# Reduced-order modeling of two-dimensional turbulent Rayleigh-Bénard flow by hybrid quantum-classical reservoir computing

Philipp Pfeffer,[1] Florian Heyder,[1] and Jörg Schumacher[1,2]

[1]*Institut für Thermo- und Fluiddynamik, Technische Universität Ilmenau, Postfach 100565, D-98684 Ilmenau, Germany*

[2]*Tandon School of Engineering, New York University, New York City, New York 11201, USA*

(Received 5 July 2023; accepted 9 November 2023; published 13 December 2023)

Two hybrid quantum-classical reservoir computing models are presented to reproduce the low-order statistical properties of a two-dimensional turbulent Rayleigh-Bénard convection flow at a Rayleigh number $Ra = 10^5$ and Prandtl number $Pr = 10$. These properties comprise the mean vertical profiles of the root mean square velocity and temperature and the turbulent convective heat flux. The latter is composed of vertical velocity and temperature and measures the global turbulent heat transfer across the convection layer; it manifests locally in coherent hot and cold thermal plumes that rise from the bottom and fall from the top boundaries. Both quantum algorithms differ by the arrangement of the circuit layers of the quantum reservoir, in particular the entanglement layers. The second of the two quantum circuit architectures, denoted H2, enables a complete execution of the reservoir update inside the quantum circuit without the usage of external memory. Their performance is compared with that of a classical reservoir computing model. Therefore, all three models have to learn the nonlinear and chaotic dynamics of the turbulent flow at hand in a lower-dimensional latent data space which is spanned by the time-dependent expansion coefficients of the 16 most energetic proper orthogonal decomposition (POD) modes. These training data are generated by a POD snapshot analysis from direct numerical simulations of the original turbulent flow. All reservoir computing models are operated in the reconstruction or open-loop mode, i.e., they receive three POD modes as an input at each step and reconstruct the 13 missing modes. We analyze different measures of the reconstruction error in dependence on the hyperparameters which are specific for the quantum cases or shared with the classical counterpart, such as the reservoir size and the leaking rate. We show that both quantum algorithms are able to reconstruct the essential statistical properties of the turbulent convection flow successfully with similar performance compared with the classical reservoir network. Most importantly, the quantum reservoirs are by a factor of four to eight smaller in comparison with the classical case.

DOI: 10.1103/PhysRevResearch.5.043242

## I. INTRODUCTION

Quantum computing algorithms have changed our ways to process, classify, generate, and analyze data [1,2]. New ways to solve classical fluid-mechanical problems have been suggested in the form of quantum amplitude estimation algorithms [3], variational quantum algorithms [4,5], quantum lattice Boltzmann methods [6–10], or quantum linear system algorithms [11–13] for one-dimensional problems. Fluid equations for inviscid or viscous fluids have also been transformed into Schrödinger equations for specific potentials [14,15]. Following Ref. [16], the applications of quantum computing can be grouped into three major fields: (1) simulation of chemical or physical processes, (2) search and optimization, and (3) processing data with complex structure. The last field comprises quantum machine-learning methods [17], such as quantum generative methods [18,19], quantum kernel methods [20], and quantum recurrent networks, in particular in the form of quantum reservoir computing [21–23].

In classical reservoir computing, the reservoir is the central building block of the neural network architecture. The reservoir is a sparse random network of neurons that substitutes the batch of successively connected hidden layers of deep convolutional neural networks of other machine-learning algorithms [24,25]. It introduces a short-term memory to process sequential data. This is the subject of the present investigation. Here, we substitute the high-dimensional classical reservoir network with a small parametric quantum circuit in which $n$ qubits span a $2^n$-dimensional complex quantum state space for a highly entangled reservoir state to save memory and computational costs. Quantum reservoir computing can be implemented in two different ways which we describe in brief in the following.

The dynamics of an interacting many-particle quantum system—the quantum reservoir—is investigated in a so-called *analog* framework. It is characterized by a Hamiltonian $H$ subject to a unitary time evolution promoted by $U(t)$. The time evolution of the density matrix $\rho(t)$, which describes the quantum reservoir state, follows then to

$$\rho(t) = U(t)\rho(0)U^{\dagger}(t) \text{ with } U(t) = \exp\left(-\frac{i}{\hbar}Ht\right). \quad (1)$$

The operator $H$ is the (many-particle) Hamiltonian, and $U^\dagger(t)$ is the adjoint of $U(t)$. These systems have been implemented in the form of spin ensembles [26–29], circuit quantum electrodynamics [30], arrays of Rydberg atoms [31], single oscillators, and networks of oscillators [32,33]. They establish a *closed quantum system* in the ideal case that follows an ideal unitary time evolution after the input state is prepared. The pure state density matrix $\rho(t)$ is given by the outer product of the (many-particle) state vector $|\boldsymbol{\Psi}(t)\rangle$ with itself, $\rho(t) = |\boldsymbol{\Psi}(t)\rangle\langle\boldsymbol{\Psi}(t)|$.

Besides the analog framework, the *digital gate-based* framework uses parametric circuits composed of universal quantum gates. They are composed to a quantum reservoir on noisy intermediate-scale quantum devices in this case [34,35]. The reservoir state is obtained by a *repeated measurement* of the equivalently prepared quantum system and gives the probabilities $p_j(t)$ for $j = 1, \ldots, 2^n$ of the $n$-qubit quantum state $|\boldsymbol{\Psi}(t)\rangle$ to collapse on the $j$th eigenvector of the standard observable in a quantum computer, the Pauli-$Z$ matrix [36]. These probabilities correspond to the diagonal elements of the density matrix $\rho$ and are summarized to the vector $\boldsymbol{p}(t) = (p_1, \ldots, p_N)^{\mathrm{T}} = (\rho_{11}, \ldots, \rho_{NN})^{\mathrm{T}}$ with $N = 2^n$. They can be read out by a measurement. Again, we assumed that the initial state $\rho(0)$ is a pure state. In Ref. [37], the digital gate-based approach was realized in the form of an *open quantum system* which implies that parts of the short-term memory of the hybrid systems are kept outside the quantum reservoir. This method allowed us to model the dynamics of low-dimensional nonlinear systems, such as the Lorenz 63 [38] and its eight-dimensional Lorenz-type model extension [39] on an actual IBM quantum computer. This algorithm will be denoted as hybrid algorithm 1, in short H1, in the following.

In the present work, we advance our investigation on quantum reservoir computing with a proof-of-concept application to a realistic, complex fluid mechanical problem. We seek to show that it is possible to achieve results for the following low-dimensional simulations which are comparable to classical reservoir computing. To this end, we present a full data processing pipeline for a two-dimensional *turbulent* Rayleigh-Bénard convection flow [40] which contains a quantum computing module—the quantum reservoir. This flow is a paradigm for turbulence that is driven by buoyancy forces in many geophysical and astrophysical processes [41,42]. The hybrid quantum-classical machine-learning algorithm will serve as a data-driven reduced-order model of the turbulent flow without knowledge of the underlying mathematical equations of motion.

The hybrid nature of the quantum machine-learning model includes a reduction of the high-dimensional turbulence data to a low-dimensional latent space. This is done by a classical snapshot-based proper orthogonal decomposition (POD) [43]. Similar to classical machine-learning algorithms, this encoding-decoding step is necessary for a fully turbulent flow since the dimension of the classical input data is high; the actual number of degrees of freedom is here $\tilde{N}_{\mathrm{dof}} = 3 \times 384 \times 96 = 110\,592$, see also Refs. [44,45]. The quantum machine-learning algorithm thus operates in a latent data space of $N_{\mathrm{dof}} = 16$ in the present case and is able to reproduce relevant large-scale features and low-order statistics of the turbulent flow, such as the vertical profile of the mean convective heat flux across the convection layer. Particularly, the latter point is of particular interest in the present application.

We also extend our previous study with an improved hybrid quantum-classical reservoir computing model (RCM) which integrates more parts of the algorithm into the quantum computing part in comparison with the previous algorithm H1 from Ref. [37]. The present work is a first step away from the traditional von Neumann architecture, in which computation and memory are located in distinct components. The new algorithm will be denoted H2 in the following. It will be compared in detail with a classical RCM, in short C, and H1, our previous approach.

The hybrid nature of our algorithm implies additionally that the optimization of the reservoir output layer is performed classically by a direct solution of the minimization task. The full data processing pipeline of the algorithm comprising a combined POD-RCM model is sketched in Fig. 1. The figure sketches the classical and quantum reservoir in Figs. 1(b) and 1(c). The quantum reservoir builds on a low-qubit-number parametric quantum circuit which spans a high-dimensional reservoir state space based on a highly entangled $n$-qubit quantum state.

The paper is organized as follows: In Sec. II, we present the turbulent flow and in Sec. III the reduction to the low-dimensional latent space in which the quantum reservoir operates. Section IV follows with a compact presentation of the algorithms C, H1, and H2. Section V discusses the results in dependence on hyperparameters of all three reservoir computing models. Moreover, we also compare different error measures. One is adapted to the specific fluid mechanical application. A summary and an outlook are given in the last section.

## II. TURBULENT FLOW

### A. Model equations and parameters

We consider a two-dimensional Rayleigh-Bénard system where a fluid is enclosed between two impermeable plates with constant temperature difference $\Delta T = T_{\mathrm{bot}} - T_{\mathrm{top}} > 0$ [40]. The Boussinesq equations connect the incompressible velocity field $\boldsymbol{u} = (u_x, u_z)$ with the temperature and pressure fields $T$ and $\mathcal{P}$. They are given by

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

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = -\boldsymbol{\nabla} P + \nu \nabla^2 \boldsymbol{u} + g\alpha(T - T_0)\boldsymbol{e}_z, \tag{3}$$

$$\frac{\partial T}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})T = \kappa \nabla^2 T. \tag{4}$$

Here, $\alpha$, $g$, $\nu$, and $\kappa$ are the thermal-expansion coefficient, the acceleration due to gravity, the kinematic viscosity, and the thermal diffusivity, respectively. We set $P = \mathcal{P}/\rho_0$ in Eq. (3). Furthermore, $T_0$ and $\rho_0$ are *constant* reference values of the temperature and mass density, respectively. These equations stand for the differential balances of mass density (2), momentum density (3), and energy density (4) of a fluid parcel. In the Boussinesq approximation, it is assumed that the fluid is incompressible (or divergence-free) and that small deviations of the density of the fluid from the reference values

