Master's Thesis

Optimization of Quantum Reservoir Computing for Small Quantum Systems

Joel Steinegger

Ludwig-Maximilians-Universität München
Department: Physics
Supervisor: PD Dr. Christoph Räth

Munich, July 23th, 2024

Masterarbeit

Optimierung von Quanten Reservoir Computing für kleine Quantensysteme

Joel Steinegger

Ludwig-Maximilians-Universität München
Fakultät: Physik
Betreuer: PD Dr. Christoph Räth

München, 23. Juli 2024

Contents

Chapter 1

Introduction

The prediction of the evolution of dynamical systems [1] is a fundamental challenge in various disciplines, including science, engineering, medicine, and economics. The ability to predict future trends and behaviours from historical data could lead to significant progress in these fields. The advancements in artificial intelligence (AI) are remarkable, and recent progress in data-driven AI has resulted in substantial improvements in forecasting complex dynamical systems. In this context, reservoir computers (RCs) have emerged as a well-suited approach for predicting the evolution of chaotic dynamical systems [2, 3, 4, 5, 6, 7, 8]. Compared to other recurrent neural networks (RNNs), RCs require only small training datasets, do not suffer from the vanishing gradient problem [9], and have relatively low computational needs. The reservoir computing (RC) approach originated from two independent works: one in the field of machine learning [10] and the other in computational neuroscience [11]. The core of the RC model is a neural network with loops, known as the reservoir, which acts as a memory and yields a reservoir state for a given input. The reservoir is kept constant, and only the weights of a linear output layer are optimized to map the reservoir state to the correct output using linear regression. This practice of linearly mapping the reservoir response results in fast and computationally efficient training. Consequently, this training leads to machine learning models that are orders of magnitude more efficient than other AI approaches.

Over the last few decades, people have enjoyed and benefited from the exponential growth in computational power, a phenomenon known as Moore's Law (first posited in [12]). Despite this extraordinary development, humanity remains unsatisfied and continually seeks more computational power. With predictions about the gradual end of Moore's Law [13], various fields have emerged, aiming to develop novel computational frameworks that surpass conventional computers and go beyond the capabilities of classical von Neumann computing concepts. One such field, which originated in the 1980s [14] but has gained significant momentum in the last decade, is quantum computing. Recent experimental progress in controlling complex quantum systems is remarkable, but a full-fledged universal quantum computer seems out of reach in the near-term future. The quantum hardware currently available and expected in the near future is known as NISQ (Noisy Intermediate-Scale Quantum) [15, 16, 17] hardware. While it has its limitations, there is hope that specific algorithms for this hardware can be developed to outperform classical computing methods. Another innovative computational method is novel neuromorphic computing systems, exemplified by physical reservoir computing. In this approach, the software implementation of the reservoir (static RNN, echo state network) is replaced by a complex physical system, which is used to project low-dimensional input into a high-dimensional dynamical state. If the dynamics of the physical reservoir exhibit memory and non-linearity, only a linear layer needs to be trained. Proposed physical systems include for example biological systems [18], optoelectronic systems [19, 20, 21, 22, 23, 24, 25], and neuromorphic chips [26]. Quantum Reservoir Computing (QRC) [27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39] is located at the intersection of the two computational frameworks discussed above. In QRC, a quantum system is employed as the physical reservoir, leveraging its exponentially large Hilbert space for time series forecasting. Due to the efficient and simple training used in reservoir computing, this framework is a promising candidate for a quantum computing method that can outperform classical computing on NISQ devices. The aim is to create an effective tool for forecasting the dynamics of various systems originating from different disciplines, based on their past evolution.

In this thesis, a modified version of the originally proposed QRC model [27, 28] is introduced. The focus of this modification is to stabilize predictions and optimize the framework for very small quantum systems. The prediction quality of the modified model is evaluated by forecasting various prototypical, synthetic chaotic systems by numerical simulations of the novel framework. The remainder of this thesis is organized as follows: In Chapter 2, the "classical" RC algorithm is outlined first. This is followed by the introduction of the QRC framework based on this foundation and the mathematical description of N-qubit systems. Chapter 3 provides details on the numerical simulations and the physical properties of the quantum systems that are employed as quantum reservoirs in the models analyzed in this thesis. In Chapter 4, the basics of chaotic dynamical systems (testing data) necessary for this thesis are discussed, and the measures used to evaluate prediction accuracy are defined. Chapter 5 presents the results of the numerical experiments. The influences of all introduced changes to the algorithm are investigated, and their necessity for achieving accurate forecasting is highlighted. Furthermore, the strength of the QRC framework in the realm of dynamical system prediction is evaluated by optimizing the hyperparameters of the QRC model for eight different chaotic systems. NISQ-era devices are significantly constrained in terms of their size (qubit count). The predictions presented in Chapter 5 are obtained by numerically simulating quantum systems that are of the smallest theoretically allowed size to showcase the predictive power of these models, even when only severely size-restricted quantum systems are accessible. In the last chapter (Chapter 6), the results of this thesis are summarized, and an outlook on future possible research is presented.

