Two-dimensional convective boundary layer: Numerical analysis and echo state network model

Florian Heyder$^1$, Juan Pedro Mellado$^2$ and Jörg Schumacher$^{1,3}$

$^1$Institut für Thermo- und Fluiddynamik, Technische Universität Ilmenau, Postfach 100565, D-98684 Ilmenau, Germany

$^2$Department of Physics, Aerospace Engineering Division, Universitat Politècnica de Catalunya, C. Jordi Girona 1–3, 08034, Barcelona, Spain

$^3$Tandon School of Engineering, New York University, New York City, NY 11201, USA

(Received xx; revised xx; accepted xx)

The numerical study of global atmospheric circulation processes requires the parametrization of turbulent buoyancy fluxes in the lower convective boundary layer which typically cannot be resolved by the coarse-scale computational grids. In the present work, a two-dimensional model of a shallow convective boundary layer in the Boussinesq limit is investigated by direct numerical simulations. A series of simulation runs evaluates the turbulent transport properties as a function of the ratio between the prescribed buoyancy fluxes at the top and the bottom of the layer. Our model is able to reproduce essential properties of the lower convective boundary layer. The resulting data records are subsequently used to train and test a recurrent neural network which is realized by an echo state network with a high-dimensional reservoir. It is shown that the echo state network reproduces the turbulence dynamics and the statistical properties of the buoyancy flux across the layer very well and is thus able to model these transport processes without solving the underlying highly nonlinear equations of motion. Focus is given to the generalization properties of the echo state network, i.e., its ability to model unseen data records with a different flux ratio.

Key words: Convective boundary layer, Recurrent neural network, Reservoir computing

1. Introduction

Turbulent convection prevails in the diurnal atmospheric boundary layer over land in weak-wind conditions. The incoming solar radiation warms the surface, and the heat transfer to the air in contact with it creates a superadiabatic temperature profile that leads to convective instability. We refer to this regime of turbulent convection in the atmospheric boundary layer as the convective boundary layer (CBL, Wyngaard 2010). Because of its relevance, the CBL has been extensively studied during the last century, but important questions remain open regarding non-local effects induced by the large-scale organization of the flow, its parametrization in larger-scale atmospheric circulation models, and its sensitivity to changes in environmental conditions (Fodor et al. 2019; LeMone et al. 2019; Edwards et al. 2020). In this paper, we explore the applicability of machine learning methods to address these issues.

Machine learning methods have changed paradigms of processing data and building data-driven parametrizations of unresolved turbulent processes in fluid mechanics (Brenner et al. 2019; Brunton et al. 2020; Pandey et al. 2020) and atmospheric science (Gentine et al. 2018; O’Gorman

† Email address for correspondence: florian.heyder@tu-ilmenau.de
Dwyer 2018), and might be able to provide a step forward in the representation of essential properties of CBLs on coarser numerical models. Given the strong variability in the environmental conditions, we are particularly interested in how robust are machine-learning methods when applied to unseen configurations that are different from the training configurations, an essential aspect that directs to generalization properties of supervised learning methods (Goodfellow et al. 2016). This sets the stage for the present work which consists of two major parts.

In a first step, following previous work in the atmospheric context (Sorbjan 1996; Fodor et al. 2019), this paper introduces a two-dimensional Boussinesq model of turbulent convection in a cell as a model of the atmospheric CBL in cloud-free and shear-free conditions. We impose constant buoyancy fluxes at the bottom and top boundaries. A major difference between this CBL model and the atmospheric CBL is that the depth of the atmospheric CBL increases with increasing time as the CBL grows into the free troposphere (penetrative convection; e.g., Adrian et al. 1986; Zilitinkevich 1991), whereas the CBL model that we use has a constant depth (non-penetrative convection). However, this growth is slow compared to the large eddy turnover times, which explains that important aspects of the structure and dynamics of the CBL are well represented, at least qualitatively, by the present CBL model, such as the well-mixed properties in the center of the convective region, the large convective cell organization, and the structure of the surface layers (Sorbjan 1996; Fodor et al. 2019). Although the present CBL model does not represent the entrainment of fluid from the free troposphere into the turbulent region that occurs in the atmospheric CBL, the model retains the effect of entrainment warming at the CBL top, which creates an upper layer of stable fluid that is important for the CBL dynamics and a challenge for the parametrization of mixing in atmospheric models. This CBL model is similar to classical Rayleigh-Bénard convection (RBC) in that it represents turbulent convection between two solid plates (Chillà & Schumacher 2012). For this reason, we refer to the present CBL model as an RBC-like configuration. Differently to most RBC studies, however, the present CBL model considers constant-flux conditions at the bottom and top boundaries instead of constant-temperature conditions. These boundary conditions provide a better approximation to the atmospheric CBL (Stull 1988; Wyngaard 2010). By means of direct numerical simulation (DNS), we study this CBL model at different ratios of the buoyancy flux at the top and bottom boundaries, which alters the convection patterns.

In a second step, we use the DNS data as a training base and study the performance of machine learning-based models of turbulent convection when applied to systems that are different from the training configuration. This study addresses an important open point of supervised machine learning algorithms, namely how well do they perform with respect to unseen data at changed conditions (point known as the generalization properties, as mentioned already above). The aim is to ascertain the potential of RBC-like configurations to develop machine learning-based models of turbulent mixing in oceanic and atmospheric boundary layers in convective regimes. The advantage of using RBC-like configurations is their statistically-steady state, which allows detailed studies under controlled conditions and better statistical convergence compared with the unsteady and quasi-steady CBL (Fodor et al. 2019).

More specifically, we apply echo state networks (ESN) which are one implementation of reservoir computing (Jaeger & Haas 2004; Lukoševičius et al. 2012). Reservoir computing (RC) uses a simple nonlinear dynamical system with recurrent connections for time series prediction. For this, the recurrent network, called the reservoir, maps a given input signal to a high-dimensional space. The dynamical state of the reservoir is then subject to external forcing by this input. After a sufficiently long propagation phase, a linear output rule is computed which maps the reservoir state of each iteration to the target output which is then the input at the next time step. The learned output rule and the initially created reservoir can then be used for the task of prediction. The RC approach is in contrast to conventional neural networks, where all parameters, i.e., weights and biases, are tuned by an optimization scheme such as stochastic gradient descent.
The concept was simultaneously proposed by Jaeger (2001) and Maass et al. (2002). Since its introduction, a large variety of such reservoir computers have been proposed (Nakajima 2020). We mention the use of water waves and their interferences (Fernando & Sojakka 2003), photonic (Vandoorne et al. 2011), and spintronic systems (Tsunegi et al. 2019) as well as novel quantum computing approaches (Fujii & Nakajima 2017, 2020).

