eigenfunctions of the linear Koopman operator $\mathcal{K}_t$ that defines a new dynamical system which rather governs the evolution of observables on $\mathcal{M}$ than the state vector $\phi_m(t)$ itself.[15] The space of observables and thus $\mathcal{K}_t$ are infinite-dimensional even though the associated nonlinear dynamical system is finite-dimensional. In practice, one seeks a finite-dimensional representation of $\mathcal{K}_t$; more detailed, a data-driven basis is calculated here to represent a regularised generator V of the unitary Koopman group $\mathcal{U}_t = \exp(tV)$ in the sense of a Galerkin approximation. This is exactly the point where unsupervised machine learning enters the scene. The goal is to find (or better to *geometrically learn*) a possibly complicated curved submanifold $\mathcal{N} \subset \mathcal{M}$ with $\dim(\mathcal{N}) \ll \dim(\mathcal{M})$ on which the few relevant degrees of freedom evolve that describe the large-scale flow of the RBC setting effectively. This is done by a diffusion process on the data snapshots, $\phi_i(t_k)$, that quantifies the local connectivity of two system states i, j

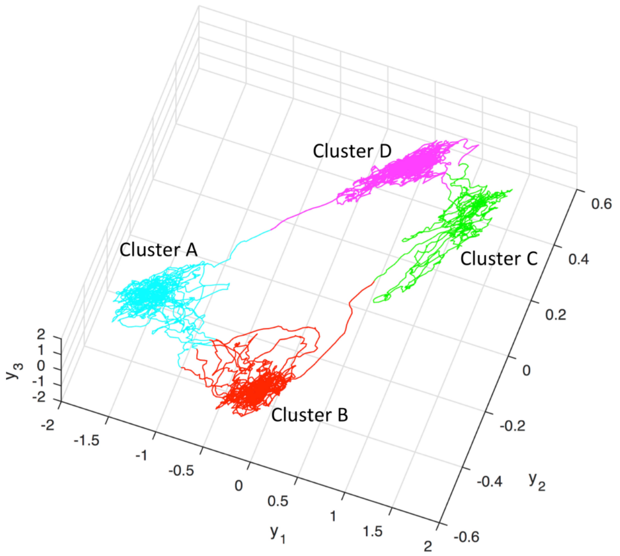

Figure 1. Dynamical trajectory of a turbulent RBC flow in a closed cubical cell at $\mathrm{Ra} = 10^7$. Shown is a three-dimensional subspace with coordinates which are constructed from the primary Koopman eigenfunctions. The trajectory is sampled over 10^4 convective free-fall time units T_f. The large-scale flow dynamics is found to cluster in four macro-states A, B, C, and D that correspond to large-scale circulation rolls in the box across the diagonals (two configurations times clockwise/counterclockwise spin). The figure is taken from Ref. 4.

and the resulting transition probability (or the Markov operator) with a Gaussian kernel which follows to

$$P_{ij} = \frac{1}{C} \exp\left(-\frac{1}{\epsilon Q} \sum_{q=0}^{Q-1} \|\boldsymbol{u}_{i-q} - \boldsymbol{u}_{j-q}\|_{L^2}^2 \right) \tag{5}$$

Here, C is a normalisation constant; we use the velocity field data only. The sum over q stands for an additional time delay-averaging. The eigenvectors of P_{ij} form the data-driven basis in which we can compute the Koopman eigenfunctions afterwards.

The resulting Koopman eigenfunctions can be grouped into subsets in accordance with the discrete symmetries in a cubic box. In particular, a projection of the velocity field onto the first group of eigenfunctions reveals the four stable large-scale circulation (LSC) states in the convection cell which are shown in Fig. 1. We recapture the preferential circulation