Chapter 2

Quantum Reservoir Computing

The QRC framework originated from the idea of combining quantum computing and physical reservoir computing to utilize quantum dynamics for machine learning tasks [27] and is therefore part of the relatively new class of quantum machine learning models. In physical RC setups the software implementation (called echo state network) of a reservoir state and its dynamics as described in Sec. 2.1 are replaced by a real physical system, and the dynamics of this system are utilized for temporal information processing. In QRC, the physical reservoir is a controllable quantum system. What makes this idea so powerful compared to other quantum machine learning models is that no fine-tuning of the parameters of the quantum system and its dynamics is required. The potential benefits of an exponentially large Hilbert space are, nevertheless, retained. This is due to the fact that the only optimization that is required takes place on a classical computer, leaving the parameters of the quantum system unchanged. This makes the algorithm well-suited for NISQ devices. Which type of quantum system is the most suitable remains an open question. The proposed quantum reservoirs include spin-based [27, 28, 30, 32, 29, 33], continuous variable bosonic [35, 40, 41, 42], and fermionic [43] systems. The subsequent introduction of the QRC model and the numerical investigations presented in the remainder of the thesis are conducted with spin-based systems. The QRC-algorithm presented in this thesis tries to leverage the complex dynamics of a quantum system for real-time machine learning tasks. There are proposals to apply quantum reservoir models to other machine learning problems such as classification and regression, necessitating modifications to the framework. These models are called quantum extreme learning machines (QELMs) [44, 28, 39] and are not investigated in this thesis. The QRC framework utilized in this thesis is outlined in this chapter. The chapter is thematically divided into an introduction to the "classical" RC algorithm (Sec. 2.1), an overview of the mathematical description of the used quantum systems (Sec. 2.2) and a detailed description of the QRC framework (Sec. 2.3).

2.1 Reservoir Computing

In the following, the RC framework is introduced. The goal is not to introduce RC in every last detail but to give a comprehensive overview that is needed to follow the presentation of the QRC framework in Sec. 2.3. For more details about RC, consult the following papers [45, 46, 47].

The RC and QRC framework are designed for modeling the relationship between an input time series $\mathbf{x}(t)$ and an output time series $\mathbf{y}(t)$. More precisely defined, the task is to approximate for a given discretized input time series $\mathbf{x}(t) = \{\mathbf{x}(t_0), \mathbf{x}(t_0 + \Delta t), \ldots\} = \{\mathbf{x}_j\}_{j=1}^{l}$ and target time series $\mathbf{y}(t) = \{\mathbf{y}(t_0), \mathbf{y}(t_0 + \Delta t), \ldots\} = \{\mathbf{y}_j\}_{j=1}^{l}$ a non-linear function f that fulfills

$$\mathbf{y}_k = f(\{\mathbf{x}_j\}_{j=1}^{k}), \tag{2.1}$$

and generalizes for unseen data. A framework capable of handling these tasks must have the ability to process sequential data and memorize past inputs. A class of machine learning methods that allows such data processing is the Recurrent Neural Network (RNN) [48]. Reservoir Computing is based on an artifical and crucially static RNN - the reservoir. Static in this context means that after generating a random network, the network does not change in the training and prediction phases. Only a linear readout layer is optimized in the training phase of the model. This readout layer is kept constant in the prediction phase. This makes RC computationally efficient and prevents the vanishing gradient problem, one of the main drawbacks of standard RNNs [9]. In the following the echo state network and its defining equations are presented. Afterwards, it is detailed how this network is used in RC for temporal tasks.

The echo state network consists of D_r nodes, also called the reservoir state, denoted by a D_r-dimensional vector

$$\mathbf{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_{D_r} \end{pmatrix}. \tag{2.2}$$

The reservoir is adequately chosen by electing the reservoir dimension significantly higher-dimensional than the input time series ($D_r \gg D_x$). The input sequence is recurrently coupled to the high-dimensional reservoir state in a way that the reservoir state becomes a non-linear function F depending on the input up to the current step in the sequence:

$$\mathbf{r}(k) = F(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots). \tag{2.3}$$

An additional feature this function F is required to exhibit is the echo state property [49]. Meaning that the influence of the past input fades away over time. Therefore, the reservoir state encodes information about the recent input history in a high-dimensional space. The conventional method to process the input series in such a way means applying

$$\mathbf{r}(k+1) = \sigma(\mathbf{W}_{\text{in}}\mathbf{x}_{k+1} + \mathbf{A}\mathbf{r}(k)) \tag{2.4}$$

to evolve the reservoir state. The input series is fed into the reservoir state recursively by the input matrix $\mathbf{W}_{\text{in}}$ (input layer) with dimension $D_r \times D_x$. The information from past inputs is retained by the connection between subsequent layers established by the $D_r \times D_r$ adjacency matrix $\mathbf{A}$ (the reservoir network). The non-linearity of the function F originates from the elementwise applied activation function σ (e.g., $\tanh, \text{sigmoid}, \text{etc.}$).