The ESN approach has found wide interest recently in inferring states of a nonlinear dynamical system. Applications showed for example that the dynamics of two of the three degrees of freedom of the Rössler system can be inferred from the evolution of the third one (Lu et al. 2017). Further, the Lyapunov exponents of the dynamical system that a trained ESN represents have been shown to match the exponents of the data generating system (Pathak et al. 2017). Moreover, hybrid models which combine both data driven (ESN) and knowledge based methods, i.e. solving the mathematical equations, have already been proposed (Pathak et al. 2018; Wikner et al. 2020) and tested in terms of a global atmospheric forecast model (Arcomano et al. 2020). Further, RC techniques, due to their computationally inexpensive training routine, could serve as a lightweight substitute for conventional parameterization schemes. Other neural network architectures have already been tested as subgrid scale parameterization (Pawar & San 2021) or for the analysis of flow experiments (Moller et al. 2020). The performance of ESNs in two-dimensional dry and moist turbulent Rayleigh-Bénard convection have already shown great promise, as low-order statistics of buoyancy and liquid water fluxes are successfully reproduced (Pandey & Schumacher 2020; Heyder & Schumacher 2021).

The outline of the manuscript is as follows. In section 2, we present the two-dimensional Boussinesq model of the convective boundary layer and define all essential parameters, in particular, the ratio of the buoyancy fluxes at the top and bottom boundaries, $\beta$. The four DNS runs at different $\beta$ and the resulting statistical properties are analysed and discussed. Section 3 introduces first the architecture of the echo state network and discusses the echo state property. It is followed by details on the training and the results of the trained echo state network in comparison to the test data. We summarize our results and give a brief outlook in the final section.

2. Boussinesq model for convective boundary layer

2.1. Governing equations

To study the turbulent convection problem, we use the Boussinesq approximation to the two-dimensional Navier-Stokes equations. For convenience and generality in case water-vapor effects become relevant, we formulate the problem in terms of the buoyancy. For temperature-only driven configurations, the buoyancy can be related to the temperature by $b = \alpha g T$, where $\alpha$, $g$, $T$ are the thermal expansion coefficient, gravitational acceleration and temperature, respectively. We consider a cell of height $H$ and length $L$ (see figure 1). In the vertical direction, we consider no-slip boundary conditions for the velocity and constant-flux boundary conditions for the buoyancy. We impose the fluxes $B_0$ and $B_1$ at the bottom and top respectively. In the horizontal direction, we consider periodic boundary conditions.

We non-dimensionalize the equations with the cell height $H$, the convective velocity $(B_0 H)^{1/3}$, the convective time $(H^2 / B_0)^{1/3}$, and the convective buoyancy $(B_0^2 / H)^{1/3}$ (Deardorff 1970). Unless stated otherwise, from here on we consider all quantities to be in their dimensionless form, i.e. scaled by the cell height, convective time, convective velocity or convective buoyancy.
The resulting evolution equations are given by

\[
\frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} = \frac{\partial p}{\partial x} + 3 \frac{\sqrt{Pr Ra_c}}{Ra_c} \left( \frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} \right)
\]  
(2.2)

\[
\frac{\partial u_y}{\partial t} + u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} = \frac{\partial p}{\partial y} + 3 \frac{\sqrt{Pr Ra_c}}{Ra_c} \left( \frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} \right) + b
\]  
(2.3)

\[
\frac{\partial b}{\partial t} + u_x \frac{\partial b}{\partial x} + u_y \frac{\partial b}{\partial y} = \frac{1}{3 \sqrt{Pr Ra_c}} \left( \frac{\partial^2 b}{\partial x^2} + \frac{\partial^2 b}{\partial y^2} \right)
\]  
(2.4)

The boundary conditions are

\[
\frac{\partial b}{\partial y} (x, y = 0, t) = -\sqrt{Pr Ra_c}
\]  
(2.5)

\[
\frac{\partial b}{\partial y} (x, y = 1, t) = \beta \sqrt{Pr Ra_c}
\]  
(2.6)

where \( \beta \) is the buoyancy-flux ratio

\[
\beta = \frac{B_1}{B_0}.
\]  
(2.7)

As further explained below, we are interested in the cases \( B_0 > 0 \) and \( B_1 < 0 \), i.e., the fluid is heated from the bottom and from the top. Here, the Prandtl number is the ratio of kinematic viscosity \( \nu \) and molecular diffusivity \( \kappa \)

\[
Pr = \frac{\nu}{\kappa}
\]  
(2.8)

and the convective Rayleigh number is

\[
Ra_c = \frac{B_0 H^4}{\nu \kappa^2}.
\]  
(2.9)

The buoyancy difference

\[
\Delta b = \langle b \rangle_x (y = 0, t) - \langle b \rangle_x (y = 1, t)
\]  
(2.10)

between the two plates is a dependent variable for configurations with constant-flux boundary conditions and needs to be diagnosed from experimental or simulation data (angle brackets indicate an averaging operation and the subscript indicates the variable with respect to which the averaging operation is performed, in this case, the horizontal coordinate \( x \)). Therefore, the Dirichlet Rayleigh number

\[
Ra_f = \Delta b \sqrt{\frac{Ra_c^2}{Pr}}
\]  
(2.11)

is a diagnostic variable as well. From eqns. (2.1)–(2.4) one can derive the vertical buoyancy profile, up to a constant, for the purely conductive case to be

\[
b_{\text{cond}} = \sqrt{Pr Ra_c} \frac{(y - 1)^2 + \beta y^2}{2} + (1 + \beta) t + \text{constant}.
\]  
(2.12)

For \( \beta = -1 \), we recover the steady, linear solution that corresponds to the problem with Dirichlet boundary conditions. In the Neumann case, the profile of \( b \) has a parabolic shape and it grows linearly in time (given that we heat from below and from above). Nonetheless, it is quasi-steady in the sense that the shape of the profile remains constant in time. The buoyancy difference between
Figure 1: Scheme of the two-dimensional Rayleigh-Bénard setup with constant buoyancy flux boundary conditions. The bottom of the cell is heated by the incoming flux $B_0$. We explore the effect of asymmetric boundary conditions by imposing a different flux $B_1 = -\beta B_0$ ($\beta > 0$) at the top. The values $\beta \in \{0.1, 0.2, 0.3\}$ are representative values for convective boundary layers. Here we consider a) Adiabatic top ($\beta = 0$) and b) warming flux at the top ($\beta > 0$).