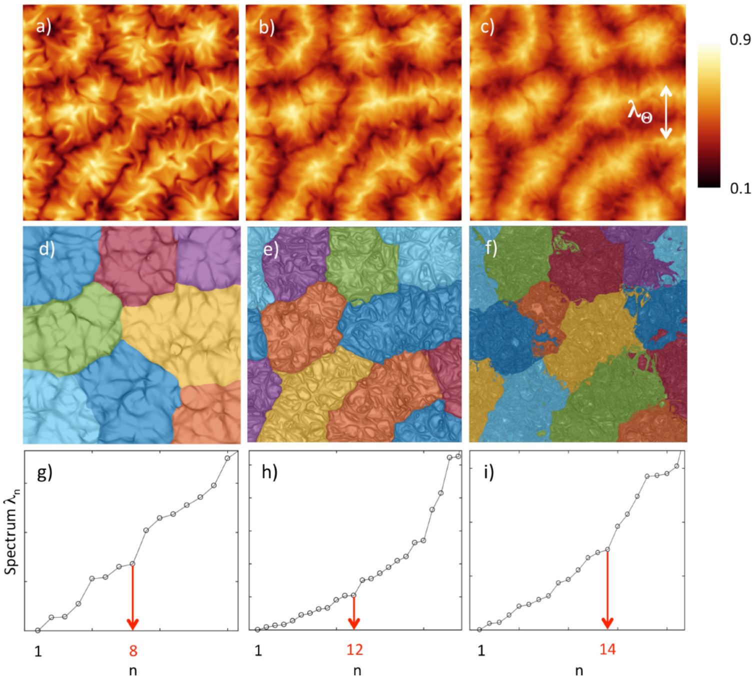

Figure 2. Lagrangian trajectory-based analysis of three-dimensional turbulent large-scale patterns in a RBC flow. We advected 512^2 individual Lagrangian tracers that started inside the thermal boundary layer at the bottom plate. (a–c) Time-averaged temperature field in the midplane. Progressing averaging times are 2.6, 10.4, and 30.1 free-fall convective time units, respectively. In panel (c), the characteristic pattern scale $\lambda_\Theta \approx 4H$ of the turbulent superstructures is indicated by a double arrow. (d–f) Lagrangian trajectory clusters obtained from the leading eigenvectors of the graph Laplacian matrix. Particles that belong to the same spectral cluster at the corresponding time are coloured equally. The background contours are the ridges of the maximum finite-time Lyapunov exponent. Ridges and clusters are indicated with respect to the initial Lagrangian particle position. (g–i) Corresponding eigenvalue spectra of the graph Laplacian matrix. The spectral gap between eigenvalues no. 8-9, 12-13, and 14-15 is used to detect k = 8, 12, and 14 trajectory clusters by a standard k-means clustering algorithm, respectively. The parameters are $(Ra, Pr, \Gamma) = (10^5, 0.7, 1)$ for this simulation. The figure is taken from Ref. 10.

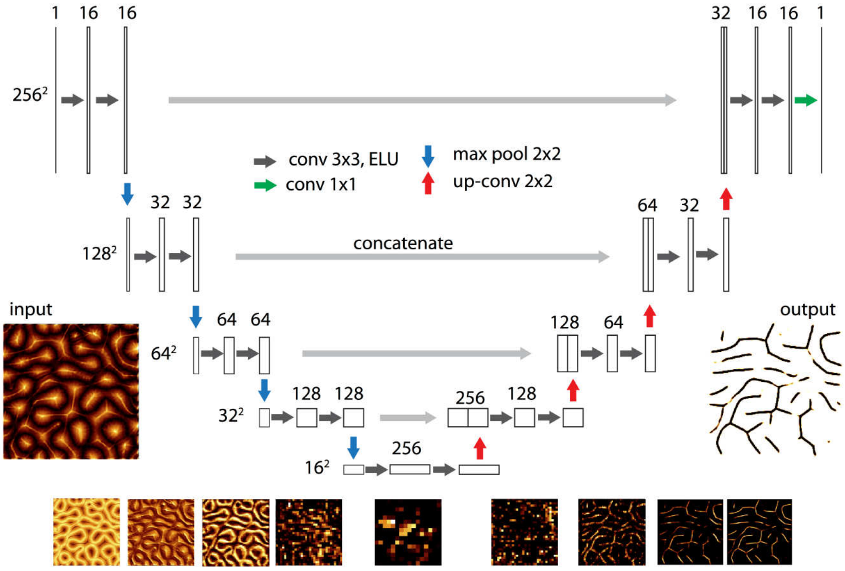

Figure 3. Workflow within the U-shaped deep neural network. The resolution of the image (or feature map) and the number of images are indicated at each step. At the bottom of the figure, we display some levels of the data processing within the hidden layers. The thumbnail contour plots are centred with respect to the corresponding hidden layer. Convolution and max-pooling operations are indicated in the legend. There are a total of 28 layers and 1 940 817 parameters. Figure is taken from the supplementary information of Ref. 3.

rolls in diagonal corners and the short-term switching through roll states parallel to the side faces which have also been seen in other simulations and experiments. The diagonal macroscopic flow states can last as long as 1000 convective free-fall time units.