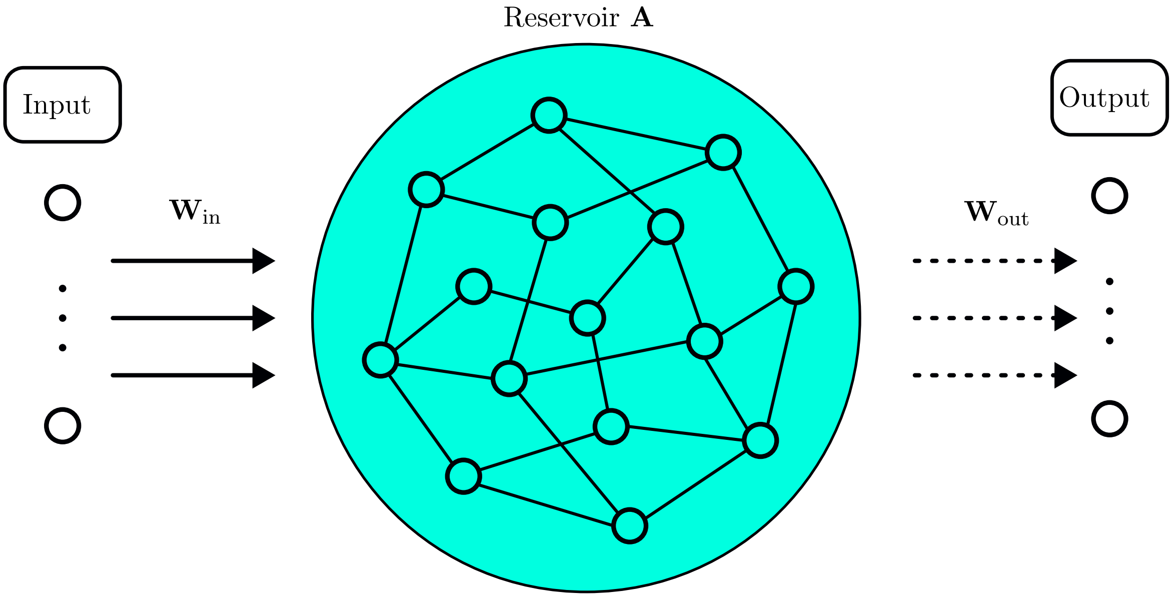


Fig. 1: Schematical illustration of RC (inspired by [27]).

The discrete output time series $\mathbf{o}(t) = \{\mathbf{o}(t_0), \mathbf{o}(t_0 + \Delta t), \ldots\} = \{\mathbf{o}_j\}_{j=1}^{l}$ of the reservoir model is determined by a linear readout, given by the $D_y \times D_r$ readout matrix $\mathbf{W}_{\text{out}}$:

$$\mathbf{o}_k = \mathbf{W}_{\text{out}}\mathbf{r}(k). \tag{2.5}$$

The output matrix is the non-static part after initialization that is optimized in the training process such that the output time series approximates the target time series. The structure of this is schematically represented in Fig. 1. The nodes of the reservoir state are symbolized by the small circles inside the big turquoise circle, the reservoir network $\mathbf{A}$ is illustrated by the connections between the nodes and the input matrix is represented by the solid arrows connecting the input to the reservoir. The solid lines of the input matrix and the reservoir network indicate that these are static, not optimized parts of the RC model. The non-static readout matrix is represented by dotted lines in Fig. 1. The task to find a function f that satisfies Eq. 2.1 is reduced to finding the right coefficients of the readout matrix $\mathbf{W}_{\text{out}}$ such that

$$\mathbf{y}_k \approx \mathbf{o}_k = \mathbf{W}_{\text{out}}\mathbf{r}(k). \tag{2.6}$$

To accomplish this, the RC model is divided into three phases: initialization, training, and prediction.

Initialization: $\mathbf{W}_{\text{in}}$ and $\mathbf{A}$ are chosen in a way which fulfills the echo state property. Here it is important to note again that these matrices are kept unchanged in the training and prediction phase. The final initialization step is to set each reservoir node to an arbitrary value. More details about reservoir initialization can be found in [50, 51, 52, 53].

Training: For a given training data set $\{\{\mathbf{x}_j\}_{j=1}^{l}, \{\mathbf{y}_j\}_{j=1}^{l}\}$. The first step of training is the washout phase (synchronization phase). The first N_{sync} inputs are used to synchronize the reservoir state $\mathbf{r}(k)$ with the dynamics of the input data by evolving the reservoir state N_{sync} times (applying Eq. 2.4). Training the reservoir is achieved by using the remaining N_{train} steps of the training data set to evolve and collect the reservoir states after each application of Eq. 2.4 in a $D_r \times N_{\text{train}}$ matrix $\mathbf{R} = [\mathbf{r}(N_{\text{sync}} + 1), \ldots, \mathbf{r}(N_{\text{sync}} + N_{\text{train}})]$. The desired outputs are collected in a matrix $\mathbf{Y} = [\mathbf{y}_{N_{\text{sync}}+1}, \ldots, \mathbf{y}_{N_{\text{sync}}+N_{\text{train}}}]$. To solve the task defined in Eq. 2.1 the objective is to find a readout matrix $\mathbf{W}_{\text{out}}$ such that the reservoir output time series approximates the target time series as well as possible by solving the equation

$$\mathbf{Y} = \mathbf{W}_{\text{out}}\mathbf{R}. \tag{2.7}$$

There are different types of regression that can be employed, but the most common one is ridge regression [54]. $\mathbf{W}_{\text{out}}$ is obtained by calculating:

$$\mathbf{W}_{\text{out}} = \mathbf{Y}\mathbf{R}^{\text{T}}(\mathbf{R}\mathbf{R}^{\text{T}} - \beta\mathbb{1})^{-1}. \tag{2.8}$$

β is called the regression parameter and is one of the essential hyperparameters of RC. The benefit of ridge regression is that large coefficients of the readout matrix are penalized. This adds regularization, prevents overfitting, and therefore makes predictions more stable [47]. This is achieved by minimizing the error function

$$\sum_{N_{\text{sync}}<k<N_{\text{train}}} \|\mathbf{W}_{\text{out}}\mathbf{q}(k) - \mathbf{u}_k\|^2 + \beta \cdot \text{Tr}(\mathbf{W}_{\text{out}}^{\text{T}}\mathbf{W}_{\text{out}}). \tag{2.9}$$