2.2. Direct numerical simulations

We fix the Prandtl and convective Rayleigh number to $Pr = 1$ and $Ra_c = 3 \cdot 10^8$ and consider extended layers with an aspect ratio $\Gamma = L/H = 24$. The control parameter that we vary is the flux-ratio parameter $\beta$ defined by eq. (2.7). In the atmospheric CBL over land, one typically finds the conditions $B_0 > 0$ and $B_1 < 0$, which represent the surface warming and the entrainment warming of the CBL, respectively. Hence, we are interested in the case $\beta > 0$. Typical atmospheric conditions correspond to the range $\beta \approx 0.1 - 0.3$ (Stull 1988; Wyngaard 2010). As $\beta$ increases, the upper region of the convective cell increasingly stabilizes (positive mean buoyancy gradient, see also later in figure 5), and preliminary simulations (not shown) indicate that the dynamics strongly change for values $\beta \approx 0.4$. Therefore, we consider the cases $\beta \in \{0.0, 0.1, 0.2, 0.3\}$ in our CBL model (see also figure 1). The case $\beta = 0$ corresponds to an upper adiabatic wall. This case is considered as a first step to understand the effect of asymmetries in the boundary conditions in the results obtained from Rayleigh-Bénard convection with constant-buoyancy boundaries.

The Boussinesq equations (2.1) – (2.4) are discretized by a high-order spectral-like compact finite difference method. The time evolution is treated by a low-storage fourth-order Runge-Kutta scheme. More details on the numerical method can be found in Mellado and Ansorge (2012). The software used to perform the simulations is freely available at https://github.com/turbulencia/tlab.

The grid size is $N_x \times N_y = 2400 \times 150$. The horizontal grid spacing is uniform. The vertical grid spacing follows a hyperbolic tangent profile: it is equal within 1.2% to the horizontal grid spacing in the center of the convection cell, and diminishes by a factor of 2.5 next to the wall. The time steps are in the range $\Delta t \approx 0.0012 - 0.0016$, the specific value depending on the simulation. They are defined to obtain data exactly every 0.25 free-fall times (definition follows). Since the free-fall time is a derived variable in the case of constant-flux boundaries considered in this study, preliminary simulations were performed to obtain the free-fall time in each case, and we repeated the simulations with the appropriate $\Delta t$. Table 1 summarizes important parameters of the four simulation runs.

2.3. Cellular structure of the convection layer and vertical profiles at different $\beta$

In the turbulent case, the fluid warms linearly with respect to progressing time as in the conduction case. Integrating the evolution equation for $b$ yields that the volume averaged buoyancy
Table 1: Simulation parameters. The time average has been calculated over the last 500 free-fall times.

| $\beta$ | $\langle \Delta b \rangle_t$ | $\langle T_f \rangle_t$ | $\langle \text{Nu} \rangle_t$ | $\langle \text{Ra} \rangle_t$ |
|---------|-----------------|-----------------|-----------------|-----------------|
| 0.0     | 22.0 ± 0.4     | 0.21            | 15.2 ± 0.3     | 0.99 × 10^7    |
| 0.1     | 19.0 ± 0.4     | 0.23            | 15.9 ± 0.3     | 0.85 × 10^7    |
| 0.2     | 15.4 ± 0.3     | 0.26            | 17.4 ± 0.4     | 0.69 × 10^7    |
| 0.3     | 10.7 ± 0.5     | 0.31            | 21.9 ± 0.9     | 0.48 × 10^7    |

Figure 2: Temporal variation of buoyancy difference $\Delta b$, see eq. (2.10), and Nusselt number $\text{Nu}$, see eq. (2.15). Both quantities become statistically stationary after an initial transient. Note that both time axis are normalized by the time mean of the free fall time $T_f = 1/\Delta b$ in the statistical stationary regime.

$\langle b \rangle_{x,y}$ increases as

$$\langle b \rangle_{x,y} = (1 + \beta)t.$$  

The mean vertical profile, however, is different to the pure conduction profile and, as mentioned above, a major dependent variable is the buoyancy difference $\Delta b$ across the cell. After an initial transient, this quantity becomes statistically stationary, as can be seen in figure 2(a) for all four simulations. The free fall time $T_f = 1/\sqrt{\Delta b}$ and free fall velocity $U_f = \sqrt{\Delta b}$ can be computed and used as scales for better comparison to the more common case of Rayleigh-Bénard convection with constant-buoyancy boundaries. Moreover, we can express the buoyancy difference in terms of a Nusselt number

$$\text{Nu} = \frac{\Delta b_{\text{cond}}}{\Delta b} = \frac{(1 - \beta) \sqrt{\text{PrRa}_c}}{2\Delta b},$$

defined here as the ratio between the buoyancy difference in the purely conductive case $\Delta b_{\text{cond}}$ (see eq. (2.13)), and the fully convective case, i.e. $\Delta b$. For $\beta = -1$, we again recover the functional relationship corresponding to Rayleigh-Bénard convection with constant-buoyancy boundaries. The relaxation to a statistically stationary state for the buoyancy difference and the Nusselt number are demonstrated in figure 2 for all four cases.

Figures 3 and 4 show snapshots of the normalized buoyancy, which is given by

$$b^*(x, y) = \frac{b(x, y) - \langle b \rangle_{x,t}(y = 1)}{\langle \Delta b \rangle_t},$$
Figure 3: Instantaneous snapshot of the normalized buoyancy field $b^* = (b - \langle b \rangle_{x,t}(y = 1))/\langle \Delta b \rangle_t$ in the statistically stationary state. The four different top boundary conditions ($\beta = 0.0, 0.1, 0.2, 0.3$) differ in their width of the top thermal boundary layer. For the adiabatic top $\beta = 0.0$ no such layer is present. Note that with increasing $\beta$ the range of $b^*$ increases.

and the vertical flux $u'_y(x, y)b'(x, y)$ in the statistically stationary regime. For $\beta = 0.0$ the flux at the top is zero and no thermal boundary layer is present. This changes when the warming flux at the top becomes greater than zero, i.e. $\beta > 0$. With increasing warming flux at the top we find thermal boundary layer at $y = 1$, which increases in thickness as $\beta \to 0.3$. Naturally, the structures in the buoyancy flux are also affected by the change of the top flux. As more buoyant fluid is transported in from the top, the cellular order is increasingly dissolved.

Figure 4: Instantaneous snapshot of the vertical buoyancy flux $u'_y(x, y)b'(x, y)$ in the statistically stationary state. The cellular order is increasingly dissolved with growing parameter $\beta$.

We show the line-time average vertical profiles $\langle \cdot \rangle_{x,t}(y)$ of $b^*$ in figure 5(a). All profiles show the tendency towards a constant mean value in the central part of the domain, implying a layer of well-mixed fluid. Contrary to the common Rayleigh-Bénard case with constant-buoyancy boundary conditions, constant-flux boundary conditions break the top-down symmetry of the mean buoyancy profile. Furthermore, for $\beta > 0$, the incoming warming flux at $y = 1$ results in positive gradients at the top, which results in a stable layer at the top.
Figure 5: Vertical profiles of a) the normalized buoyancy $b^* = (b - \langle b \rangle_{x,t}(y = 1))/\langle \Delta b \rangle_t$, b) buoyancy fluctuations, c) vertical velocity fluctuations, d) normalized total buoyancy flux $F_b/F_b|_{y=0}$. While the boundary conditions significantly affect the buoyancy and its fluctuations, the influence on the vertical velocity profiles is less important. The fluxes show linear variation across the cell. The legend shown in a) is valid for all graphs shown.