Schneide *et al.*[12] applied two unsupervised machine learning techniques to a three-dimensional RBC flow with aspect ratio $\Gamma = 16$ – spectral clustering and density-based spatial clustering applications with noise (DBSCAN) – in order to analyse the time evolution of a Lagrangian particle ensemble in the RBC flow as a whole. To this end, the particle trajectories are composed into a network and its connection to the large-scale organisation of an extended three-dimensional turbulent convection flow – termed turbulent superstructures of convection – was studied. The set of N_p individually advected Lagrangian particles at a time t forms a set of vertices $\{v_1, \ldots, v_{N_p}\} \in V$ of a weighted and undirected graph $G = (V, E, w)$. The vertices are connected by edges $\{e_1, \ldots, e_M\} \in E$. The dynamical history of the particle ensemble up to time t can then be encoded into the weights w which are assigned to each edge. These weights are calculated as the inverse of a time-averaged distance of mutual tracer trajectories and set to zero if this distance exceeds a threshold value.[12] The latter is the graph sparsification step. By solving a balanced cut problem via an equivalent generalised eigenvalue problem of the Laplacian matrix of the graph[13] and a subsequent k-means clustering of the leading eigenvectors, the network is decomposed into k subgraphs or clusters. The obtained clusters of the graph could be related to large-scale ridge patterns of temperature as seen in Fig. 2. The dispersion of the Lagrangian particles that start their evolution in close distance to each other is an immanent property of a turbulent flow and can be quantified by the largest Lyapunov exponent which is also shown as a background texture in the middle row panels of Fig. 2. Therefore, such a spectral clustering analysis is applicable for the shorter-time evolution only. For larger evolution times, a density-based clustering of pseudo-trajectories, which are obtained as time averages of Lagrangian trajectories, was applied to probe the long-living patterns in the convection flow. The latter analysis step gives us the most coherent subset of trajectories which are trapped for a long time in the core of the circulation rolls.

3 Supervised machine learning application

Supervised machine learning makes use of the fact that it is often easier to train a system, such as a deep convolutional neural network in the present case, with a number of labelled examples of an intended input-output, than to develop a specific computer program to provide the correct answer for all possible input data.

In Fonda *et al.* we discuss the application of a deep convolutional neural network (DCNN) to quantify the role of turbulent superstructures in the turbulent heat transport. We analysed data records that were obtained in a domain with a large aspect ratio of 25. The Prandtl number of the data sets was $Pr = 7$ and the Rayleigh numbers were $Ra = 10^5, 10^6$, and 10^7. As already stated in the introduction, the superstructures manifest as a temporal skeleton of ridges of hot upwelling and cold downwelling fluid with defects where the ridges merge or end. We trained a DCNN to reduce the three-dimensional temperature field to a temporal planar network in the midplane of the layer – a data compression by more than 5 orders of magnitude at the highest Rayleigh number. The resulting network quantified the turbulent heat transferred by the superstructure. Defect points that disappear and form anew with time are found to be "hot spots", which are points of locally enhanced heat flux (see Fig. 5). It was shown that the fraction of heat carried by the superstructure decreases as the Rayleigh number increases. We also found that an increasing amount of heat is carried by the small-scale background turbulence from the bottom to the top rather than by the coherent large-scale superstructure of the convection flow.

The convolutional deep neural network which we applied is U-shaped as seen in Fig. 3, denoted as U-net for short.[10] The U-net combines a contraction path of a standard convolutional neural network[5] with a subsequent expansion path of concatenations and up-convolutions that finally creates a detailed segmentation map. This U-net has been applied successfully to image segmentation of touching objects, *e. g.* of neuronal cell structures in electron microscopy data. The specific architecture of the U-net is essential for the present usage in a large-aspect-ratio RBC flow as it requires small sets of manually annotated data for the training only. This is in contrast to standard DCNNs for such images. The slow evolution of the superstructures in the large-aspect-ratio domain would otherwise require extremely long simulations of the full RBC flow over a few tens of thousands of convective time units to obtain an appropriate amount of independent training data, as discussed by Pandey *et al.*[9]

The U-net algorithm[10] was used within the Tensorflow software environment[14] in combination with the application programming interface Keras.[7] The temperature data in the midplane were originally obtained at a 1024^2 resolution by a spectral interpolation onto a Cartesian uniform mesh. We verified that a reduction of the resolution to 256^2 grid