Prediction: The prediction phase strongly depends on the specific details of the temporal machine learning task. In this thesis the tasks are time series prediction tasks where the input and the output time series are given by the true discrete time (training) series $\mathbf{u}(t)$

$= \{\mathbf{u}(t_0), \mathbf{u}(t_0 + \Delta t), \dots\} = \{\mathbf{u}_j\}_{j=1}^{L}$. Meaning that the model is designed to forecast (continue) a d-dimensional time series from past data. This is achieved by training the RC network to predict the next step in the time series from the current reservoir state holding information about the present and recent past of the time series. The function f, meant to generalize for unseen data, which the model aims to approximate, therefore is

$$\mathbf{u}_{k+1} = f(\{\mathbf{u}_j\}_{j=1}^{k}). \tag{2.10}$$

There are L time steps of the time series $\mathbf{u}(t)$ that are used to train the model. This training entails synchronizing the reservoir states and training the readout weights with the following data split:

$$\begin{aligned} &\text{Synchronization input} : \mathbf{x}_i = \mathbf{u}_i \\ &\text{Training input} : \mathbf{x}_j = \mathbf{u}_j \\ &\text{Training output} : \mathbf{y}_j = \mathbf{u}_{j+1} \end{aligned} \tag{2.11}$$

with $i \in \{1, 2, \dots, N_{\text{sync}}\}$, $j \in \{N_{\text{sync}} + 1, N_{\text{sync}} + 2, \dots, N_{\text{train}}\}$ and $N_{\text{sync}} + N_{\text{train}} + 1 = L$. The training is executed as described in the last paragraph and consequently, $\mathbf{W}_{\text{out}}$ is determined such that the matrix maps $\mathbf{r}(k)$ to $\mathbf{u}_{k+1}$ and is fixed after the training phase. In the prediction phase the external input is replaced by the prediction of the RC model, generating an autonomously evolving closed loop (see Fig. 2) continuing the time series. The equations defining the prediction phase are:

$$\mathbf{o}_k = \mathbf{W}_{\text{out}} \mathbf{r}(k) \tag{2.12}$$

and

$$\mathbf{r}(k+1) = \sigma(\mathbf{W}_{\text{in}} \mathbf{o}_k + \mathbf{A}\mathbf{r}(k)). \tag{2.13}$$

By applying this closed loop, the discrete time series can be continued for an arbitrary amount of steps.

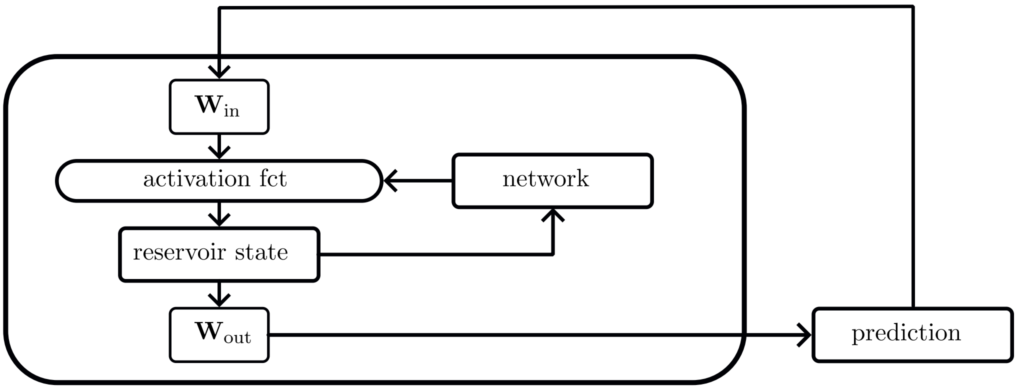


Fig. 2: Schematical illustration of prediction phase of the RC model.

2.2 N-qubit system and unitary evolution

This section provides a brief overview of the mathematical representation of a system of N qubits and their dynamics. The main goal is to motivate the choice of a quantum reservoir by showing the resemblance of an echo state network (see Sec. 2.1) and the mathematical description of a system of qubits evolving under a unitary evolution.

A qubit is mathematically described as

$$|\Phi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \text{ with } |\alpha|^2 + |\beta|^2 = 1. \tag{2.14}$$

It is the smallest, two-dimensional ($\mathbb{C}^2$) quantum object. A system of N qubits is defined by a tensor product space $(\mathbb{C}^2)^{\otimes N}$ and a state of this system is described as

$$|\Psi\rangle = \sum_{x_1,\ldots,x_N \in \{0,1\}} \alpha_{x_1,\ldots,x_N} |x_1,\ldots,x_N\rangle . \tag{2.15}$$