We examine the variability of the fields and decompose both into their volume mean $\langle u_x \rangle_{x,y}, \langle u_y \rangle_{x,y}, \langle b \rangle_{x,y}$ and their fluctuations $u'_y, b'$

\begin{align}
  u_x(x, y, t) &= \langle u_y \rangle_{x,y}(t) + u'_y(x, y, t), \\
  u_y(x, y, t) &= \langle u_y \rangle_{x,y}(t) + u'_y(x, y, t), \\
  b(x, y, t) &= \langle b \rangle_{x,y}(t) + b'(x, y, t).
\end{align}

Note that $\langle b \rangle_{x,y}$ depends on time, as it incorporates the linear warming of the fluid. Meanwhile, $\langle u_x \rangle_{x,y}$ and $\langle u_y \rangle_{x,y}$ are statistically stationary and vary weakly about their zero mean. The vertical profiles of the root mean square (r.m.s.) of $u'_y$ and $b'$ are shown in figures 5(b) and (c). The r.m.s. of the fluctuations of the buoyancy differ greatly in their magnitude and trend in the upper portion of the domain. The vertical r.m.s. velocity component $\langle u'^2 \rangle_{x,t}^{1/2}$, on the other hand, does not vary too much while changing $\beta$. Additionally, the total buoyancy flux

\begin{equation}
  F_b = \langle u'_y b' \rangle_{x,t} - Ra_c^{-1/3} \frac{\partial \langle b \rangle_{x,t}}{\partial y}
\end{equation}

normalized by its bottom value is shown in figure 5(d). We find that the flux decreases linearly...
with increasing height. As indicated by figure 5(a), the molecular terms mostly contribute to the near-wall regions. The turbulent transport (not shown), on the other hand, declines linearly over the middle of the domain and results in negative contributions near the top. This is expected by the stabilization by entrainment warming in the CBL (Wyngaard 2010), here considered by imposing the negative buoyancy flux $B_1$ at the top boundary. One goal of this study is to reproduce the vertical profiles of the turbulent contributions by the recurrent neural network which will be presented in the next section.

For now, we can conclude that the present two-dimensional model of a convective boundary layer incorporates already several important physical properties that are known from real atmospheric CBL in cloud-free conditions and without an imposed shear flow. To underline this remark, we add profiles of a CBL case of as dashed brown lines to all 4 panels of figure 5. This CBL is a 2D version of 3D configurations used by Fodor et al. (2019) to compare the CBL with RBC, and represents penetrative convection, i.e., it retains entrainment and the growth of the CBL depth that are present in the atmospheric CBL. One important property, which goes beyond the classical Rayleigh-Bénard convection case (Chillà & Schumacher 2012) is the top-down asymmetry which is obvious from the mean vertical profiles.

In the following, we use the DNS data of $\beta \neq 0$ to train a recurrent neural network and make subsequent predictions for an unseen data set. This is done to explore the generalization properties of the echo state networks. We therefore interpolate all fields from the non-uniform grid with $2400 \times 150$ points to a $720 \times 30$ uniform grid by cubic splines. This grid will be denoted as the coarse-grained grid, the data as coarse-grained DNS data.

### 3. Convective boundary layer statistics from an echo state network

#### 3.1. Echo state network and echo state property

In the following, we specify the architecture of the ESN that will be applied to process the DNS data of the CBL model described in the previous section. The reservoir state dynamics is given by

$$r(n) = (1 - \gamma)r(n - 1) + \gamma \tanh \left[ W^r r(n - 1) + W^{in} x(n) + d \right]. \quad (3.1)$$
where \( \mathbf{r}(n) \in \mathbb{R}^{N_r}, \mathbf{x}(n) \in \mathbb{R}^{N_{in}} \) are the reservoir state and input at time step \( n \) respectively. \( \mathbf{W}^r \in \mathbb{R}^{N_r \times N_r}, \mathbf{W}^{in} \in \mathbb{R}^{N_{in} \times N_r} \) are the reservoir and input weight matrices and \( \gamma \in [0, 1] \), \( d \) are the constant leaking rate and constant bias. In this formulation the components of the reservoir state \( r_i \) are automatically \( r_i \in [-1, 1] \). The reservoir output \( \hat{\mathbf{y}} \in \mathbb{R}^{N_{in}} \) is computed by a linear mapping of the extended reservoir state \( \hat{\mathbf{r}}(n) = [d, \mathbf{x}(n), \mathbf{r}(n)] \) (vertical concatenation of bias, reservoir input and state)

\[
\hat{\mathbf{y}}(n) = \mathbf{W}^\text{out} \hat{\mathbf{r}}(n).
\]

The fitted output weights \( \mathbf{W}^\text{out} \in \mathbb{R}^{N_{in} \times (1+N_{in}+N_r)} \) are chosen as to minimize the mean square cost function

\[
C(\mathbf{W}^\text{out}) = \sum_{n=-T_L}^{-1} \left\| \mathbf{y}(n) - \mathbf{W}^\text{out} \hat{\mathbf{r}}(n) \right\|^2_2 + \lambda \left\| \mathbf{w}^\text{out}_i \right\|^2_2,
\]

where \( \mathbf{y} \) are the target outputs, which are part of the training data. \( T_L \) is the number of training time steps, \( \mathbf{w}^\text{out}_i \) is the \( i \)th row of \( \mathbf{W}^\text{out} \) and \( \| \cdot \|_2 \) denotes the \( L^2 \) norm. The last term penalizes large values of the rows of the output weight matrix by adjusting the regression parameter \( \lambda \). This concept is one possibility to counter the problem of overfitting, where the machine learning algorithm learns the training data by heart, consequently performing poorly when operating on data outside the training data set. The solution to this \( L^2 \)-penalized linear regression problem is given by

\[
\mathbf{W}^\text{out} = \mathbf{Y} \mathbf{R}^T \left( \mathbf{R} \mathbf{R}^T + \lambda \mathbf{I} \right)^{-1}
\]

where the \( n \)th column of \( \mathbf{Y} \in \mathbb{R}^{N_{in} \times T_L}, \mathbf{S} \in \mathbb{R}^{N_r \times T_L} \) are \( \mathbf{y}(n) \) and \( \hat{\mathbf{r}}(n) \) respectively. \( \mathbf{I} \in \mathbb{R}^{N_r \times N_r} \) denotes the identity matrix and \((\cdot)^T, (\cdot)^{-1}\) are the transpose and inverse. After the training phase an initial input is given at \( n = 0 \) and the reservoir output at time step \( n \geq 0 \) is fed back to the input layer, by letting \( \mathbf{x}(n) = \mathbf{W}^\text{out}_r \mathbf{r}(n-1) \). During this testing phase the ESN autonomously predicts the next \( T_T \) iterations of the initial input. Figure 6 summarizes the architecture of the ESN in a sketch.