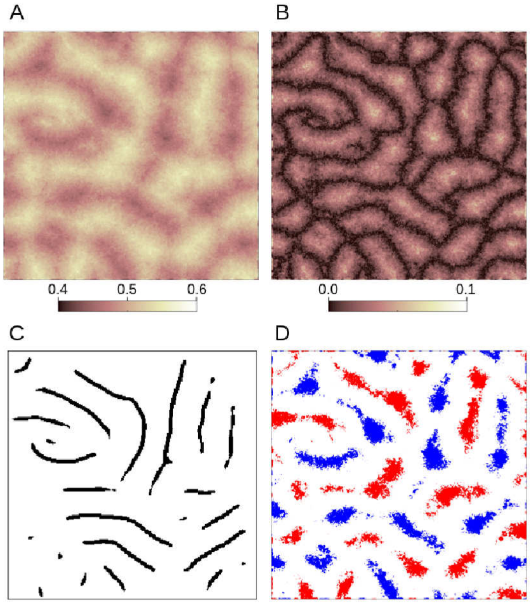

Figure 4. Input and output of the U-net. (A) Original two-dimensional and time-averaged temperature slice for $Ra = 10^7$ and $Pr = 7$. (B) Corresponding symmetrised temperature field $\Theta(x, y)$ that enters the U-net. (C) Ridge pattern as output of the U-net. (D) Comparison with a simple thresholding set. Blue pixels are for cold downflows and red pixels for hot upflows. These ridges have a strongly varying thickness and are partly interrupted when a simple thresholding is conducted. The figure is taken from Ref. 3.

points that cover the whole plane is sufficient to capture the ridge pattern. The training of the neural network for 1000 epochs took about 15 minutes on a cluster with NVIDIA GeForce GTX 1080. Note, however, that even 100 epochs were enough in our case. Fig. 4 demonstrates the extraction of the ridge pattern by the deep neural network, even though the data were for the highest Rayleigh number case very noisy. Input to the U-net is the symmetrised temperature field which is given by $\Theta(x, y) = |T(x, y, z = 1/2) - 1/2|$ and shown in panel B of the figure.

Since supervised machine learning is highly sensitive to the quality of the training data, the annotation of the samples is a key task. Different Rayleigh and Prandtl numbers cause changes in the sharpness, the size, and the amount of noise of the underlying ridge network of temperature field superstructures. In Fonda *et al.*[3] the trends with Rayleigh number at $Pr = 7$ were investigated. Here, we will take a closer look at the Prandtl number dependence at fixed Ra that affects the superstructures in a different way as discussed in Pandey *et al.*[9] in detail. Two scenarios were tested. In the first case, we created training data for the U-net based on low-noise snapshots with a clearly visible ridge network (see Fig. 5 (A, E)). The samples were highly augmented by several geometric and photometric manipulations including artificial blurring, addition of noise and distortion. In the second case, examples from three different Prandtl numbers, $Pr = 7, 0.7$ and 0.021, were combined for the training of the U-net leaving the corresponding manual annotations of the ridges much less detailed. It is found that the training data that contained many details and were highly