$|x_1,\ldots,x_N\rangle$ are called the basis states and the tensor product space is described by a 2^N vector space $\mathbb{C}^{2^N}$. Which means that the dimension of the Hilbertspace increases exponentially with the number of qubits. A more general form, also being able to describe a classical ensemble of quantum states, is to describe the state of the quantum system of N qubits in operator form by a $2^N \times 2^N$ density matrix

$$\rho = \sum_k p_k |\Psi_k\rangle \langle\Psi_k| , \tag{2.16}$$

where $|\Psi_k\rangle$ are the quantum states comprising the ensemble, and p_k are their respective probabilities. The density matrix is hermitean, positive semidefinite and the trace (Tr(.)) of the density matrix is equal to one. Under unitary evolution (e.g., time evolution) does an N-qubit system described by a density matrix ρ evolve as

$$\rho' = \mathbf{U}\rho\mathbf{U}^\dagger . \tag{2.17}$$

A unitary evolution acting on an N-qubit system is described by a $2^N \times 2^N$ unitary matrix $\mathbf{U}$. The density matrices fulfill the axioms of a linear space. A consequence of this is that the operator formalism can be rewritten into a vector representation by defining an inner product for two operators A and B. The Hilbert-Schmidt inner product is defined as

$$\mathrm{Tr}(\mathrm{A}^\dagger \mathrm{B}). \tag{2.18}$$

The density matrix of an N-qubit system can be spanned by tensor products of the Pauli operators $\{\mathbb{1}, \sigma_\mathrm{x}, \sigma_\mathrm{y}, \sigma_\mathrm{z}\}^{\otimes N}$ with

$$\mathbb{1} = \sigma_{0,0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \sigma_\mathrm{x} = \sigma_{1,0} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_\mathrm{y} = \sigma_{1,1} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \text{ and } \sigma_\mathrm{z} = \sigma_{0,1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.19}$$

The tensor products

$$\mathrm{P}(\mathbf{i}) = \bigotimes_{k=1}^{N} \sigma_{i_{2k-1}, i_{2k}}, \tag{2.20}$$

so called N-qubit Pauli operators, form an orthogonal basis $\{\mathrm{P}(\mathbf{i})\}$ of the operator space with 4^N elements. $\mathbf{i}$ is a $2N$-dimensional vector with binary entries and i_k is the k-th coefficent of this vector. Meaning that any density matrix ρ can be decomposed as a linear combination of the N-qubit Pauli operators:

$$\rho = \sum_{\mathbf{i}} r_\mathbf{i} \mathrm{P}(\mathbf{i}) \qquad \text{with } r_\mathbf{i} = \mathrm{Tr}(\mathrm{P}(\mathbf{i})\rho)/2^N. \tag{2.21}$$

Here, $\sum_\mathbf{i}$ denotes the sum over all possible binary vectors of size $2N$. Therefore, any state of an N-qubit system can be represented by a 4^N-dimensional real vector

$$\mathbf{r} = \begin{pmatrix} r_{0,0,\ldots,0} \\ \vdots \\ r_{1,1,\ldots,1} \end{pmatrix} . \tag{2.22}$$

In this representation the unitary evolution $\rho' = \kappa(\rho) = \mathbf{U}\rho\mathbf{U}^\dagger$ is rewritten as a matrix $\mathbf{K}$ acting on the vector $\mathbf{r}$ (describing the quantum state):

$$\rho' = \kappa(\rho) \Leftrightarrow \mathbf{r}' = \mathbf{K}\mathbf{r}. \tag{2.23}$$

RC	N-qubit system
reservoir state: $\mathbf{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_{D_r} \end{pmatrix}$	quantum state: $\mathbf{r} = \begin{pmatrix} r_{0,0,\dots,0} \\ \vdots \\ r_{1,1,\dots,1} \end{pmatrix}$
layer connection: $\mathbf{r}' = \mathbf{A}\mathbf{r}$	unitary evolution: $\mathbf{r}' = \mathbf{K}\mathbf{r}$
input matrix: $\mathbf{W}_{\text{in}}$	?
activation function: σ	?

Table 1: Comparison of the RC implementation and an N-qubit quantum system evolving under unitary evolution.

The elements of matrix $\mathbf{K}$ are given by

$$K_{ij} = \mathrm{Tr}(\mathrm{P}(\mathbf{i})\kappa(\mathrm{P}(\mathbf{j})))/2^N. \tag{2.24}$$

This vector representation shows the similarity between the reservoir state of the RC model (Eq. 2.2) and a quantum state of an N-qubit system. Furthermore there is a similarity between the part of Eq. 2.4 connecting two consecutive reservoir states described by the adjacency matrix $\mathbf{A}$ and the evolution $\mathbf{K}$ of the quantum state. Table 1 displays the resemblance between the state and dynamics of RC and the state and dynamics of a quantum system in the right formalism. This motivates the replacement of the reservoir state with an N-qubit system. There are multiple potential advantages to this replacement. One of the main advantages is due to the exponentially growing and therefore high-dimensionality of the quantum state (dim($\mathbf{r}$)= 4^N) which is already for 6 qubits significantly higher than the typical "classical" reservoir state size of a few hundred nodes. This means that the encoding of the input is also significantly higher-dimensional. The requirement of relatively few qubits is another advantage of the QRC model compared to other quantum computing algorithms. This makes an application on near-term available quantum hardware more likely. Table 1 also shows that there are open questions about the replacement. Especially how the time series is encoded (input matrix) and how the essential non-linearity (activation function in RC) enters the model. The next section (Sec. 2.3) explains the full QRC framework in all details. Sec. 3.5 highlights how the required non-linearity emerges.