This inexpensive training procedure comes at a cost of finding a suitable set of hyperparameters, i.e., parameters which are not learned and have to be tuned beforehand. Here we restrict ourselves to \( h = \{ \gamma, \lambda, N_r, D, \varrho \} \). The last two quantities are the reservoir density \( D \) and spectral radius \( \varrho \). They are algebraic properties of the reservoir weight matrix and represent the number of non-zero elements and largest absolute eigenvalue of \( \mathbf{W}^r \), respectively. Finding a right setting of these hyperparameters is crucial, as they influence the memory capacity of the reservoir (Hermans & Schrauwen 2010). In Jaeger (2001) a necessary condition for an effective reservoir was proposed: the \textit{echo state property}. A reservoir is said to possess echo states when two different reservoir states \( \mathbf{r}_1(n-1), \mathbf{r}_2(n-1) \) converge to the same reservoir state \( \mathbf{r}(n) \), provided the same input \( \mathbf{x}(n) \) is given and the system has been running for many iterations \( n \). This property highly depends on the data one uses, a suitable set of hyperparameters \( h \), as well as the reservoir initialization (Lukoševičius 2012). So far, no universal rule for the presence of echo states has been proposed. On top of that, the echo state property is merely a necessary condition and no feasible sufficient condition has yet been found as discussed in Yildiz \textit{et al.} (2012). We will keep using reservoir initializations and hyperparameter ranges, which have shown good results, e.g., in Pandey & Schumacher (2020) or Heyder & Schumacher (2021). We initialize the input and reservoir weights as random, i.e., \( \mathbf{W}^{in} \sim \mathcal{U} [-0.5, 0.5] \) and \( \mathbf{W}^r \sim \mathcal{U} [0, 1] \). \( \mathbf{W}^r \) is then normalized by its largest absolute eigenvalue and is subsequently scaled by \( \varrho \). Afterwards, randomly selected entries of this matrix are set to zero to assure the specified value of the reservoir density \( D \) is obtained. The specific value of each of the quantities in \( h \) is chosen by a grid search procedure which will be discussed further below.
3.2. Network training with data from DNS cases $\beta = 0.1$ and 0.3

In the following, we explore whether we can use the ESN to infer changes in the convective flow, induced by changes in the buoyancy flux at the top of the two-dimensional domain. A trained network is thus exposed to unseen data at a different physical parameter set. Such a procedure probes the generalization properties of the ESN. The subject is also connected to a transfer of the learned parameters from one task to a similar one which is known as transfer learning (Pan & Yang 2010). Due to the computationally inexpensive training scheme of ESNs, transfer learning is not often applied for this class of algorithms, even though implementations have been proposed very recently (Inubushi & Goto 2020).

Here, we take a different approach which is sketched in figure 7. A reservoir is trained with data of two different cases of buoyancy boundary conditions at $y = 1$, namely $\beta = 0.1$ and 0.3. Finally, we use this network for predicting the dynamics and statistical properties of a third convective flow with buoyancy flux parameter $\beta = 0.2$. By this choice of training data, the target flow represents an intermediate state of the two training cases $\beta = 0.1$ and $\beta = 0.3$.

We sample 700 time steps of our coarse-grained DNS data in an interval of $0.25T_f$ for both simulations of $\beta = 0.1$ and $\beta = 0.3$ in the statistically stationary regime. Also, snapshots of 700 further time steps with the same sampling interval are gathered for the target simulation ($\beta = 0.2$). The coarse-grained DNS data possesses many degrees of freedom, so that we have to introduce a preprocessing step before passing the convection data to the reservoir. For this, we decompose the buoyancy fluctuations further

$$b'(x, y, t) = \langle b' \rangle_t (x, y) + b''(x, y, t).$$  \hspace{1cm} (3.5)

Finally, we apply the Proper Orthogonal Decomposition (POD) with the methods of snapshots (Sirovich 1987; Bailon-Cuba & Schumacher 2011) on the vector $g = (u'_x, u'_y, b'')^T$, such that its $k$th component can be written as

$$g_k(x, y, t) = \sum_{i=1}^{N_{dof}} a_i(t) \Phi_i^{(k)}(x, y).$$  \hspace{1cm} (3.6)

This linear method decomposes the scalar field $g_k$ into time dependent coefficients $a_i(t)$ and spatial modes $\Phi_i^{(k)}(x, y)$, such that the truncation error is minimized. The degrees of freedom $N_{dof}$ can then be reduced, by taking only $N_{POD} \ll N_{dof}$ modes and coefficients with the most variance into account.

$$g_k(x, y, t) \approx \sum_{i=1}^{N_{POD}} a_i(t) \Phi_i^{(k)}(x, y).$$  \hspace{1cm} (3.7)

The individual and cumulative contribution of each mode can be seen in figure 8 and in table 2.

In the following we will consider the $N_{POD} = 300$ most energetic POD time coefficients as input for the ESN. The total number of degrees of freedom (dof) is thus reduced from 3 fields on a grid with size $2400 \times 150$ which results to $N_{dof} = 1.08 \times 10^6$ and thus a reduction by a factor of 3600. With this choice of cut-off mode, we capture about 80% of the original energy.

We construct the training data set for our ESN by concatenating 700 time coefficients $a(n) = (a_1(n), a_2(n), ..., a_{N_{POD}}(n))^T$ of $\beta = 0.1$ and 0.3 along the time axis, such that after 700 inputs of $\beta = 0.1$, 700 inputs of $\beta = 0.3$ are passed to the reservoir. This results to a total training length of $T_L = 1400$. During this phase the reservoir is trained to predict the respective next time instance of the POD expansion coefficients $a(n + 1)$, see again Eq.(3.4).

Once the ESN has learned to model the dynamics at both $\beta$ values, it is exposed to unseen data of the case with the intermediate parameter value, $\beta = 0.2$. For this, we initialize a new reservoir state which is preceded by 50 iterations of Eq.(3.1), where the reservoir input is given by 50 time steps of $a^{\beta=0.2}$. With this washout phase, we intend to transition to the new regime.
Figure 7: Transfer learning concept: a reservoir is trained with the combined data (concatenated along the time axis) of the POD time coefficients $a^{\beta}$ of the two cases $\beta = 0.1, 0.3$. The network learns both dynamics and computes the output weights. The reservoir is then used to infer the POD time coefficients of the intermediate flow of $\beta = 0.2$.