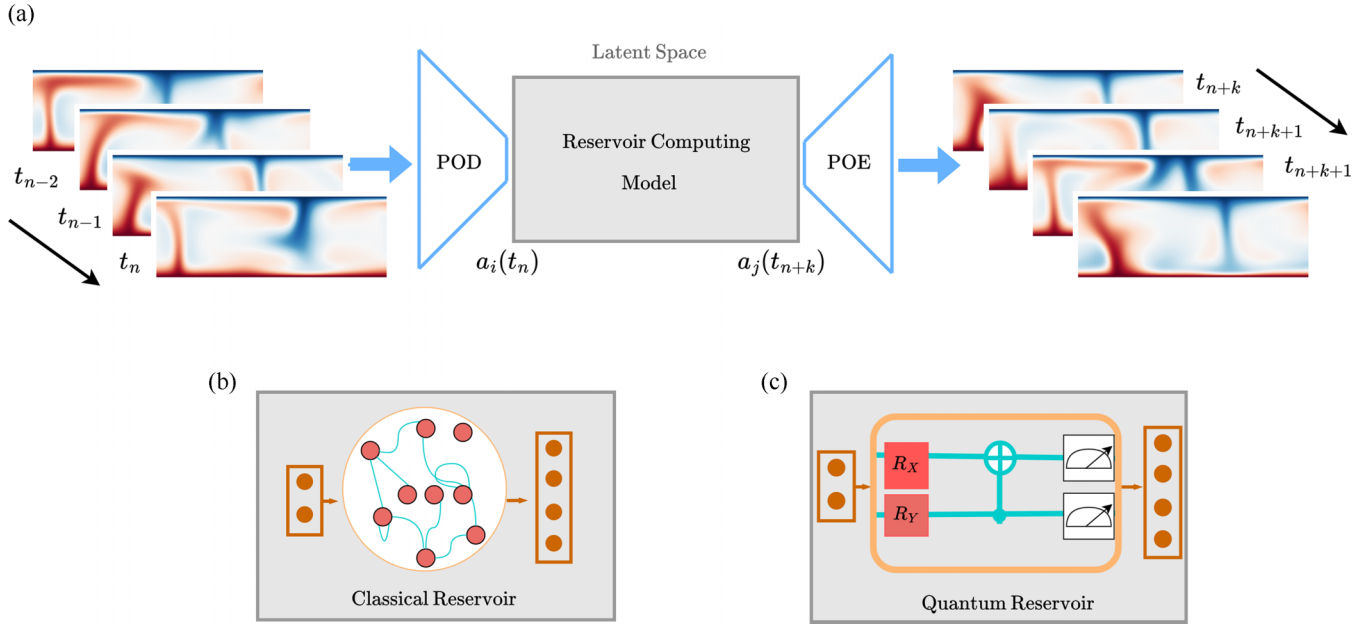


FIG. 1. Pipeline of data processing in the classical and hybrid quantum-classical reservoir computing models. (a) Sketch of the whole algorithm. Direct numerical simulation snapshots of the turbulent convection flow to the left are encoded in a low-dimensional latent data space by a proper orthogonal decomposition (POD). A reservoir computing model operates in the latent space receiving the input sequences $a_i(t_n)$ with $i = 1, \ldots, N_{\text{in}} < N_{\text{out}}$. Its output is a sequence $a_j(t_{n+k})$ with $j = 1, \ldots, N_{\text{dof}}$; typically one takes $k = 1$. The output is finally decoded by a proper orthogonal expansion (POE) to simulation data at later times and/or a subsequent statistical analysis. The reservoir takes two different forms in this work. (b) The classical reservoir, which we use for comparison, consists of a sparse random network of neurons that possesses a dynamic memory for temporal data processing. (c) The quantum reservoir is a digital gate-based quantum circuit. In this work, two different quantum algorithms are employed for the quantum reservoir. Both classical and quantum reservoir contain input and output layers to the left and right, respectively. The reservoir architectures are further detailed in Sec. IV.

depend linearly on temperature deviations [40]. This leads to the last term in the momentum equation (3) which couples the temperature field with the vertical velocity component.

The system is made dimensionless by the choice of the free-fall velocity scale $U_f = \sqrt{\alpha g \Delta T H}$, the free-fall timescale $T_f = H/U_f$, and $\Delta T$. Here, $H$ is the height of the convection layer, the characteristic spatial scale in the setting. In this way all material parameters and scales can be summarized in dimensionless parameters that determine the operating point of the turbulent flow. These parameters are the Rayleigh and the Prandtl numbers, $Ra = \alpha g \Delta T H^3/(\nu\kappa)$ and $Pr = \nu/\kappa$, respectively. They take values of $\text{Ra} = 10^5$ and $\text{Pr} = 10$ in the present proof-of-concept study. Alternatively, one can use $U_{\text{diff}} = \kappa/H$ as a characteristic velocity [37,38]. This does not affect the physical outcome. Figure 2 illustrates the configuration which we want to investigate in the following by the hybrid quantum-classical algorithm.

We conduct direct numerical simulations using the spectral element solver Nek5000 [46] to solve the Rayleigh-Bénard system (2)–(4) in a domain $A = L \times H$ with aspect ratio $\Gamma = L/H = 2\sqrt{2}$. Dimensionless coordinates are thus $x \in [0, 2\sqrt{2}]$ and $z \in [0, 1]$. Dirichlet boundary conditions are imposed for the temperature field at top and bottom, $T(z = 0) = 1$ and $T(z = 1) = 0$. Furthermore, we choose free-slip boundary conditions for the velocity field in the $z$ direction, $u_z(z = 0, 1) = 0$ and $\partial u_x/\partial z = 0$ at $z = 0, 1$. Periodic boundaries for all fields are taken in the horizontal $x$ direction. The chosen boundary conditions, aspect ratio, and Prandtl number correspond to a popular standard case for the Lorenz systems [37], but at a higher $Ra$ and thus fully turbulent in contrast with our previous work.

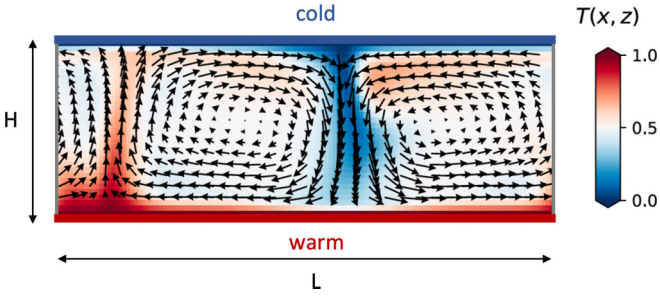


FIG. 2. Two-dimensional Rayleigh-Bénard convection flow. The fluid with a kinematic viscosity $\nu$ and temperature diffusivity $\kappa$ is confined between two impermeable boundary planes at the top ($z = H$) and bottom ($z = 0$). The bottom plane has a uniform temperature $T_{\text{bot}}$; the top one a uniform temperature $T_{\text{top}} < T_{\text{bot}}$. The fluid layer has a length $L = 2\sqrt{2}$ and a height $H = 1$, such that the aspect ratio $\Gamma = L/H = 2\sqrt{2}$ results. The figure displays a snapshot of the turbulent flow. Background contours stand for the temperature field $T$; the vectors illustrate the two-component velocity field $(u_x, u_z)$. The turbulent heat transfer across the layer from the bottom to the top is manifest by the thin plume structures which are visible: hot lighter fluid (in red) rises to the top while cold heavier fluid (in blue) falls downwards. Since the velocity field is divergence-free a circulation roll pattern develops

### B. Turbulent fluctuations and heat transfer

A central physical question in convection flows is related to the turbulent transfer of heat and momentum across the convection layer in dependence on the two dimensionless parameters, the Rayleigh number $Ra$ and the Prandtl number $Pr$. In response to both, these transfers can be summarized in two further dimensionless parameters, the Nusselt number $Nu$ for the turbulent heat transfer and the Reynolds number $Re$ for the turbulent momentum transfer. They are given by

$$Nu = \frac{\langle u_z T\rangle_{x,t} - \kappa \frac{\partial \langle T\rangle_{x,t}}{\partial z}}{\frac{\kappa \Delta T}{H}} \text{ and } Re = \frac{u_{\mathrm{rms}} H}{\nu}. \tag{5}$$

The symbol $\langle\cdot\rangle_{x,t}$ stands for a combined average with respect to the horizontal $x$ direction with $x \in [0, \Gamma]$ and time $t$. The root mean square (rms) velocity is given by $u_{\mathrm{rms}} = (\langle u_x^2 + u_z^2\rangle_{A,t})^{1/2}$ where the average is now a combination of averages with respect to the simulation plane $A$ and time $t$. The two terms in the definition of the Nusselt number stand for two heat currents across the layer, the convective and the diffusive one. Their sum is constant and equal to $Nu$ for each height $z \in [0, 1]$. However, their contributions to the total turbulent heat flux differ with respect to $z$, caused for example by the boundary condition $u_z = 0$ at the bottom and top.

It is exactly the mean profile of the convective heat flux $\langle u_z T\rangle_{x,t}$ as a function of $z$, which we want to obtain as a reservoir computing model output. The same holds for the vertical profiles of the root mean square velocity and temperature. This is the low-order statistics of the turbulent flow which is generated by the different reservoir computing models without knowledge of the nonlinear Boussinesq equations of motion. We come back to these results in Sec. V D.

## III. DATA REDUCTION TO LATENT SPACE

The numerical simulations are performed on a nonuniform grid of size of $384 \times 96$ points with a second-order equidistant time stepping of $\Delta t = 5 \times 10^{-4}$. For the analysis, the simulation data were interpolated to a uniform grid of size $N_x \times N_z = 128 \times 32$. The dataset consists of a sequence of snapshots of the fields $\boldsymbol{u}(x, z, t_m)$ and $T(x, z, t_m)$ with $m = 1, 2, \ldots, 10^4$; they are equidistant in time with $\tau = 0.25H/U_f$. The sequence covers the statistically stationary regime of the turbulent convection flow. The three turbulent fields possess an input vector of size 12 288. To circumvent high computational effort for the machine learning model, we add a preprocessing step.