2.3 Quantum reservoir computing framework

The similarity between the software implementation of the RC algorithm and the state of an N-qubit system and its dynamics under a unitary evolution is striking. But going from the RC to a QRC algorithm is not completely straight forward (as previously mentioned) and requires some adjustments to the algorithm. The QRC algorithm investigated in this thesis is a modified version of the original QRC algorithm described in [27, 28, 30]. The type of tasks that the QRC model is designed to solve is the same as defined for the RC model (described in Sec. 2.1 and defined in Eq. 2.1). The objective is to replace the software implementation of a reservoir with a quantum system of N interacting qubits. The interactions with the classical reservoir are threefold: input injection, dynamics, and output extraction. The same three interactions have to be defined for a quantum reservoir. At first these important components of the algorithm are described. Afterwards the entire algorithm is explained in detail. The reservoir in QRC is an N-qubit system described by a density matrix ρ. The quantum state is supposed to encode sequentially the information of the input sequence $\mathbf{x}(t) = \{\mathbf{x}(t_0), \mathbf{x}(t_0+\Delta t), \ldots\} = \{\mathbf{x}_j\}_{j=1}^{l}$. For this, a recursive operation must be defined that encodes the current input into the quantum state, preserves the information of previous inputs, and spreads the information throughout the state. This operation defines how to get from the quantum state $\rho(k)$ of step k to the quantum state $\rho(k+1)$ of step k+1. This operation is defined by

$$\rho(k+1) = \mathbf{U}(\rho_{x_k^1} \otimes .... \otimes \rho_{x_k^{D_x}} \otimes \mathrm{Tr}_{1,\ldots,D_x}(\rho(k)))\mathbf{U}^\dagger \tag{2.25}$$

with

$$\rho_{x_k^i} = \left|x_k^i\right\rangle\!\left\langle x_k^i\right| \text{ and } \left|x_k^i\right\rangle = \sqrt{1-x_k^i}\,|0\rangle + \sqrt{x_k^i}\,|1\rangle\,. \tag{2.26}$$

Here and in the following, the i-th coefficient of the k-th step of the discrete time series $\mathbf{x}(t)$ is denoted as x_k^i and $\mathrm{Tr}_{1,\ldots,D_x}(.)$ is the partial trace over the first D_x qubits. The recurrent map can be split into two parts. The first part is the encoding of the current step of the time series. For each dimension of the input time series, the current state of one qubit is replaced by a state that encodes the information in its amplitude. This has two immediate consequences. The first consequence is that to retain information about past inputs the number of qubits has to be greater than the dimension of the input data, i.e., $N > D_x$. The second consequence is that the input data has to lie in the interval [0,1], which means that the input time series has to be scaled into that interval. In other words, the time series has to be normalized. This adds two hyperparameters to the model. The time series is scaled into the interval $[a,b]$ with the hyperparameters $0 \leq a < b \leq 1$. More comments about this can be found in Sec. 3.3. After the injection of the input, the system evolves under the unitary evolution $\mathbf{U}$. Details about the numerical implementation can be found in Chapter 3. This recurrent map can be rewritten in the vector representation of the density matrices:

$$\mathbf{r}(k+1) = \mathbf{K}\mathbf{X}(k)\mathbf{r}(k). \tag{2.27}$$

Here, $\mathbf{X}(k)$ is the matrix representation of the completely positive trace-preserving (CPTP) map injecting the quantum reservoir with the input time series. The matrix elements of $\mathbf{X}(k)$ are defined by

$$X(k)_{ij} = \mathrm{Tr}(\mathrm{P}(\mathbf{i})\ \rho_{x_k^1} \otimes .... \otimes \rho_{x_k^{D_x}} \otimes \mathrm{Tr}_{1,\ldots,D_x}(\mathrm{P}(\mathbf{j})))/2^N. \tag{2.28}$$

$\mathbf{K}$ (defined in Eq. 2.24) is the matrix representation of the unitary evolution acting on the vector representation of the density matrices.

The missing part of the information processing with a quantum system is how to extract information out of the quantum system and use the high-dimensional encoding of the time series for temporal machine learning tasks. In the RC algorithm the information extraction is achieved by Eq. 2.5 which means all the nodes are used when training the model. This is not possible in the QRC framework because obtaining all the entries of the quantum state vector representation of the density matrix through measurements is not a realistic assumption. What is experimentally feasible, is to measure observables such as spin projections and spin correlations. The output layer of the quantum reservoir (reservoir response) is constructed out of a number of expectation values of these observables. The coefficients of the state vector that are used for training are collected in a vector $\mathbf{n}(k)$ and are called the true nodes. This means that only parts of the vector $\mathbf{r}$(k), which describes the quantum system at step k, are used directly in the presented framework. The rest of the coefficients are called the hidden nodes. They contribute to the dynamics, the memory, and the encoding, but not directly to the training. This is illustrated by Fig. 3 that schematically represents the QRC model. The nodes in the turquoise circle embody the true nodes and the ones in the red circle the hidden nodes. Comparing this graphic with the graphic of the RC model (Fig. 1) illustrates one of the key differences between the two models. The RC model has fewer nodes but all of them are used