Figure 8: POD spectrum: a) individual contribution of each POD mode to the total energy, b) cumulative contribution. The green shaded area marks the first 300 modes. See also table 2.
| β  | 0.1 | 0.2 | 0.3 |
|----|-----|-----|-----|
| Cumulative Contribution (%) | 82.4 | 80.2 | 78.4 |

Table 2: Cumulative contribution of the first \( N_{\text{POD}} = 300 \) POD modes for the three values of the boundary condition parameter \( \beta \), which are used in our RC approach.

| \( \gamma \) | \( \lambda \) | \( N_r \) | \( D \) | \( \varphi \) |
|----|----|-----|-----|-----|
| 0.7 | 0.5 | 4096 | 0.84 | 1.96 |

Table 3: Choice of optimal hyperparameters \( h_* \). The values were chosen according to a grid search. See the appendix for more information on the grid search.

of \( \beta = 0.2 \). Starting from this state of the convective flow, the ESN will autonomously predict \( T_T = 700 \) future time steps with its learned output weights. We validate these predictions by a direct comparison with \( a^{\beta=0.2}(n) \) \((n \in [1, T_T])\) applying the mean square prediction error (MSE) which is given by

\[
\text{MSE}_{h} = \frac{1}{T_T} \sum_{n=1}^{T_T} \| \hat{y}(n) - a^{\beta=0.2}(n) \|^2,
\]

as well as the normalized average relative error (NARE) of the reconstructed fields \( u''_y, b'' \) and \( u''_y b'' \). The definition follows the work of Srinivasan et al. (2019) and is given for example for \( b'' \) by

\[
E_h \left[ \langle b'' \rangle_{x,t} \right] = \frac{1}{C_{\text{max}}} \int_0^{1} \left| \langle b'' \rangle_{x,t}^{\text{ESN}} (y) - \langle b'' \rangle_{x,t}^{\text{POD}} (y) \right| dy
\]

with

\[
C_{\text{max}} = \frac{1}{2 \max_{y \in [0,1]} \left( \langle |b''| \rangle_{x,t}^{\text{POD}} \right)}.
\]

The superscript indicates whether the field is reconstructed, see Eq. (3.7), from the \( N_{\text{POD}} \) POD time coefficients (POD) or the ESN predictions (ESN). This measure quantifies errors in the line-time average profiles of the physical fields.

Our choice of the optimal hyperparameters \( h_* \) is listed in table 3. We conducted grid searches of \( N, D, \gamma \) and \( \varphi \). See the appendix for more details. For each setting, we additionally took 100 random realizations of the same reservoir setting and computed \( \text{MSE}_{h} \) and \( E[u''_y b''] \). The final setting \( h_* \) was chosen according to the lowest third quartile of \( E[u''_y b''] \) of all 100 samples. We deliberately choose the third quartile over the median, as it assures robust reservoir outputs for different random weights \( W^m, W^r \) and therefore more reliable predictions. Furthermore, we choose the NARE of the buoyancy flux, due to its physical relevance, as opposed to the MSE. Moreover, it is comprised of two quantities which are prone to prediction errors.
3.3. Inferring the $\beta = 0.2$ case from trained echo state network

We reconstruct each component of the physical fields $u_x, u_y$ and $b$ via eq. (3.7) using the decompositions (2.17)–(2.19) and (3.5). For the validation, we use the expansion coefficients of the first $N_{\text{POD}}$ modes of the $\beta = 0.2$ data. Instantaneous snapshots in the middle of the prediction phase (the time step is $n = 350$) of the turbulent kinetic energy $E_{\text{kin}}(x, y) = 0.5(u_x^2(x, y) + u_y^2(x, y))$, the vertical velocity component $u_y(x, y)$, and the normalized buoyancy $b^*(x, y)$ can be seen in figure 9. The ground truth, i.e. the POD data, is shown for comparison. We find common features in the predicted and the validation fields. Even though some magnitudes deviate, roll patterns in the kinetic energy can be identified in the prediction case. In the velocity field component, vertical up- and downdrafts can be clearly identified. Their width and shape differs slightly from the ground truth. Moreover, the thermal boundary layer at $y = 1$ is reproduced in the predicted buoyancy field. Thermal plumes which detach primarily from the bottom wall can also be identified. It is clear that some features are not perfectly reproduced, but the qualitative picture agrees well. We also compute the normalized root mean square error (NRMSE) of these three fields at each grid point and averaged over all time steps of the test period. For example for the vertical velocity component, the NRMSE is defined as

$$\text{NRMSE}_{u_y}(x, y) = \frac{1}{\left( \sqrt{\left( u_y^{\text{POD}}(x, y) \right)^2} \right)} \sqrt{\left( \left( u_y^{(\text{ESN})}(x, y) - u_y^{(\text{POD})}(x, y) \right)^2 \right)} \right),$$

Figure 10 shows the spatial distribution of the NRMSE for the three fields of figure 9. Its volume average is additionally listed in table 4. We find that the bulk domain for $E_{\text{kin}}$ and $u_y$ shows the biggest differences. This is where the strongest up- and downflows are present. Bigger deviations are observed for the boundary layer regions of $b^*$, i.e., in the regions of the convection layer where the plumes detach or impact. We emphasize that these results were obtained for one
Figure 10: Normalized time-averaged root mean square error (NRMSE). (a) Turbulent kinetic energy $E_{kin}^{(ESN)}$. (b) Vertical velocity component $u_y^{(ESN)}$. (c) Normalized buoyancy field $b^*_{ESN}$.

|                |                         |                         |                         |
|----------------|--------------------------|--------------------------|--------------------------|
|                | $\langle\text{NRMSE}_{E_{kin}}\rangle_{x,y}$ | $\langle\text{NRMSE}_{u_y}\rangle_{x,y}$ | $\langle\text{NRMSE}_{b^*}\rangle_{x,y}$ |
|                | 1.08                     | 1.36                     | 0.35                     |

Table 4: Volume average of the normalized time-averaged root mean square error (NRMSE) at each grid point of the turbulent kinetic energy $E_{kin}$, the vertical velocity component $u_y$, and the normalized buoyancy $b^*$, see eq.(3.11).

particular realization out of the 100 reservoirs with the same hyperparameter setting, that were taken typically. Nevertheless, both results are exemplary for their setting $h_s$, as they correspond to the median NARE of the buoyancy flux. Further insights about the statistical fluctuations of the same reservoir setting will be discussed further below.

We now investigate the generalization capability of the reservoir further, by computing line-time average profiles $\langle \cdot \rangle_{x,t}$ of the fluctuations of the corresponding fields for $\beta = 0.2$. These are important parameters of simulations of large-scale turbulence. The profiles are given in figure 11. We find that in this setting $h_s$ the average reservoir produces reasonable approximations to the profiles of the true low-order statistics of the target case. The asymmetry is captured in all cases. Especially the buoyancy fluctuations are reproduced well. While the ESN reproduces the linear decrease of the convective buoyancy flux, it differs with respect to the slope and thus in the maximum and minimum value of $\langle u_y' b'' \rangle_{x,t}$.