Here, we apply a POD in the form of a snapshot method [43,47,48]. It is a linear method, where the data reduction is realized by a truncation to a set of Galerkin modes. For this, we decompose the physical fields into time mean and fluctuations,

$$u_x(x, z, t) = \langle u_x\rangle_t(x, z) + u_x'(x, z, t), \tag{6}$$

$$u_z(x, z, t) = \langle u_z\rangle_t(x, z) + u_z'(x, z, t), \tag{7}$$

$$T(x, z, t) = \langle T\rangle_t(x, z) + \theta'(x, z, t). \tag{8}$$

Finally, we perform the snapshot POD to the fluctuating component fields $g_k$ into time-dependent coefficients $a_i(t)$ and spatial modes $\Phi_i^{(\mathrm{k})}(x, z)$, such that the truncation error is minimized. The degrees of freedom $\tilde{N}_{\mathrm{dof}}$ can then be reduced, by taking only $N_{\mathrm{dof}} \ll \tilde{N}_{\mathrm{dof}}$ modes and coefficients with the most variance into account,

$$g_k(x, z, t) \approx \sum_{i=1}^{N_{\mathrm{dof}}} a_i(t)\Phi_i^{(\mathrm{k})}(x, z), \tag{9}$$

with $g_k = \{u_x', u_z', \theta'\}$ and thus $k = 1, 2, 3$. Figures 3(a)–3(f) compare the reconstruction of the temperature and velocity fields from 3 and 16 modes with the original simulation data at a time instant. The time series of the expansion coefficients of the three first POD modes, $a_1(t)$ to $a_3(t)$, will be fed into the recurrent network in the reconstruction phase after training.

In the following, we use the more compact notation $\boldsymbol{a}^t = (a_1(t), a_2(t), \ldots, a_M(t))$ with $M = N_{\mathrm{dof}}$ and the discrete snapshot time superscript $t$. This is our dynamical system state which has to be learned by the reservoir computing model. We choose the cutoff at $N_{\mathrm{dof}} = 16$ and capture 87% of the variance of the original fields as seen in Fig. 3(g). It also implies that the notation $\rho(t)$ changes to $\rho^t$ and so on, see again Eq. (1).

## IV. RESERVOIR COMPUTING FRAMEWORKS

A reservoir computing model [24,25] is one realization of recurrent machine learning besides long short-term memory networks or gated recurrent units [49,50]. Its fundamental element is the reservoir state vector $\boldsymbol{r}^t \in \mathbb{R}^{N_{\mathrm{res}}}$, which evolves from $t$ to $t + 1$ by a certain update equation characterizing the approach. As indicated in the introduction, this work compares three RCMs, algorithm C, a classical RCM [44,45,48,51–53] with linear memory and nonlinear activation, see Eq. (10), algorithm H1, a hybrid quantum-classical RCM [37] with classical linear memory and nonlinear quantum dynamics, see Eq. (12), and algorithm H2, a full quantum RCM which induces memory by reduced gate parametrization, see Eq. (14). Figure 4 illustrates the different quantum circuits of H1 (left) and H2 (right). All algorithms run here in the open-loop or reconstruction mode [54], where the reservoir receives $\boldsymbol{a}_{\mathrm{in}}^t = (a_1^t, a_2^t, a_3^t)$ at each time step and outputs the missing expansion coefficients at the next time step, $\hat{\boldsymbol{a}}_{\mathrm{out}}^{t+1} = (\hat{a}_4^{t+1}, \ldots, \hat{a}_{N_{\mathrm{dof}}}^{t+1})$ with $N_{\mathrm{dof}} = 16$ for our study. The hat symbol always identifies the network prediction in the following. Note that in this scenario the reservoir receives the coefficients corresponding to the large-scale fluid motion while returning the small-scale features of the higher modes.

### A. Classical reservoir computing algorithm

The classical algorithm C is characterized by the following iterative equation for the reservoir state vector $\boldsymbol{r}_C$ [45,48,55],

$$\boldsymbol{r}_{\mathrm{C}}^{t+1} = (1 - \varepsilon)\boldsymbol{r}_{\mathrm{C}}^t + \varepsilon \tanh\left[W^{\mathrm{in}}\boldsymbol{a}_{\mathrm{in}}^t + W^{\mathrm{res}}\boldsymbol{r}_{\mathrm{C}}^t\right]. \tag{10}$$

where $W^{\mathrm{res}} \in \mathbb{R}^{N_{\mathrm{res}} \times N_{\mathrm{res}}}$ is the sparsely occupied random reservoir matrix and $W^{\mathrm{in}} \in \mathbb{R}^{N_{\mathrm{res}} \times N_{\mathrm{in}}}$ is the random input matrix. The scalar $\varepsilon \in [0, 1]$ is called leaking rate because it scales the influence of the first memory term and the second dynamical term. This discrete time stepping from snapshot $t$ to $t + 1$ comprises external forcing by the inputs $\boldsymbol{a}_{\mathrm{in}}^t$ as well as a

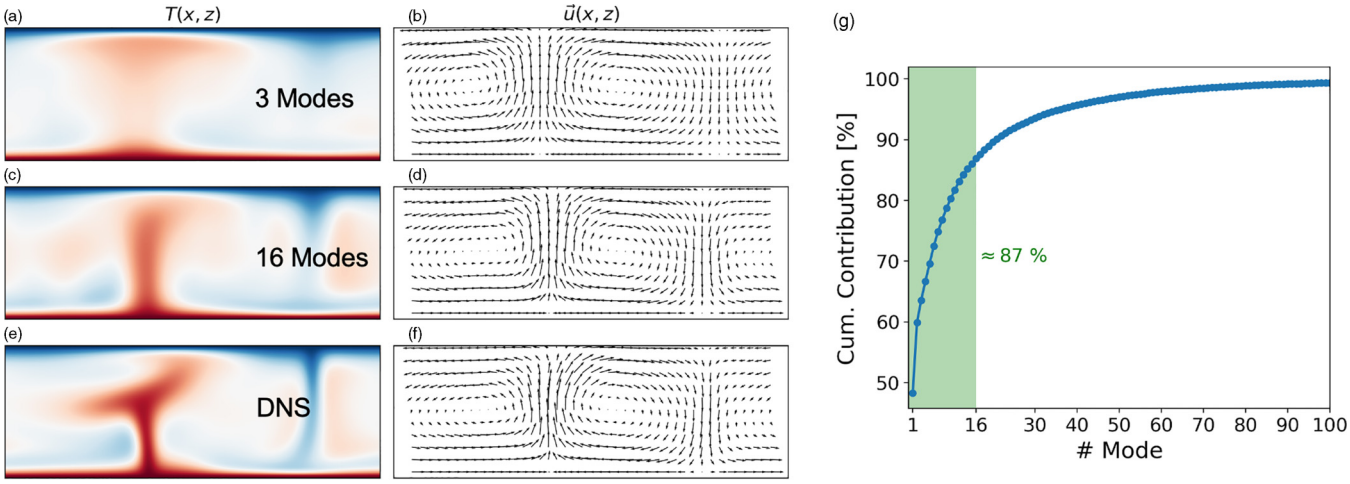


FIG. 3. Snapshot of two-dimensional convection flow. Panel (a), (c), and (e) show the temperature field $T(\boldsymbol{x}, t_0)$, panels (b), (d), and (f) the corresponding velocity vector field $\boldsymbol{u}(\boldsymbol{x}, t_0) = (u_x(x, z, t_0), u_z(x, z, t_0))$. Panels (a) and (b) display the reconstruction by means of the first three POD modes, panels (c) and (d) of the first 16 POD modes, and panels (e) and (f) display the full snapshot. (g) Cumulative spectrum of the proper orthogonal decomposition. The shaded region indicates the chosen cutoff mode at $N_{\text{dof}} = 16$. The first three POD modes include 63% of the total energy, the first 16 modes include 87%.

self-interaction with the reservoir state $\boldsymbol{r}_{\text{C}}^t$. The hyperbolic tangent $\tanh(\cdot)$ is the nonlinear activation function, which is applied to the elements of its argument vector. The reservoir matrix $W^{\text{res}}$ is specified further by the reservoir density $D$, its sparsity, and the spectral radius $\varrho(W^{\text{res}})$, the magnitude of the maximum eigenvalue. The hyperparameters of the classical RCM are thereby $N_{\text{res}}$, $\varepsilon$, $D$, and $\varrho(W^{\text{res}})$. In all reservoir computing models, the optimization is done directly for the output layer only, which is represented by the matrix $W^{\text{out}*} \in \mathbb{R}^{N_{\text{out}} \times N_{\text{res}}}$. The asterisk stands for the optimized matrix after training, see Refs. [37,48] for the details. Also there, the Tikhonov parameter $\gamma$ is explained further, which is used as a prefactor of an additional penalty term in the cost function. This hyperparameter is only applied in the classical RCM.

To give an explicit example for the one-step reconstruction mode in which the model will be used here: the classical RCM utilizes Eq. (10) to calculate $\boldsymbol{r}_{\text{C}}^{t+1}$ from $\boldsymbol{r}_{\text{C}}^t$ and $\boldsymbol{a}_{\text{in}}^t$ with its three

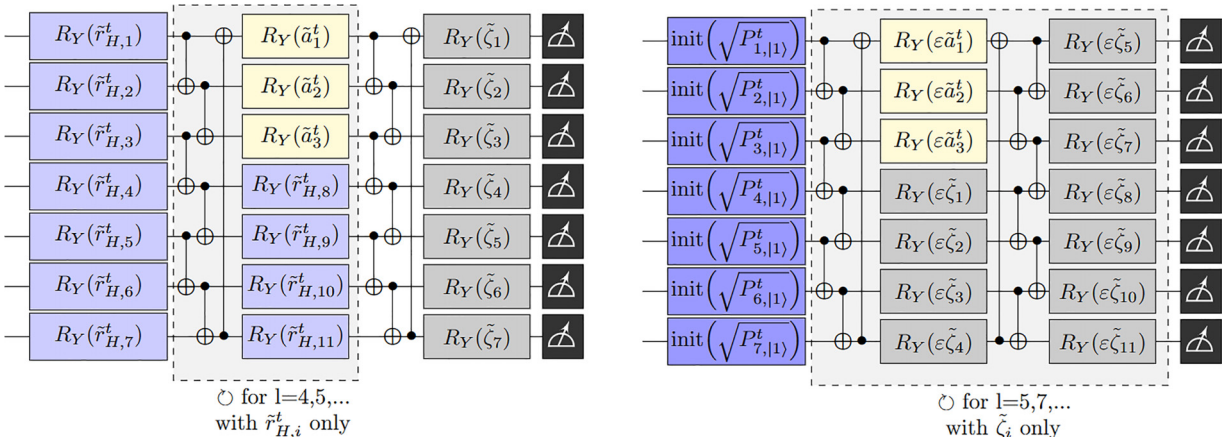


FIG. 4. Quantum circuits of the two applied hybrid quantum classical algorithms which represent the quantum computing version of the reservoir. (left) Hybrid quantum-classical reservoir computing algorithm H1, which represents our previous approach of Ref. [37] with a new hyperparameter $l$ to denote the circuit complexity (or depth). (right) Hybrid quantum-classical reservoir computing algorithm H2 which we in short denote to as the full quantum reservoir computing model. The new structure incorporates an efficient pseudo-initialization of the last reservoir time step as well as an efficient memory implementation by the leaking rate $\varepsilon$, which now scales the dynamical activation of the approach without classical postprocessing. In both cases, every qubit of the quantum register starts in the basis state $|0\rangle$ to the left. Each quantum circuit is shown for seven qubits and $l = 3$ layers of $R_Y$ rotation gate at a discrete time step $t$. The CNOT gates between the $R_Y$ gate connect neighboring qubits and generate entanglement. Since this pairwise two-qubit operation is done successively over all qubits, one ends up with a fully entangled seven-qubit quantum state. The tilde symbol indicates that the rotation gate arguments are rescaled to cover the respective range, as given in the text. Also, in the right circuit diagram $\text{init}(\omega) = R_Y(2\arccos(\omega))$. The last column of symbols (to the right) in both panels stands for the measurement to read out the reservoir state.

components to obtain the estimate

$$\hat{\boldsymbol{a}}_{\text{out}}^{t+1} = W^{\text{out}*}\boldsymbol{r}_{\text{C}}^{t+1}, \tag{11}$$

with $\hat{\boldsymbol{a}}_{\text{out}}^{t+1} = (\hat{a}_4^{t+1}, \ldots, \hat{a}_{16}^{t+1})$ in the present turbulent convection flow study.