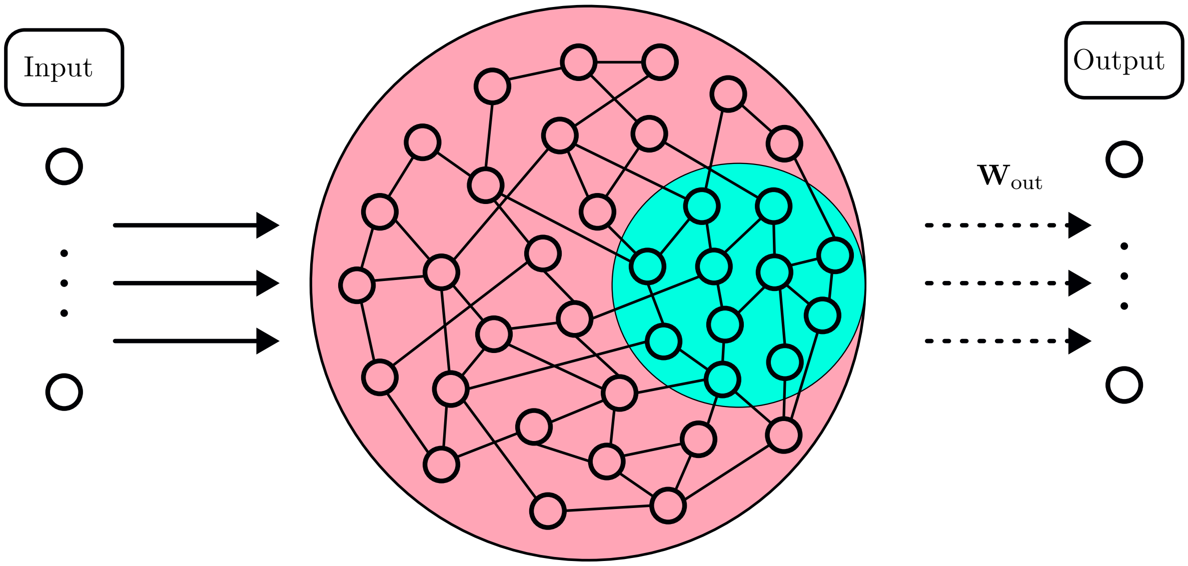


Fig. 3: QRC schematical illustration (inspired by [27]).

for training. The QRC model has more nodes but only a fraction can be directly used as output.

The small number of true nodes limits the performance of QRC, especially when considering the relatively small number of qubits (small quantum systems) due to simulation restrictions, while also keeping in mind the near-term experimental availability. To improve performance by increasing the number of nodes used for training, without increasing the size of the quantum system, several methods are employed.

Rather than employing the unitary evolution and the measurement one time before encoding the next step of the time series, the evolution and measurement process is carried out V times. The output vectors for each of these single measurement phases are merged together into one output vector

$$\mathbf{v}(k) = \begin{pmatrix} \mathbf{n}_1(k) \\ \vdots \\ \mathbf{n}_V(k) \end{pmatrix} \tag{2.29}$$

of step k. For n true nodes the number of nodes used for training is expanded to $n \times V$ nodes. This method is called temporal multiplexing. This procedure is sketched in Fig. 4.

Another method is spatial multiplexing (first introduced by [29]). The concept involves increasing the number of employed quantum systems rather than the size of a single quantum system. Instead of using one quantum system, r quantum systems are used. At each step k the output of one reservoir is the $n \times V$-dimensional vector $\mathbf{v}(k)$. These are collected from all of the r reservoirs and combined to one output vector

$$\mathbf{p}(k) = \begin{pmatrix} \mathbf{v}_1(k) \\ \mathbf{v}_2(k) \\ \vdots \\ \mathbf{v}_r(k) \end{pmatrix}. \tag{2.30}$$

This process produces an $r \times n \times V$-dimensional output vector. For this process to be useful, the unitary evolution of the systems has to be different for each system. This technique is depicted for V=1 (for clearity) in Fig. 5. But as described above spatial muliplexing can be leveraged

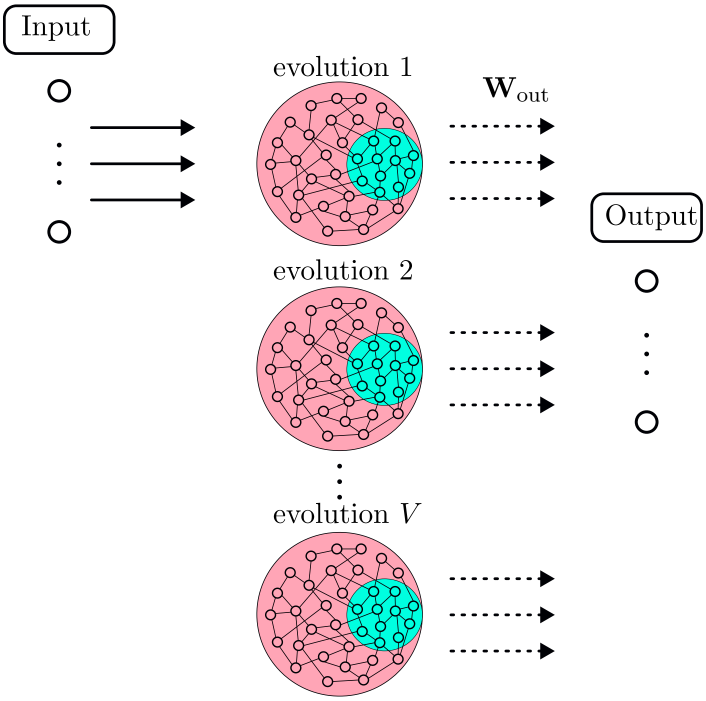


Fig. 4: Schematical illustration of temporal multiplexing.

together with temporal multiplexing. In the figure, the same position of the nodes, but changed connections between the nodes, symbolize the different unitary evolutions of the reservoirs.