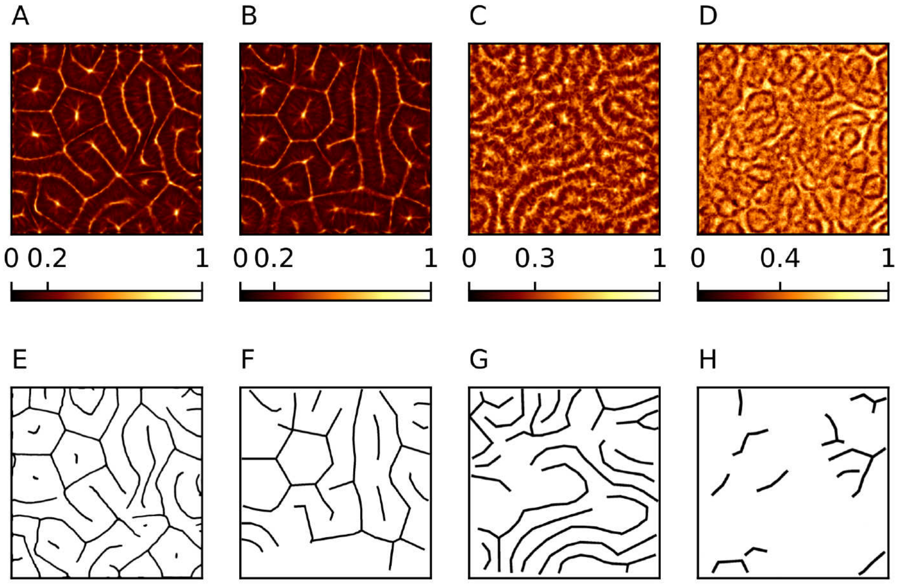

Figure 5. Different training data annotations of the symmetrised temperature field $\Theta(x, y)$ all for a Rayleigh number $Ra = 10^5$. (A, E) sample and annotation for $Pr = 7$. (B, F) again for $Pr = 7$. (C, G) for $Pr = 0.7$, and (D, H) for $Pr = 0.021$. All input data are normalised such that they vary between 0 to 1 and can be combined. The colour bars below the top row of panels also indicate the corresponding mean value.

augmented (case 1) slow down the convergence of the DCNN significantly. The corresponding U-net gives very detailed image segmentations. In contrast, the U-net which was trained with examples from all three simulation records (case 2) converged faster and produced segmentations that show the coarsest ridges only. The results of the ridge extraction with both U-nets are summarised in Fig. 6 for two different Prandtl numbers. Depending on the subsequent use of the output, one either obtains a fine-scale segmentation or a much coarser ridge pattern. Our present results show clearly that the application of the U-net becomes more challenging as the Prandtl number decreases. In these cases, the ridge patterns of the symmetrised temperature are more washed out due to the more vigorous fluid turbulence and the ridges appear to be more disconnected. Clearly, even the U-net approaches here its limitations.

The extracted ridge networks, as obtained in Fig. 6 (C, G), were subsequently used to quantify the heat transport across the convection layer. The dimensionless number that measures the turbulent heat transport in RBC is the Nusselt number, which is given by $Nu = 1 + \sqrt{RaPr}\langle u_z T\rangle_{V,t}$ with the vertical velocity component u_z, the temperature T, and the combined volume-time average $\langle\cdot\rangle_{V,t}$. In Fig. 7 (A) we compare Nu with a Nusselt number that measures the transport across the ridge network (and thus due to the turbulent superstructure (TSS)). It is given by $Nu^{\mathrm{TSS}} = \sqrt{RaPr}\, w_e \langle u_z \Theta\rangle_r$. Here, w_e is a weight factor that represents the area fraction of the network compared to the area content of the horizontal plane and r a segment of unit length in the ridge network. Fig. 7 (B-D) show the transport capability of the ridge network for three Pr. The heat flux by the

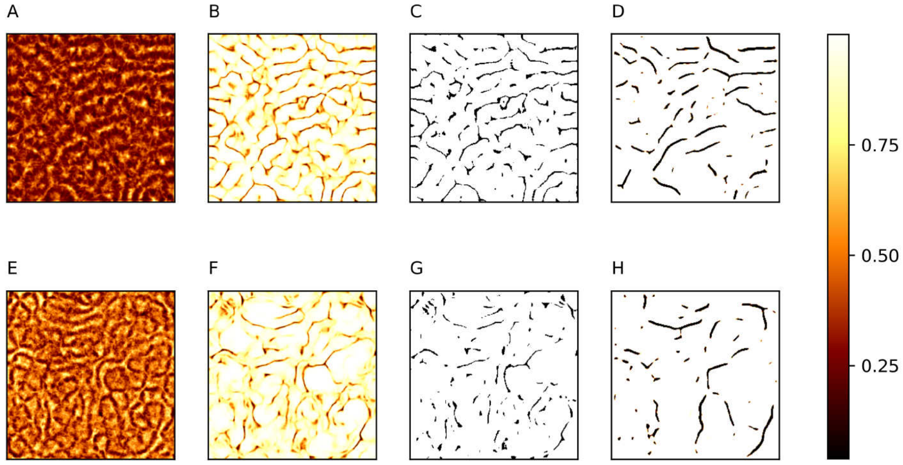

Figure 6. Output of the U-net for $Pr = 0.7$ (top row) and $Pr = 0.021$ (bottom row), both at $Ra = 10^5$. (A, E) Normalised input data of symmetrised temperature. (B, F) U-net output when trained with detailed annotation and high augmentation rate. (C, G) Conversion of (B, F) into a binary data set with a threshold of 0.5 is shown. (D, H) Direct output result of the U-net trained with less detailed annotations.