### B. Hybrid quantum-classical reservoir computing algorithm 1

The algorithm H1 follows the classical reservoir iteration procedure (10) closely, but the update equation receives another dynamical part. We introduced H1 in Ref. [37]. It is given by

$$\boldsymbol{r}_{\text{H1}}^{t+1} = (1-\varepsilon)\boldsymbol{r}_{\text{H1}}^{t} + \varepsilon \boldsymbol{p}^{t+1}, \tag{12a}$$

with

$$p_j^{t+1} = \rho_{jj}^{t+1} = \left|\langle e_j | U\left(\boldsymbol{\zeta}, \boldsymbol{a}_{\text{in}}^{t}, \boldsymbol{r}_{\text{H1}}^{t}\right)|e_1\rangle\right|^2, \tag{12b}$$

where $|e_j\rangle$ is the $j$th basis vector of the $n$-qubit Hilbert space $\mathbb{C}^{\otimes n}$ with a dimension $N = 2^n$. Furthermore, $|e_1\rangle = |0\rangle^{\otimes n}$ is the $n$-qubit basis vector for which every of the $n$ individual qubits is in the base state $|0\rangle$. The quantum circuit $U(\boldsymbol{\zeta}, \boldsymbol{a}_{\text{in}}^{t}, \boldsymbol{r}_{\text{H1}}^{t})$ is parametrized with random values $\boldsymbol{\zeta}$, the current input $\boldsymbol{a}_{\text{in}}^{t}$ and the last reservoir state $\boldsymbol{r}_{\text{H1}}^{t}$. We initialize $\boldsymbol{r}_{\text{H1}}^{0}$ as a uniform-amplitude probability vector at all entries, which guarantees that $\boldsymbol{r}_{\text{H1}}^{t}$ will remain a probability vector for all times $t$.

The structure of the circuit is a repeated pattern of $R_Y$ gates and separating CNOT-gate layers, where the arguments of the $R_Y$ gates are the only difference between the layers. The circuit always starts and ends with an $R_Y$-gate layer, as shown in Fig. 4 (left). As a new hyperparameter, we introduce the depth $l$ as the amount of $R_Y$-gate layers inside the circuit. An $R_Y$-gate layer applies an $R_Y$-gate on every qubit, where the arguments are scaled versions of the aforementioned variables. The random values $\boldsymbol{\zeta}$ vary in $[0, 4\pi]$, the probabilities $\boldsymbol{r}_{\text{H1}}^{t}$ are multiplied by $\pi$ to vary in $[0, \pi]$ and the inputs $\boldsymbol{a}_{\text{in}}^{t}$ are rescaled to vary in $[-\pi, \pi]$. For example, the random values $\boldsymbol{\zeta}$ can build various unitary matrices, as the matrix definition of the $R_Y(\varphi)$ gate is given by

$$R_Y(\varphi) = \begin{pmatrix} \cos(\varphi/2) & -\sin(\varphi/2) \\ \sin(\varphi/2) & \cos(\varphi/2) \end{pmatrix}. \tag{13}$$

Note that the loading of the reservoir state and the dynamical system modes by rotations into H1 and H2 introduces a nonlinearity in the (linear) quantum dynamics.

The CNOT-gate layer uses $n$ CNOT gates such that every qubit is control and target qubit once. The specific arrangement is random but fixed. We always assure that it is impossible to separate any subgroup of qubits from the remaining one; see Ref. [37] on the importance of the entanglement for the reconstruction quality measured by the mean squared error. The sorting of the $R_Y$ gate arguments is such that the last layer is filled with random values, all other layers receive probabilities, and the first three entries in the second layer are always the inputs. There are too many possibilities to prove that this specific circuit construction is the optimum, but we tested many architectures and chose the described one due to its best overall performance.

### C. Hybrid quantum-classical reservoir computing algorithm 2

The new hybrid algorithm H2 is modified such that the complete execution on a quantum computer is enabled. It is given by

$$\boldsymbol{r}_{\text{H2}}^{t+1} = \boldsymbol{p}^{t+1}, \tag{14a}$$

with

$$p_j^{t+1} = \rho_{jj}^{t+1} = \left|\langle e_j | U\left(\varepsilon\boldsymbol{\zeta}, \varepsilon\boldsymbol{a}_{\text{in}}^{t}\right)|\hat{\boldsymbol{r}}_{\text{H2}}^{t}\rangle\right|^2. \tag{14b}$$

An identity transformation is imposed by

$$U(0,0) = \mathcal{I}_N. \tag{15}$$

This is a consequence of the inclusion of the leaking rate $\varepsilon$ in the arguments for H2, the major difference to H1. Furthermore, the approximate reservoir vector is

$$|\hat{\boldsymbol{r}}_{\text{H2}}^{t}\rangle = \bigotimes_{i=1}^{n}\left[\sqrt{1-P_{i,|1\rangle}^{t}}|0\rangle + \sqrt{P_{i,|1\rangle}^{t}}|1\rangle\right], \tag{16}$$

where $P_{i,|1\rangle}^{t}$ denotes the probability of measuring qubit $i$ of the whole $n$-qubit register in basis state $|1\rangle$ at reservoir time step $t$. In other words, the two probabilities, $1 - P_{i,|1\rangle}^{t}$ and $P_{i,|1\rangle}^{t}$ are the diagonal elements of the $2 \times 2$ density matrix $\tilde{\rho}_i^t$ which is obtained by the following partial trace of the original density matrix of the $n$-qubit state $\rho^t$, which traces out all qubits except qubit $i$. It is given by

$$\tilde{\rho}_i^t = \text{Tr}_{1,2,\ldots,i-1,i+1,\ldots,n}(\rho^t). \tag{17}$$

In a quantum algorithm, this is realized by an individual measurement at qubit $i$ of the whole quantum register only. We thus structure the circuit such that it is initialized by a completely separable approximation of the last reservoir state and thus ease the reservoir initialization at each step, see Refs. [23,56] for alternative solutions to circumvent this bottleneck in analog quantum reservoir computing. This combines two advantages: It is a minimal initialization for the quantum circuit with only $n$ operations which contains the integral information on the reservoir. Combined with the adapted definition of the unitary $U$ in Eq. (14), we include the previously external memory in the quantum circuit iteration step. That is, the dynamics will correspond to an identity transformation, see Eq. (15), once the leaking rate is $\varepsilon = 0$.

Here, the structure of the quantum circuit starts with the preparation of $|\hat{\boldsymbol{r}}_{\text{H2}}^{t}\rangle$, which comes down to an $R_Y$-gate layer parametrized with the $P_{i,|1\rangle}^{t}$, as done in upcoming Eq. (23). The following circuit layer has pairwise CNOT-gate layers, where every second CNOT layer is the inverse of the first CNOT layer, as indicated in the right panel of Fig. 4. Thereby, we satisfy the central condition of Eq. (15), as $R_Y(0) = \mathcal{I}_2$. Beside the input gates, all $R_Y$ gates are filled with random values $\zeta_k$. Further details of the circuit are adopted from H1. Note however, that the circuit ends with a CNOT layer for an even number of $R_Y$ layers.

### D. Advantages of hybrid reservoir computing approaches

There are several aspects which motivate the presented hybrid quantum-classical approaches H1 and particularly H2. First, we want to investigate whether the nonlinear activation of the quantum circuits is preferable over $\tanh(\cdot)$ of Eq. (10).

The inherent nonlinearity of product concatenations from the unitary gates to larger matrices approximates the inherent nonlinearity of the original flow problem, see the second term on the left-hand side of Eq. (3).

Second, we note that for online processing of the data, we can strongly reduce the required memory once the output of a time step is generated. For instance, one needs to save the used probabilities only, i.e., for H1 the $r^t_{H,i}$ and for H2 the $P^t_{i,|1\rangle}$, to recompute the reservoir state at the subsequent time step again. This would not be possible for the classical reservoir. Especially for H2, this always means that for a reservoir of size $N$, one keeps $\log_2(N)$ values only to store sufficient information of the reservoir evolution. If necessary, it is then possible to reconstruct all $2^n$ reservoir states in parallel.

Third, H2 is in contrast with H1 fully executable on a quantum computer, as the leakage rate of the reservoir step is included in the rotation gate parameters of the quantum circuit. Therefore, no external memory is required, cf. Eq. (12a) for H1. Additionally, it might be possible to extend the present scheme H2 to a multistepping approach on the quantum computer if one implements the output weights as well. This could be done by encoding the output on an ancilla qubit to either measure it directly or insert it back for the next time step via a rotation controlled by this ancilla qubit. Future work in this direction might elaborate whether this approach can strongly reduce the computational complexity in comparison to the classical approach by reducing the number of sampling steps (or shots) for multiple time steps in the hybrid case.

## V. COMPARISON OF THE MODELS

### A. Error quantification measures

To evaluate the reconstruction quality of the proposed RCMs, we need appropriate error measures. The first standard is the root mean squared error of the prediction. Since multiple modes have to be reconstructed, we take the normalized error with respect to each mode and combine the $N_{\text{out}} = 13$ individual errors of the reconstructed modes to the normalized root mean squared error (NRMS). This results in

$$E_{\text{NRMS}} = \frac{1}{N_{\text{out}}\mathcal{T}} \sum_{j=N_{\text{in}}+1}^{N_{\text{dof}}} \left[ \frac{\sqrt{\sum_{t=t_a}^{t_b} \left(a^t_j - \hat{a}^t_j\right)^2}}{\max_t\left(a^t_j\right) - \min_t\left(a^t_j\right)} \right], \tag{18}$$

where $\mathcal{T} = t_b - t_a$ is the length of the testing phase (measured in discrete time steps as discussed in Sec. III). The maxima and minima are determined with respect to the time interval $[t_a, t_b]$.