The actual reservoir outputs in comparison to the evolution of the true time coefficients $a_t^{\beta=0.2}$ are shown figure 12(a). The ESN predictions do not match the validation data, which might be due to the ESNs lack of information about the target system. Besides the post-training washout phase with data of $\beta = 0.2$, the reservoir had only access to $\beta = 0.1$ and $\beta = 0.3$. Therefore an exact match of the time series is not expected. On the other hand, the temporal scales of the true signal are captured by the output at higher mode numbers. This can be seen in figure 12(b), which shows the corresponding Fourier power spectrum $|\mathcal{F}_{s_t}|^2$. We now follow up on the fact that the results, which where shown above, are merely one realization of a total of 100 random initialization of the same ESN with setting $h_s$. For better intuition, the error landscape over 100 different values of the spectral radius $\varrho$ is shown in figure 13. For a fixed value of $\varrho$, we observe a large variation of $E[u_y' b'']$ among the different reservoirs. See therefore figure 13(a). The first and third quartile get closer to each other as the minimum of $\varrho = 1.96$ is passed. The MSE in figure 13(b) on the other hand does show an almost linear decline with increasing spectral radius,
where the fluctuations among the reservoir realizations does not shrink as much as for the NARE.

Finally, we want to mention that a further scenario has been investigated with a similar positive outcome. That is to infer from the cases at $\beta = 0.1$ and $0.2$ to the one at $\beta = 0.3$ which can be considered as an extrapolation to a parameter outside the training range. We do not show the corresponding results to leave the manuscript compactly.

4. Conclusions and outlook

In this work, we presented a two-dimensional Boussinesq model of turbulent convection in an extended rectangular domain as a model of atmospheric convective boundary layers. We impose a buoyancy flux at that top and bottom boundaries according to typical entrainment and surface fluxes in real atmospheric convective boundary layers. The model is hence characterized by the buoyancy flux ratio $\beta$ between the top and the bottom fluxes. An increasing value of this parameter quantifies a counter-heating that stabilizes the top layer and results in negative values of the convective buoyancy flux close to the top boundary. This prescribed flux at the top mimics the warming associated with the entrainment of free tropospheric air in atmospheric convective boundary layers. Our model thus resembles properties that are absent in a standard Rayleigh-Bénard setup with uniform temperatures at the top and bottom. In particular, the top-
Figure 12: Left: Reservoir outputs $\hat{y}_i$. The true trajectories for $\beta = 0.2$ are shown in blue. Right: Power spectrum $\| \mathcal{F} \hat{y}_i \|^2$ of the corresponding modes. Here, we only show the first 300 frequencies.

Figure 13: Grid search for the spectral radius $\varphi \in [0, 2]$. The reservoir performance measured by (a) the NARE of the buoyancy flux $E[u'_i, b''_i]$ (see Eq. (3.9)) and (b) the MSE (see Eq (3.8)). The solid line represents the median error over 100 different realizations of the reservoir network while the shaded area indicates the range of first to third quartile.

down symmetry of the boundary layers is broken – in this respect it is similar to a non-Boussinesq flow.

We conducted a series of direct numerical simulations for values of $\beta$ that vary between 0 and 0.3, a range that represents midday atmospheric conditions over land. An adiabatic top boundary, i.e., zero incoming and outgoing flux ($\beta = 0$), is also considered for comparison. The four simulations reproduce well-known features of atmospheric convective boundary layers. The mean buoyancy is constant throughout the middle of the domain, which resembles the mixed layer inside the convective boundary layer. The positive buoyancy gradients at the top can be seen as approximations to the buffer zone between mixed layer and free atmosphere. Moreover, the covariance of vertical velocity and buoyancy show that the well-known linear decline with height is reproduced by our model. The four simulations also display different dynamics and convection patterns, which demonstrates the impact of the incoming top flux. These differences become evident when considering the low-order statistics of the buoyancy and its vertical flux. As
\( \beta \) increases, so does the thickness and magnitude of the stable layer at the top of the convection cell, and the intensity of the buoyancy fluctuations. This simplified model of an atmospheric boundary layer is then used as a testing bed for machine learning.

In the second part of this study, we employ a recurrent neural network in form of an echo state network to predict the dynamics and low-order statistics for the simulation run at \( \beta = 0.2 \). The echo state network is trained with two simulation data records at \( \beta = 0.1 \) and 0.3. In this way, we can explore the generalization properties of the neural network, or in other words, the performance of the machine learning algorithm to unseen data with different physical parameters. We find that the combined training of the neural network with data of the low (\( \beta = 0.1 \)) and high (\( \beta = 0.3 \)) fluxes at the top yields good approximations of the dynamics of the intermediate turbulent convective boundary layer case at \( \beta = 0.2 \). We are also able to reconstruct velocity and buoyancy fields very well. This is in line with a low-order statistics of these fields which is also reconstructed very well, for example vertical profiles of the buoyancy flux. We can conclude that for our setup, data emerging from two different constant flux boundary conditions can be used to infer at least statistical features and temporal scales of a third case with different conditions. The echo state network can thus serve as a reduced-order and scalable dynamical model that generates the appropriate turbulence statistics without solving the underlying Navier-Stokes equation of the flow.

Several directions for the future research are possible from this point. First, an extension to the three-dimensional case is desirable. This requires a stronger reduction of the data, e.g., by the application of encoder-decoder steps in form of deep convolutional neural networks in combination with spatial filtering of the direct numerical simulation data. Such a model could lead to dynamical version of a recent approach by Fonda et al. (2019) which reduced the turbulent heat transport across a convection layer to a dynamic planar network. The reservoir computing model would then operate in a low-dimensional latent space that contains the most prominent convection patterns as extracted features. Secondly, by definition neural networks are not designed to process data that live on a continuum of different lengths and times, a property which is immanent to turbulent flows. Architectures which can represent the multiscale nature of turbulence are required. Studies in these directions are currently underway and will be reported elsewhere.

Acknowledgment

This work is supported by the project "DeepTurb – Deep Learning in and of Turbulence" which is funded by the Carl Zeiss Foundation. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

Declaration of interests.

The authors report no conflict of interest.

REFERENCES

Adrian, R. J., Ferreira, R. T. D. S. & Boberg, T. 1986 Turbulent thermal convection in wide horizontal fluid layers. Exp. Fluids 4, 121–141.

Arcomano, T., Szunyogh, I., Pathak, J., Wikner, A., Hunt, B. R. & Ott, E. 2020 A machine learning-based global atmospheric forecast model. Geophys. Res. Lett. 47, e2020GL087776.