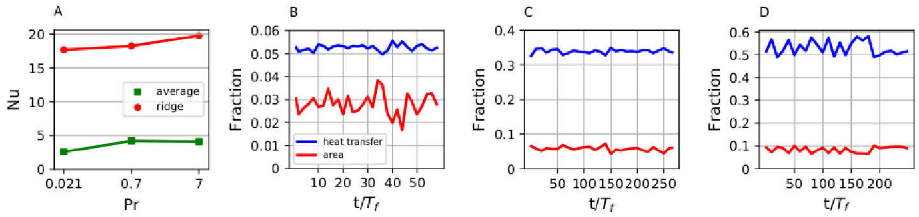

Figure 7. Turbulent heat transport analysis across the ridge network. (A) Mean convective heat transport across the whole plane (green) and transport due to the ridge network (red). (B-D) Amount of heat transport across the network *vs.* time and area fraction of the network *vs.* time. Data are for at $Ra = 10^5$ and $Pr = 0.021$ in B, $Pr = 0.7$ in C and $Pr = 7$ in D.

network increases faster than the area occupied by it with increasing Pr, suggesting that the transport by the ridge network becomes more efficient.

4 Concluding remarks

We have discussed a number of applications of unsupervised and supervised machine learning in turbulent convection. They were used to extract coherent patterns in these flows and to reduce the big amount of three-dimensional simulation data effectively. As in many other applications, we showed that the success of supervised machine learning with DCNNs is tightly related to the training methods. Future tasks in this field are related to predictions of turbulent flow behaviour itself, *e. g.* by the use of recurrent neural networks and thus to construct effective small-scale parametrisations. These efforts are currently in progress and will be reported elsewhere.

Acknowledgements

The work of D. K. is supported by the Deutsche Forschungsgemeinschaft (DFG) and the work of A. P. and C. S. by the Priority Programme DFG-SPP 1881 *Turbulent Superstructures*. JS thanks the Tandon School of Engineering at New York University for financial support. We want to thank the John von Neumann-Institute for Computing for the steady support with supercomputing resources that make our research work in turbulent convection possible. The reported work was and is supported by the project HIL12.

References

1. F. Chillà and J. Schumacher, *New perspectives in turbulent Rayleigh-Bénard convection*, Eur. Phys. J. E **35**, 58 (25 pages), 2012.
2. K. Duraisamy , G. Iccarino, and H. Xiao, *Turbulence modeling in the age of dat*a, Annu. Rev. Fluid Mech. **51**, 357–377, 2019.
3. E. Fonda, A. Pandey, J. Schumacher, and K. R. Sreenivasan, *Deep learning in turbulent convection networks*, Proc. Natl. Acad. Sci. USA **116**, 8667–8672, 2019.
4. D. Giannakis, A. Kolchinskaja, D. Krasnov, and J. Schumacher, *Koopman analysis of the long-term evolution in a turbulent convection cell*, J. Fluid. Mech. **847**, 735–767, 2018.
5. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, The MIT Press, Cambridge, MA, 2017.
6. P. Holme and J. Saramäki, *Temporal networks*, Phys. Rep. **519**, 97–125, 2012.
7. F. Chollet *et al.*, *Keras*, `https://keras.io`, 2015.
8. `https://nek5000.mcs.anl.gov/`
9. A. Pandey, J. D. Scheel, and J. Schumacher, *Turbulent superstructures in Rayleigh-Bénard convection*, Nat. Commun. **9**, 2118 (11 pages), 2018.
10. O. Ronneberger, P. Fischer, and T. Brix, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Lecture Notes in Computational Science **9351**, 234–241, 2015.
11. J. D. Scheel, M. S. Emran, and J. Schumacher, *Resolving the fine-scale structure in turbulent Rayleigh-Bénard convection*, New J. Phys. **15**, 113063, 2013.
12. C. Schneide, A. Pandey, K. Padberg-Gehle, and J. Schumacher, *Probing turbulent superstructures in Rayleigh-Bénard convection by Lagrangian trajectory clusters*, Phys. Rev. Fluids **3**, 113501 (15 pages), 2018.
13. J. Shi and J. Malik, *Normalized Cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell. **22**, 888–905, 2000.
14. M. Abadi *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, `https://www.tensorflow.org`, 2015.
15. M. O. Williams, I. O. Kevrekidis, and C. W. Rowley, *A data-driven approximation of the Koopman operator: Extending Dynamic Mode Decomposition*, J. Nonlinear Sci. **25**, 1307–1346, 2015.