A second popular approach is the correlation error, also known as the coefficient of determination, or R2-score, which computes the correlation between the original and predicted modes [23]. Here, we average the square of correlation, such that the correlation error is given by

$$E_{\text{corr}} = 1 - \frac{1}{N_{\text{out}}} \sum_{j=N_{\text{in}}+1}^{N_{\text{dof}}} \left( \frac{\text{cov}\left(a^t_j, \hat{a}^t_j\right)}{\sigma\left(a^t_j\right)\sigma\left(\hat{a}^t_j\right)} \right)^2. \tag{19}$$

Here, $\sigma(\cdot)$ is the standard deviation and $\text{cov}(\cdot,\cdot)$ the covariance of the arguments. Note that by the square of the correlation, we value anticorrelation as much as positive correlation, thus $0 \leqslant E_{\text{corr}} \leqslant 1$. Strongly correlated time series send $E_{\text{corr}} \to 0$. Both error measures are applicable to any dynamical system. However, in the present work, we consider a turbulent flow; the RCM application is focused to reconstructed statistical properties, as motivated in the introduction and in Sec. II. Such properties can be the mean vertical profiles of the velocity components or the temperature field.

Therefore, we define an additional measure which is directly related to the low-order statistical reconstruction results, the normalized average relative error (NARE), which has been used in classical RCM applications [48,57] and is given by the $L_1$ norm,

$$E_{\text{NARE}} = \frac{1}{C_{\text{max}}} \int_0^1 \left| \langle u_z T\rangle_{x,t} - \langle \hat{u}_z \hat{T}\rangle_{x,t} \right| dz, \tag{20}$$

with the normalization constant

$$C_{\text{max}} = 2 \max_{z\in[0,1]} |\langle u_z T\rangle_{x,t}|. \tag{21}$$

Here, $\hat{u}_z$ and $\hat{T}$ are the reconstructed flow values which are obtained by the proper orthogonal expansion (POE) from the modes $\hat{a}^t_j$. As the convective heat flux is prone to the error of two fields; it is a suited measure of the accordance of the inferred convection flow. We compare these three errors for the hybrid quantum RCMs H1 and H2 in Fig. 5 as a function of the reservoir size controlled by the qubit number $n$ and for different amplitude encoding methods, which will be detailed in the next section.

First, it can be observed that $E_{\text{corr}}$ is relatively large with a minimum for eight to nine qubits. The reason is that the accurate reconstruction of the POD modes is difficult as frequent deviations of the time series $\hat{a}^t_i$ from the ground truth are inevitable in this higher-dimensional, turbulent flow problem. In contrast with low-dimensional dynamical systems, such as the Lorenz model, a reservoir computing model will not reconstruct the exact systems trajectory in the high-dimensional phase space with a sampling time step of $0.25t_f$, but generate a trajectory which gives the right low-order statistics. This is even the case for the classical RCMs [45,48,52,57]. Thus it is not appropriate to optimize the network on the basis of $E_{\text{corr}}$. Particularly weak correlations are not directly linked to the dynamical quality of the reconstruction.

Second, we observe from the figure that both $E_{\text{NRMS}}$ and $E_{\text{NARE}}$ grow eventually with a large qubit number $n$. However, it can be observed that particularly for eight, nine, and ten qubits, the physical error $E_{\text{NARE}}$ improves by almost one order of magnitude while $E_{\text{NRMS}}$ remains relatively constant and insensitive. We thus conclude that while the accuracy of the reconstructed modes is difficult to improve further, the physical properties of the flow are more sensitive. Thereby, for most of the remaining analysis, we continue our RCM evaluation with $E_{\text{NARE}}$ only because it is the physically relevant measure for the fluid-mechanical application of the quantum algorithm.

### B. Different amplitude encodings of classical data

Besides the hyperparameters, which will be discussed further below, the quantum circuits need to encode the classical

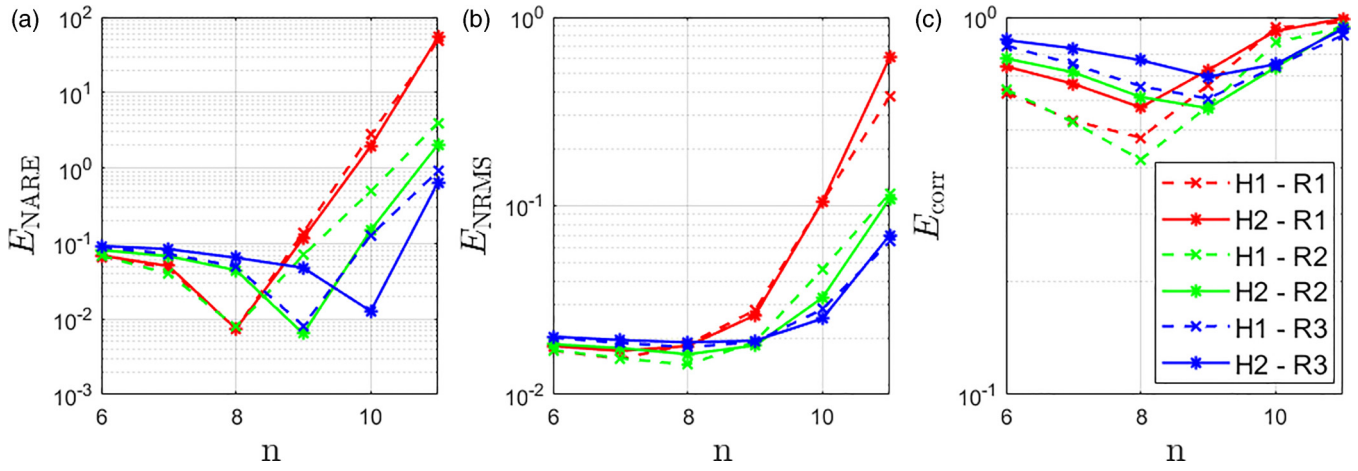


FIG. 5. Comparison of three error measures, which are defined in Eqs. (18)–(20) and three amplitude encoding methods, which are defined in Eqs. (22)–(24) for both hybrid RCM cases, H1 and H2. They are termed $R_1$, $R_2$, and $R_3$. Here, each data point is the median obtained from ten different random reservoir seeds $\boldsymbol{\zeta}$. The hyperparameters $l$ and $\varepsilon$ are optimal in each case. For the optimization, $2 \leqslant l \leqslant 7$ and $0.05 \leqslant \varepsilon \leqslant 1$. We trained for 5000 and evaluated on the subsequent 500 discrete time steps.

input data $a_j^t$. This can be realized in different ways. We discuss the following three amplitude encoding methods:

$$R_1 := R_Y\left[2\cos^{-1}\left(\tilde{a}_j^t\right)\right] = \begin{pmatrix} \tilde{a}_j^t & -\sqrt{1-\tilde{a}_j^{t2}} \\ \sqrt{1-\tilde{a}_j^{t2}} & \tilde{a}_j^t \end{pmatrix}, \tag{22}$$

$$R_2 := R_Y\left[2\cos^{-1}\left(\sqrt{\tilde{a}_j^t}\right)\right] = \begin{pmatrix} \sqrt{\tilde{a}_j^t} & -\sqrt{1-\tilde{a}_j^t} \\ \sqrt{1-\tilde{a}_j^t} & \sqrt{\tilde{a}_j^t} \end{pmatrix}, \tag{23}$$

$$R_3 := R_Y\left[2\pi\tilde{a}_j^t\right] = \begin{pmatrix} \cos\left(\pi\tilde{a}_j^t\right) & -\sin\left(\pi\tilde{a}_j^t\right) \\ \sin\left(\pi\tilde{a}_j^t\right) & \cos\left(\pi\tilde{a}_j^t\right) \end{pmatrix}. \tag{24}$$

The tilde symbol in the equations indicates, again, that the input mode $a_j^t$ needs to be rescaled such that it only varies in the interval $[-1, 1]$. Encoding $R_1$ ensures that $\tilde{a}_j^t$ is the component of the corresponding qubit, while encoding $R_2$ reveals $\tilde{a}_j^t$ after a measurement. Encoding $R_3$ is a natural encoding inside the $R_Y$ gate, where we only rescale the input to harness the largest but still unique range of the trigonometric functions.

Each approach induces a specific nonlinear characteristic and the superiority of each encoding may change for different learning tasks. We come back to Fig. 5 where we plotted all three error measures versus qubit number $n$ for $R_1$ to $R_3$. The error measure $E_{\mathrm{NRMS}}$ shows an approximate independence on the specific encoding scheme for $n \leqslant 9$ for both H1 and H2. Only for the larger qubit numbers $R_3$ performs best. In the case of $E_{\mathrm{NARE}}$ a local minimum can be observed for each encoding scheme. All three amplitude encodings have their optimum at a magnitude of approximately $10^{-2}$. This is obtained for $R_1$ and $R_2$ already at a smaller number of eight qubits. Again, for the lower reservoir dimensions of $n = 6$ and $n = 7$ all errors are of the same order of magnitude. As already said, for the largest reservoir dimensions, the error increases, similar to $E_{\mathrm{NRMS}}$, for $n \geqslant 10$. In conclusion, we do not fix the specific amplitude encoding for the following analysis, but use it as a further degree of freedom to be optimized.

### C. Hyperparameter dependence

Table I summarizes all hyperparameters that appear either in the classical or in the hybrid quantum-classical RCMs. The hyperparameters, which exist in the quantum case only, are the circuit layer depth $l$ and the type of amplitude encoding. The latter was already discussed in the past section. Joint parameters of the classical and quantum case, which will be studied in the following, are the reservoir size $N_{\mathrm{res}}$ and the leaking rate $\varepsilon$. We also mention at this point that all studies for H1 and H2 are conducted with the IBM Qiskit statevector simulator where the measurements are not subject to shot noise [58].

We train all cases for 5000 output time steps and validate the trained network on the subsequent 500 output time steps.

#### *1. Quantum circuit layer depth*

The quantum circuits H1 and H2 can be parametrized by the number of $R_Y$ layers $l$ and the positions of the CNOT gates.