Bailon-Cuba, J. & Schumacher, J. 2011 Low-dimensional model of turbulent rayleigh-bénard convection in a cartesian cell with square domain. Phys. Fluids 23, 077101.
Brenner, M. P., Eldredge, J. D. & Freund, J. B. 2019 Perspective on machine learning for advancing fluid mechanics. *Phys. Rev. Fluids* 4, 100501.

Bruntón, S. L., Noack, B. R. & Koumoutsakos, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* 52, 477–508.

Chillà, F. & Schumacher, J. 2012 New perspectives in turbulent Rayleigh-Bénard convection. *Eur. Phys. J. E* 35, 58.

Deardorff, J. W. 1970 Convective velocity and temperature scales for the unstable planetary boundary layer and for Rayleigh convection. *J. Atmos. Sci.* 27, 1211–1213.

Edwards, J.M., Beljaars, A.C.M., Holtslag, A.A.M. & Lock, A.P. 2020 Representation of boundary-layer processes in numerical weather prediction and climate models. *Boundary-Layer Meteorol.* 177, 511–539.

Fernando, C. & Sojakka, S. 2003 Pattern recognition in a bucket. In *Advances in Artificial Life*, pp. 588–597. Springer Berlin Heidelberg.

Fodor, K., Mellado, J. P. & Wilczek, M. 2019 On the role of large-scale updrafts and downdrafts in deviations from monin–obukhov similarity theory in free convection. *Bound.-Layer Meteorol.* 172, 371–396.

Fonda, E., Pandey, A., Schumacher, J. & Sreenivasan, K. R. 2019 Deep learning in turbulent convection networks. *Proc. Natl. Acad. Sci.* 116, 8667–8672.

Fuji, K. & Nakajima, K. 2017 Harnessing disordered-ensemble quantum dynamics for machine learning. *Phys. Rev. Applied* 8, 020430.

Fuji, K. & Nakajima, K. 2020 Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices, arXiv: 2011.04890.

Gentile, P., Pritchard, M., Rasp, S., Reinaudi, G. & Yacalis, G. 2018 Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.* 45 (11), 5742–5751.

Goodfellow, I., Bengio, Y. & Courville, A. 2016 *Deep Learning*. MIT Press.

Hermans, M. & Schrauwen, B. 2010 Memory in reservoirs for high dimensional input. In *The 2010 international joint conference on neural networks (IJCNN)*, pp. 1–7.

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

Inubushi, M. & Goto, S. 2020 Transfer learning for nonlinear dynamics and its application to fluid turbulence. *Phys. Rev. E* 102, 043301.

Jaeger, H. 2001 The” echo state” approach to analysing and training recurrent neural networks - with an erratum note. *GMD-Forschungszentrum Informationstechnik Technical Report* 148.

Jaeger, H. & Haas, H. 2004 Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* 304, 78–80.

LeMone, M. A., Angevine, W. M., Bretherton, C. S., Chen, F., Dudhia, J., Fedorovich, E., Katsaros, K. B., Lenschow, D. H., Mahrt, L., Patton, E. G., Sun, J., Tienström, M. & Weil, J. 2019 100 years of progress in boundary layer meteorology. *Meteorological Monographs* 59, 1–85.

Lu, Z., Pathak, J., Hunt, B. R., Girvan, M., Brockett, R. & Ott, E. 2017 Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos* 27, 041102.

Lukoševičius, M. 2012 A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science.*, pp. 659–686. Springer Berlin Heidelberg.

Lukoševičius, M., Jaeger, H. & Schrauwen, B. 2012 Reservoir computing trends. *Künstl. Intell.* 26, 365–371.

Maass, W., Natschlager, T. & Markram, H. 2002 Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14, 2531–2560.

Moller, S., Resagk, C. & Cierpka, C. 2020 On the application of neural networks for temperature field measurements using thermochromic liquid crystals. *Exp. Fluids* 61, 111.

Nakajima, K. 2020 Physical reservoir computing: an introductory perspective. *Jpn. J. Appl. Phys.* 59 (6), 060501.

O’Gorman, P. A. & Dwyer, J. G. 2018 Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *J. Adv. Model Earth Sy.* 10 (10), 2548–2563.

Pan, S. J. & Yang, Q. 2010 A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 1345–1359.

Pandey, S. & Schumacher, J. 2020 Reservoir computing model of two-dimensional turbulent convection. *Phys. Rev. Fluids* 5, 113506.
Pandey, S., Schumacher, J. & Sreenivasan, K. R. 2020 A perspective on machine learning in turbulent flows. J. Turbul. 21 (9-10), 567–584.

Pathak, J., Lu, Z., Hunt, B. R., Girvan, M. & Ott, E. 2017 Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. Chaos 27, 121102.

Pathak, J., Wikner, A., Fusse, R., Chandra, S., Hunt, B. R., Girvan, M. & Ott, E. 2018 Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. Chaos 28, 041101.

Pawar, S. & San, O. 2021 Data assimilation empowered neural network parametrizations for subgrid processes in geophysical flows. Phys. Rev. Fluids 6, 050501.

Sirovich, L. 1987 Turbulence and the dynamics of coherent structures. Part I: Coherent structures. Q. Appl. Math. XLV, 561.

Sorbian, Z. 1996 Numerical study of penetrative and "solid lid" nonpenetrative convective boundar layers. J. Atmos. Sci. 53, 101–112.

Srinivasan, P. A., Guastoni, L., Azizpour, H., Schlatter, P. & Vinuesa, R. 2019 Predictions of turbulent shear flows using deep neural networks. Phys. Rev. Fluids 4, 054603.

Stull, R. B. 1988 An introduction to boundary layer meteorology. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Tsunegi, S., Taniguchi, T., Nakajima, K., Miwa, S., Yakusji, K., Fukushima, A., Yuasa, S. & Kubota, H. 2019 Physical reservoir computing based on spin torque oscillator with forced synchronization. Appl. Phys. Lett. 114, 164101.

Vandoorne, K., Dambré, J., Verstraeten, D., Schrauwen, B. & Bienstman, P. 2011 Parallel reservoir computing using optical amplifiers. IEEE Trans. Neural Netw. 22, 1469–1481.

Wikner, A., Pathak, J., Hunt, B. R., Girvan, M., Arcomano, T., Szunyogh, I., Pomerance, A. & Ott, E. 2020 Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems. Chaos 30, 053111.

Wyngaard, J. C. 2010 Turbulence in the atmosphere. Cambridge University Press, Cambridge, UK.

Yildiz, I. B., Jaeger, H. & Kebel, S. J. 2012 Re-visiting the echo state property. Neural Netw. 35, 1–9.

Zilitinkevich, S. S. 1991 Turbulent Penetrative Convection. Avebury Technical.