TABLE I. List of hyperparameters for the classical and both hybrid quantum-classical algorithms. The number of used qubits follows by $n = \log_2(N_{\mathrm{res}})$ in H1 and H2. The hyperparameters, which are not plotted, in the figures are always pre-optimized; that is, we take the optimal values such that the single-best representation is illustrated.

| Algorithm | Hyperparameter | Symbol |
|---|---|---|
| Classical | Reservoir density | $D$ |
| | Spectral radius | $\rho(W^{\mathrm{res}})$ |
| | Tikhonov parameter | $\gamma$ |
| Quantum (H1,H2) | Circuit layer depth | $l$ |
| | Input encoding | $R_i$ |
| Joint | Reservoir size | $N_{\mathrm{res}}$ |
| | Leaking rate | $\varepsilon$ |

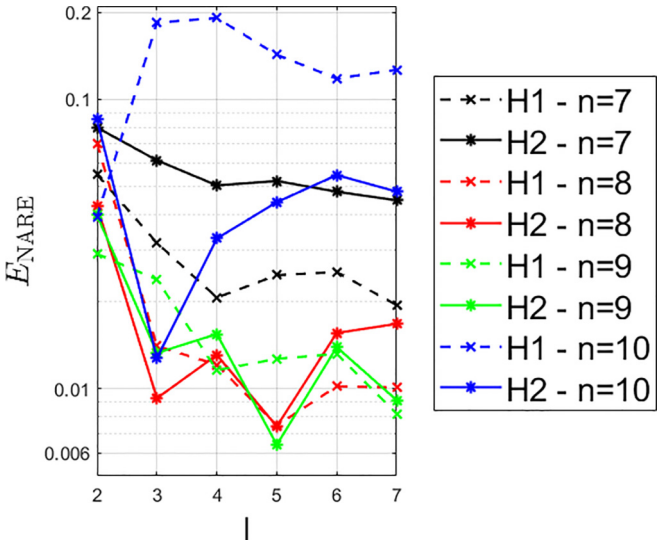


FIG. 6. Evaluation of the NARE as a function of circuit depth $l$ for the hybrid quantum cases H1 and H2. Each data point is the median of $E_{\mathrm{NARE}}$, obtained from ten random reservoir seeds, using the optimal leaking rate $\varepsilon$ and optimal input amplitude encoding. In nearly all cases, the optimal values are $\varepsilon = 0.2$ and $R_1$ for the input encoding.

The latter aspect showed no significant influence during our analysis; that is, the performance of both algorithms is relatively constant as long as all qubits are connected, as described in Sec. IV B. Meanwhile, the depth of the circuit is of interest because it is directly proportional to the computational effort of the approach, i.e., the number of operations, as well as the realizability on real quantum computers. Figure 6 displays the dependence of $E_{\mathrm{NARE}}$ on the layer depth for H1 and H2. We collect the results for $n = 7$, 8, 9, and 10. Except for $n = 10$, the overall trend of the error $E_{\mathrm{NARE}}$ is a decrease with growing circuit depth $l$. For $n = 10$, we observe a clear advantage of H2 compared with H1. For $n = 8$ and $n = 9$, the cases H1 and H2 perform similarly well, almost at their optimal error measure. We finally mention that the median was taken over a rather small number of different reservoir realizations.

#### 2. *Reservoir size and leaking rate*

In Fig. 7, we show the median of $E_{\mathrm{NARE}}$ in dependence on the leaking rate $\varepsilon$ for different reservoir sizes $N_{\mathrm{res}} = 2^n$. For all points, we averaged here over 100 seeds for C and 10 seeds for H1 and H2, while all other parameters are pre-optimized; that is, we choose the optimal spectral radius $\rho(W^{\mathrm{res}})$ and the Tikhonov parameter $\gamma$ in the classical RCM case, the optimal input encoding $R_i$ and the amount of layers $l$ in the hybrid quantum-classical cases. We choose this optimum such that the single-best median is illustrated for the respective approach and reservoir size. We observe that H1 and H2 seem to outperform the classical approach for qubit numbers $n < 10$. The global optimum, i.e., the minimal amplitudes of $E_{\mathrm{NARE}}$, are obtained for the new architecture H2 at $n = 9$, although the other RCMs can perform similarly well if the reservoir is large enough.

### D. Statistical analysis of the reconstructed fields

Of particular relevance in turbulent convection is the mean vertical convective heat flux profile, which is the one-point correlation of the vertical velocity component and the temperature field, $\langle u_z T(z)\rangle_{x,t}$, see Sec. II B. It is a measure of the amount of heat transported by fluid motion from the bottom of the layer to the top of the layer. Such a vertical profile is more difficult to reconstruct, as it combines the statistics of two reconstructed fields. In addition, we monitor the root mean square profiles of the velocity components and the temperature. These are essential low-order statistical properties of the flow at hand. They are also important when the turbulence cannot be modeled down to the smallest physically relevant scale and has to be parametrized. This is the case for subgrid-scale parametrizations in global circulation models, e.g., in atmospheric turbulence [41].

We illustrate in Fig. 8 the single best reconstruction of each RCM model combined with the mean statistical profiles of the

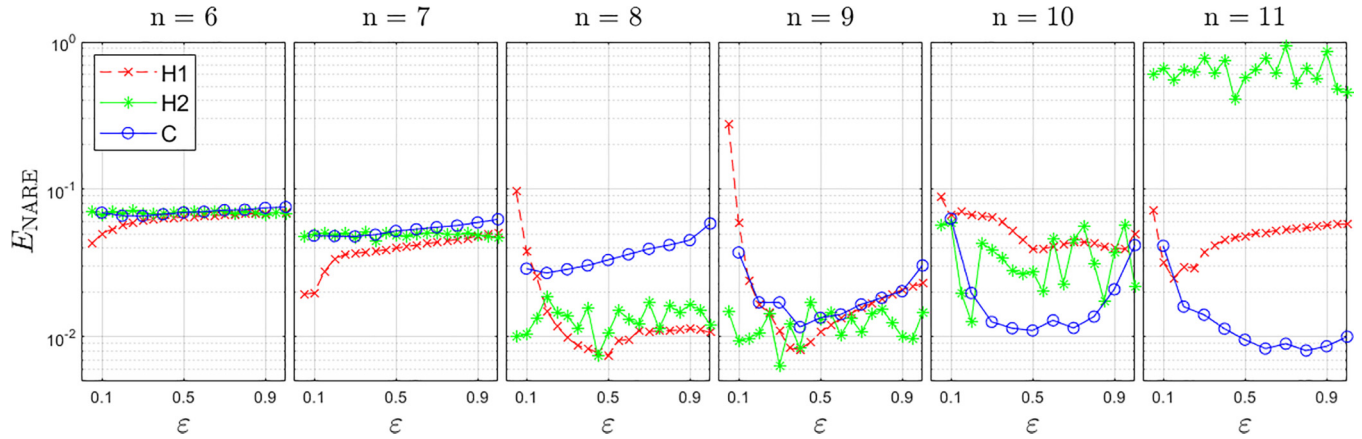


FIG. 7. Comparison of the error $E_{\mathrm{NARE}}$ for the convective heat flux $\langle u_z T\rangle_{x,t}$ versus the leaking rate $\varepsilon$. We show the results for the three reservoir computing approaches C, H1, and H2 and for different reservoir sizes $2^n$. Displayed are the median values for 10 seeds with H1 and H2 and 100 seeds for $C$. The optimum of each curve is the single-best median for the respective approach at the given reservoir size. Note that the best classical reservoir performance is achieved at substantially larger reservoir sizes ($n = 11$) as compared with the quantum algorithms ($n = 8$ for H1 and $n = 9$ for H2).

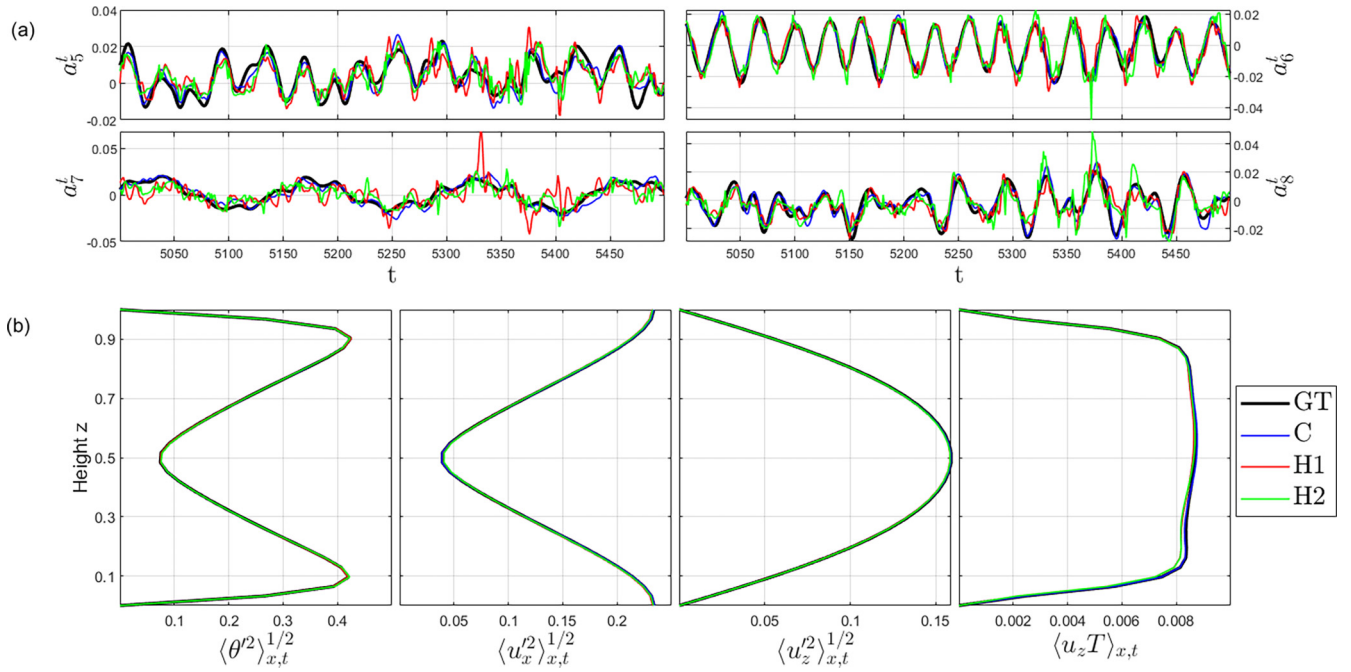


FIG. 8. Single-best reservoir computing results for $N_{\text{in}} = 3$. We compare the ground truth (GT) with a classical reservoir computing model (C) with $N_{\text{res}} = 2048$ and the hybrid quantum-classical quantum algorithms (H1, H2) with $N_{\text{res}} = 512$ and six layers in both cases. All networks output 500 time steps. Panels (a) display the time series of modes $a_5(t)$ to $a_8(t)$. Panels (b) show vertical mean profiles of essential low-order statistical measures. These are the thermal variance $\langle \theta'^2 \rangle_{x,t}$, the horizontal velocity fluctuations $\langle u_x'^2 \rangle_{x,t}^{1/2}$, the vertical velocity fluctuations $\langle u_z'^2 \rangle_{x,t}^{1/2}$, and the convective heat flux $\langle u_z T \rangle_{x,t}$.

flow. We evaluated the previous grid search over the hyperparameters for the single best results. Figure 8(a) displays the reconstructed time series of four POD expansion coefficients from C, H1, and H2 in comparison to the ground truth (GT). It is seen that the curves are not followed exactly, but that the overall trends and thus the low-order statistics are represented well. Note that this is not only for the case for H1 and H2 but also for C. Illustrated are the best cases for the physical error measure $E_{\text{NARE}}$. In other words, we optimized for the lower part of the figure.

The mean convective heat flux profile in the rightmost panel of Fig. 8(b) is the most sensitive, which is the reason why we utilized it for the hyperparameter optimization. This statistical correlation is connected to the hot rising and the cold falling plumes which are visible in Fig. 3(e). Figure 9 displays finally the spatial reconstruction of velocity and heat flux fields for all three cases. We find that both quantum algorithms, as well as the classical counterpart, produce statistical profiles that are in good accordance with the ground truth, for both, the root mean square profiles of the velocity components and temperature as well as for the convective heat flux. This demonstrates the applicability of the hybrid quantum-classical reservoir computing model as a reduced-order model in combination with the encoder-decoder module in the form of POD or POE, respectively.

## VI. FINAL DISCUSSION

Two central objectives can be given for the present proof-of-concept study. First, we wanted to extend the application of hybrid quantum-classical reservoir computing algorithms towards more complex classical dynamical systems. Starting with the well-known Lorenz 63 benchmark case and its extension to eight degrees of freedom in Ref. [37], we increased here the complexity of the task to be learned further by proceeding to a *turbulent* convection flow at the same geometry and Prandtl number as in the Lorenz cases, but at a significantly higher Rayleigh number (the latter of which measures the driving of the flow by buoyancy forces). We integrated the quantum circuit therefore into a combined encoder-decoder–reservoir computing pipeline which has to be used as well when classical machine learning is applied to turbulent flows [44], see again Fig. 1. Since the phase space of the Rayleigh-Bénard system is higher-dimensional and thus the dimensionality of the turbulent attractor, the reservoir computing algorithm is only able to predict low-order turbulence statistics rather than exactly following a specific dynamical systems trajectory for a longer time. But this is exactly the task that we had in mind, reproducing second-order statistics, such as the convective heat flux, in a data-driven model without solving the full nonlinear partial differential equation system of Rayleigh-Bénard convection.

The second objective is related to the modified architecture of H2 in comparison to H1. We have compared both hybrid quantum-classical algorithms with respect to various hyperparameters and found that they mostly perform equally well for the reconstruction tasks. Nonetheless, the update of the circuit architecture H2 can be evaluated completely on a quantum computer, which enables further steps of the hybrid algorithm on the quantum device and avoids additional external memory as H1. Additionally, the simulation with the circuit architecture H2 can be realized more efficiently

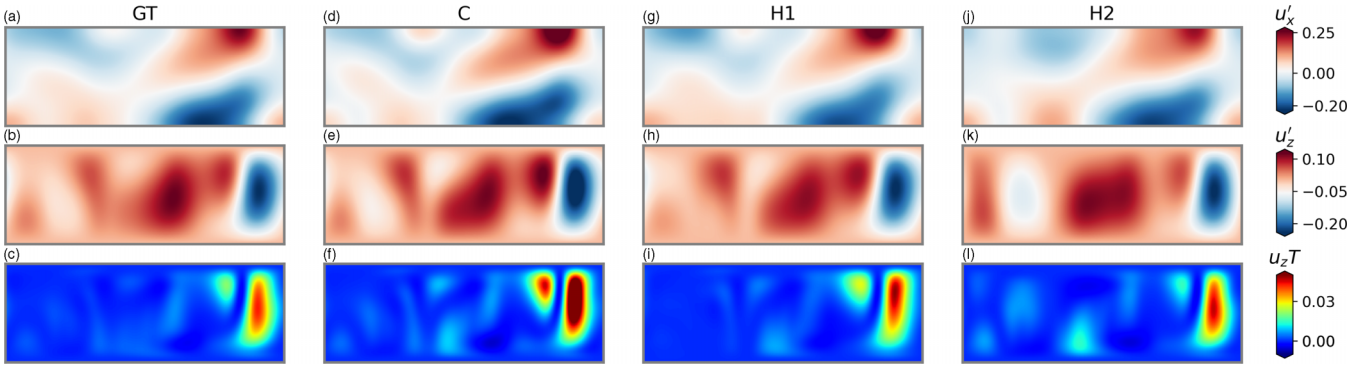


FIG. 9. POD reconstruction based on the latent space variables in Fig. 8. We show the horizontal and vertical velocity fluctuation fields, $u_x'$ in panels (a), (d), (g), and (j) and $u_z'$ in panels (b), (e), (h), and (k). Furthermore, the local convective heat flux $u_zT$ is displayed in panels (c), (f), (i), and (l). The fields of the classical reservoir algorithm reconstruction are shown in panels (d)–(f), while the quantum algorithm results for H1 and H2 are shown in panels (g)–(i) and (j)–(l), respectively. For comparison, the ground truth, i.e., the POD model, is shown in panels (a)–(c). It can be seen that both quantum reservoirs, as well as the classical counterpart, reproduce well the spatial features of all three fields of the ground truth (GT).

than the one for H1 once the layer depth is $l > 3$. The reason is that in H2 every operation that follows the input encoding can be summarized to one precomputable matrix which acts on the time-dependent inputs. This is not possible for the architecture H1, as further subsequent circuit layers also have to be filled with the time-dependent components of the probability amplitude vector $\boldsymbol{p}^t$. We also investigated, if a random value encoding which is used in H2 works for the original hybrid algorithm H1, but it was found that this method strongly impairs the performance of H1. Therefore it can be concluded that H2 is a more efficient implementation of the reduced-order model of the turbulent convection flow by means of a hybrid quantum-classical reservoir computing algorithm; it is comparable to the best-case scenarios of the classical reservoir computing approaches which however need at least twice as large reservoirs in the present case, see Fig. 7. In future applications, this could reduce the numerical effort for both, hyperparameter optimization and production runs.

The demonstration of the capabilities and potential of the present framework, but also its current limitations, has to our point of view its value for following studies in this subject in the future applied to realistic fluid flow problems.

A first open point for our future work is to solve the sampling problem. We used the ideal Qiskit statevector simulator for both algorithms H1 and H2, which circumvents the crucial problem of approximating the necessary probabilities [58]. A deeper analysis of this aspect shows that the computational overhead to approximate the probabilities, both, for the best cases of H1 and H2, is big. In detail, more than $2^{20}$ samples (or shots) are necessary for comparable results. This seems to damp the prospects for an application on current noisy intermediate-scale quantum devices. However, a repetition of the hyperparameter grid search with sample-based probabilities and the additional implementation of weak measurements as in Ref. [22] might ease this problem. Furthermore, it has to be evaluated if the hybrid reservoir computing approach can be further scaled up to more vigorous turbulence, i.e., flows at higher Rayleigh number. This would imply a higher dimension of the latent data space and possibly a different encoding-decoding scheme, see Refs. [44,45] for the classical cases. These investigations are going on and will be reported elsewhere.

## ACKNOWLEDGMENTS

This work is supported by the Project no. P2018-02-001 "DeepTurb—Deep Learning in and of Turbulence" of the Carl Zeiss Foundation and by the Deutsche Forschungsgemeinschaft under Grant no. DFG-SPP 1881. We acknowledge support for the publication costs by the Open Access Publication Fund of the Technische Universität Ilmenau.

## APPENDIX: QUANTUM COMPUTING BASICS

In this Appendix, we summarize some basic definitions of quantum computing. Further details can be found in the textbook of Nielsen and Chuang [36] or a review by Bharadwaj and Sreenivasan [11]. While a single classical bit can take two discrete values only, namely {0, 1}, a single quantum bit (in short, qubit) is a superposition of the two basis states of the vector (or better Hilbert) space $\mathbb{C}^2$. This is sometimes illustrated as an arbitrary point on the surface of the so-called *Bloch sphere* (a unit sphere). One writes

$$|q_1\rangle = c_1|0\rangle + c_2|1\rangle = c_1\begin{pmatrix}1\\0\end{pmatrix} + c_2\begin{pmatrix}0\\1\end{pmatrix}, \tag{A1}$$

with $c_1, c_2 \in \mathbb{C}$ and $(|c_1|^2 + |c_2|^2)^{1/2} = 1$. Vectors $|0\rangle$ and $|1\rangle$ are the basis vectors in Dirac notation [36]. A qubit can be considered as the simplest quantum system. In other words, the qubit can be consequently found in infinitely many superposition states, all the points that fill the surface of the unit sphere. It is the building block of an $n$-qubit system, also denoted as an $n$-qubit quantum register. They are formed by successive tensor products of qubits. For example, a two-qubit state vector is the tensor product of two single-qubit vectors,

$$|q_1\rangle \otimes |q_1'\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2. \tag{A2}$$

The basis of this tensor product space is given by four vectors: $|\boldsymbol{j}_1\rangle = |0\rangle \otimes |0\rangle$, $|\boldsymbol{j}_2\rangle = |0\rangle \otimes |1\rangle$, $|\boldsymbol{j}_3\rangle = |1\rangle \otimes |0\rangle$, and $|\boldsymbol{j}_4\rangle = |1\rangle \otimes |1\rangle$. An $n$-qubit quantum state $|\boldsymbol{\Psi}\rangle$ is

TABLE II. Logical table of CNOT gate.

| Input | | Output | |
|---|---|---|---|
| Control $\|q_0\rangle$ | Target $\|q_1\rangle$ | Control $\|q_0\rangle$ | Target $\|q_1\rangle$ |
| $\|0\rangle$ | $\|0\rangle$ | $\|0\rangle$ | $\|0\rangle$ |
| $\|0\rangle$ | $\|1\rangle$ | $\|0\rangle$ | $\|1\rangle$ |
| $\|1\rangle$ | $\|0\rangle$ | $\|1\rangle$ | $\|1\rangle$ |
| $\|1\rangle$ | $\|1\rangle$ | $\|1\rangle$ | $\|0\rangle$ |

consequently an element of a $2^n$-dimensional tensor product Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$. The state vector is given by

$$|\Psi\rangle = \sum_{k=1}^{2^n} c_k |\boldsymbol{j}_k\rangle \text{ with } \sum_{k=1}^{2^n} |c_k|^2 = 1. \tag{A3}$$

An $n$-qubit state vector is called *fully separable* if it can be written as a tensor product,

$$|\Psi\rangle = \bigotimes_{i=1}^{n} |q_i\rangle, \tag{A4}$$

where $|q_i\rangle$ are single qubit quantum states given by Eq. (A1). It is called *separable* if a tensor product decomposition of $|\Psi\rangle$ into blocks is possible with at least one multiqubit quantum state $|q_i\rangle$ that is not fully separable. Multiqubit quantum states which are not separable are called *entangled*. An $n$-qubit quantum state is called *fully entangled* if no subspace of separable qubits exists. Fully entangled quantum states are characterized by correlations which do not classically exist. The entanglement is a unique property of quantum computing; it is supposed to be responsible for the quantum advantage of some quantum algorithms with respect to their classical counterparts, such as prime factorization [36], as single qubit operations act nontrivially on a large quantum state and produce global parallel processing by a single computational step.

The time evolution of a quantum state is described by a unitary transformation,

$$|\Psi\rangle(t) = U(t)|\Psi\rangle(0) \text{ with } U(t)^{-1} = U(t)^{\dagger}. \tag{A5}$$

Elementary unitary transformations are supplied by *quantum gates*, for example, rotations of a single qubit. Note that they are reversible transformations. The $R_Y$ rotation gate is defined by Eq. (13) in Sec. IV B. A second central gate is the controlled NOT gate (in short CNOT) which connects two qubits. It represents a flip of the target qubit once the control qubit is in state $|1\rangle$. The logical table of the two-qubit CNOT gate is shown in Table II, which can be transformed into a $4 \times 4$ unitary matrix. Rotation and CNOT gates are elementary gates which are composed to *quantum circuits* that are required for the input of the classical data into a quantum algorithm as well as for the unitary evolution of the same. The quantum version of the reservoir is composed of exactly these gates. The readout of information is done by a measurement process which causes the collapse of the $n$-qubit quantum state.


[1] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[2] I. H. Deutsch, Harnessing the power of the second quantum revolution, PRX Quantum **1**, 020101 (2020).

[3] F. Gaitan, Finding flows in a Navier-Stokes fluid through quantum computing, npj Quantum Inf. **6**, 61 (2021).

[4] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, Variational quantum algorithms for nonlinear problems, Phys. Rev. A **101**, 010301(R) (2020).

[5] O. Kyriienko, A. E. Paine, and V. E. Elfving, Solving nonlinear differential equations with differentiable quantum circuits, Phys. Rev. A **103**, 052416 (2021).

[6] B. N. Todorova and R. de Steijl, Quantum algorithm for the collisionless Boltzmann equation, J. Comput. Phys. **409**, 109347 (2020).

[7] L. Budinski, Quantum algorithm for the advection–diffusion equation simulated with the lattice Boltzmann method, Quantum Inf. Process. **20**, 57 (2021).

[8] M. A. Schalkers and M. Möller, Efficient and fail-safe collisionless quantum Boltzmann method, arXiv:2211.14269.

[9] X. Li, X. Yin, N. Wiebe, J. Chun, G. K. Schenter, M. S. Cheung, and J. Mülmenstädt, Potential quantum advantage for simulation of fluid dynamics, arXiv:2303.16550.

[10] W. Itani, K. R. Sreenivasan, and S. Succi, Quantum algorithm for lattice Boltzmann (QALB) simulation of incompressible fluids with a nonlinear collision term, arXiv:2304.05915.

[11] S. S. Bharadwaj and K. R. Sreenivasan, Quantum computation of fluid dynamics, Indian Acad. Sci. Conf. Ser. **3**, 77 (2020).

[12] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, Efficient quantum algorithm for dissipative nonlinear differential equations, Proc. Natl. Acad. Sci. USA **118**, e2026805118 (2021).

[13] S. S. Bharadwaj and K. R. Sreenivasan, Hybrid quantum algorithms for flow problems, Proc. Natl. Acad. Sci. USA **120**, e2311014120 (2023).

[14] S. Succi and A. Tiribocchi, Navier-Stokes-Schrödinger equation for dissipative fluids, arXiv:2308.05879.

[15] Z. Meng and Y. Yang, Quantum computing of fluid dynamics using the hydrodynamic Schrödinger equation, Phys. Rev. Res. **5**, 033182 (2023).

[16] F. F. Flöther, The state of quantum computing applications in health and medicine, Research Directions: Quantum Technologies **1**, e10 (2023).

[17] J. Biamonte, P. Wittek, N. Pancotti, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[18] S. Lloyd and C. Weedbrook, Quantum generative adversarial learning, Phys. Rev. Lett. **121**, 040502 (2018).

[19] M. S. Rudolph, N. B. Toussaint, A. Katabarwa, S. Johri, B. Peropadre, and A. Perdomo-Ortiz, Generation of high-resolution handwritten digits with an ion-trap quantum computer, Phys. Rev. X **12**, 031010 (2022).

[20] M. Schuld and N. Killoran, Quantum machine learning in feature Hilbert spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[21] D. Marković and J. Grollier, Quantum neuromorphic computing, Appl. Phys. Lett. **117**, 150501 (2020).

[22] P. Mujal, R. Martínez-Peña, J. Nokkala, J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Opportunities in quantum reservoir computing and extreme learning machines, Adv. Quantum Technol. **4**, 2100027 (2021).

[23] P. Mujal, R. Martínez-Peña, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Time-series quantum reservoir computing with weak and projective measurements, npj Quantum Inf. **9**, 16 (2023).

[24] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks - with an erratum note, GMD-Forschungszentrum Informationstechnik Technical Report 148 (2001), pp. 1–48.

[25] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science **304**, 78 (2004).

[26] K. Fujii and K. Nakajima, Harnessing disordered-ensemble quantum dynamics for machine learning, Phys. Rev. Appl. **8**, 024030 (2017).

[27] K. Nakajima, K. Fujii, M. Negoro, K. Mitarai, and M. Kitagawa, Boosting computational power through spatial multiplexing in quantum reservoir computing, Phys. Rev. Appl. **11**, 034021 (2019).

[28] A. Kutvonen, K. Fujii, and T. Sagawa, Optimizing a quantum reservoir computer for time series prediction, Sci. Rep. **10**, 14687 (2020).

[29] A. Sakurai, M. P. Estarellas, W. J. Munro, and K. Nemoto, Quantum extreme reservoir computation utilising scale-free networks, Phys. Rev. Appl. **17**, 064044 (2022).

[30] G. Angelatos, S. A. Khan, and H. E. Türeci, Reservoir computing approach to quantum state measurement, Phys. Rev. X **11**, 041062 (2021).

[31] R. Araiza Bravo, N. Khadijeh, X. Gao, and S. F. Yelin, Quantum reservoir computing using arrays of Rydberg atoms, PRX Quantum **3**, 030325 (2022).

[32] J. Nokkala, R. Martínez-Peña, G. L. Giorgi, V. Parigi, M. C. Soriano, and R. Zambrini, Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing, Commun. Phys. **4**, 53 (2021).

[33] L. C. G. Govia, G. J. Ribeill, G. E. Rowlands, H. K. Krovi, and T. A. Ohki, Quantum reservoir computing with a single nonlinear oscillator, Phys. Rev. Res. **3**, 013077 (2021).

[34] J. Chen, H. I. Nurdin, and N. Yamamoto, Temporal information processing on noisy quantum computers, Phys. Rev. Appl. **14**, 024065 (2020).

[35] Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto, Natural quantum reservoir computing for temporal information processing, Sci. Rep. **12**, 1353 (2022).

[36] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).

[37] P. Pfeffer, F. Heyder, and J. Schumacher, Hybrid quantum-classical reservoir computing of thermal convection flow, Phys. Rev. Res. **4**, 033176 (2022).

[38] E. N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. **20**, 130 (1963).

[39] A. Gluhovsky, C. Tong, and E. Agee, Selection of modes in convective low-order models, J. Atmos. Sci. **59**, 1383 (2002).

[40] F. Chillà and J. Schumacher, New perspectives in turbulent Rayleigh-Bénard convection, Eur. Phys. J. E: Soft Matter Biol. Phys. **35**, 58 (2012).

[41] B. Stevens, Atmospheric moist convection, Annu. Rev. Earth Planet. Sci. **33**, 605 (2005).

[42] J. Schumacher and K. R. Sreenivasan, Colloquium: Unusual dynamics of convection in the sun, Rev. Mod. Phys. **92**, 041001 (2020).

[43] L. Sirovich, Turbulence and the dynamics of coherent structures. I - Coherent structures, Q. Appl. Math. **45**, 561 (1987).

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

[45] F. Heyder, J. P. Mellado, and J. Schumacher, Generalizability of reservoir computing for flux-driven two-dimensional convection, Phys. Rev. E **106**, 055303 (2022).

[46] nek5000 version 17.0 (2017), https://nek5000.mcs.anl.gov/.

[47] J. Bailon-Cuba and J. Schumacher, Low-dimensional model of turbulent Rayleigh-Bénard convection in a cartesian cell with square domain, Phys. Fluids **23**, 077101 (2011).

[48] S. Pandey and J. Schumacher, Reservoir computing model of two-dimensional turbulent convection, Phys. Rev. Fluids **5**, 113506 (2020).

[49] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Comput. **9**, 1735 (1997).

[50] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv:1412.3555.

[51] S. Pandey, J. Schumacher, and K. R. Sreenivasan, A perspective on machine learning in turbulent flows, J. Turbul. **21**, 567 (2020).

[52] F. Heyder and J. Schumacher, Echo state network for two-dimensional turbulent moist Rayleigh-Bénard convection, Phys. Rev. E **103**, 053107 (2021).

[53] V. Valori, R. Kräuter, and J. Schumacher, Extreme vorticity events in turbulent Rayleigh-Bénard convection from stereoscopic measurements and reservoir computing, Phys. Rev. Res. **4**, 023180 (2022).

[54] M. Lukoševičius and A. Uselis, Efficient implementations of echo state network cross-validation, Cogn. Comput. **15**, 1470 (2023).

[55] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, Phys. Rev. Lett. **120**, 024102 (2018).

[56] S. Čindrak, B. Donvil, L. Aujogue, and K. Lüdge, Solving the time-complexity problem and tuning the performance of quantum reservoir computing by artificial memory restriction, arXiv:2306.12876.

[57] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, Predictions of turbulent shear flows using deep neural networks, Phys. Rev. Fluids **4**, 054603 (2019).

[58] M. S. Anis, H. Abraham *et al.*, Qiskit: An open-source framework for quantum computing (2021), https://www.ibm.com/quantum/qiskit.