The final technique to increase the dimension of the output vector is to apply a function F_{res} to the output. The output vector $\mathbf{q}(k)$ at step k is defined as:

$$\mathbf{q}(k) = \mathrm{F}_{\text{res}}(\mathbf{p}(\mathrm{k})). \tag{2.31}$$

In this work powers of the reservoir readout up to the fourth exponent are considered as readout function. The studied choices are thus only using the readout vector $\mathbf{p}(k)$ and a bias term

$$\mathbf{q}(k) = \mathrm{F}_{\text{res}}(\mathbf{p}(k)) = \begin{pmatrix} 1 \\ \mathbf{p}(k) \end{pmatrix}, \tag{2.32}$$

the readout of the r reservoirs, its square and a bias term [55]

$$\mathbf{q}(k) = \mathrm{F}_{\text{res}}(\mathbf{p}(k)) = \begin{pmatrix} 1 \\ \mathbf{p}(k) \\ \mathbf{p}^2(k) \end{pmatrix}, \tag{2.33}$$

the readout vector $\mathbf{p}(k)$, its second and third power and a bias term

$$\mathbf{q}(k) = \mathrm{F}_{\text{res}}(\mathbf{p}(k)) = \begin{pmatrix} 1 \\ \mathbf{p}(k) \\ \mathbf{p}^2(k) \\ \mathbf{p}^3(k) \end{pmatrix}, \tag{2.34}$$

and the readout vector $\mathbf{p}(k)$, its powers up to four and a bias term

$$\mathbf{q}(k) = \mathrm{F}_{\text{res}}(\mathbf{p}(k)) = \begin{pmatrix} 1 \\ \mathbf{p}(k) \\ \mathbf{p}^2(k) \\ \mathbf{p}^3(k) \\ \mathbf{p}^4(k) \end{pmatrix}. \tag{2.35}$$

This selection of readout functions adds the hyperparameter G indicating the powers (for each coefficient) of the readout being included to the model. This method is inspired by recently

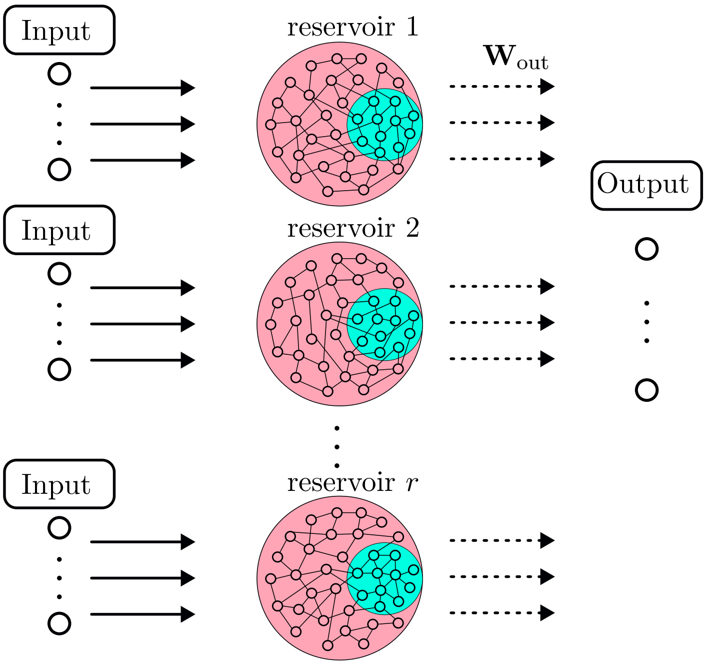


Fig. 5: Schematical illustration of spatial multiplexing.

developed methods in "classical" RC [56]. The final output vector $\mathbf{q}(k)$ is an $G \times r \times n \times V + 1$-dimensional vector.

This final output vector $\mathbf{q}(k)$ (equivalent to the reservoir state in RC) is mapped by the readout matrix $\mathbf{W}_{\text{out}}$ to the reservoir output

$$\mathbf{o}_k = \mathbf{W}_{\text{out}}\mathbf{q}(k). \tag{2.36}$$

To train the model, the same training as in the RC framework is implemented. The output matrix $\mathbf{W}_{\text{out}}$ is optimized in such a way that the discrete output time series $\mathbf{o}(t) = \{\mathbf{o}(t_0), \mathbf{o}(t_0 + \Delta t), \dots\} = \{\mathbf{o}_j\}_{j=1}^{l}$ approximates the target time series $\mathbf{y}(t) = \{\mathbf{y}(t_0), \mathbf{y}(t_0 + \Delta t), \dots\}$=$\{\mathbf{y}_j\}_{j=1}^{l}$

$$\mathbf{y}_k \approx \mathbf{o}_k = \mathbf{W}_{\text{out}}\mathbf{q}(k). \tag{2.37}$$

The one classical part of the algorithm is optimized whereas the parameters of the quantum system stay unchanged. Because the QRC model follows the same structure as the RC model, it is split into the same 3 phases: initilization, training and prediction phase.

Initilization: The hyperparameters have to be chosen to initialize the model. The unitary operator has to be defined (for each reservoir). More details about the choice of the unitary operator for this thesis are presented in Sec. 3.1. Furthermore it is required to select the number of employed quantum reservoirs r, the number of evolution and measurement processes V, the normalization interval $[a, b]$, the type of readout function (in this thesis controlled by G), the regression parameter β, and the observables used as true nodes. The final step is the initilization of the quantum states. The starting state is given by $\rho(0) = (|0\rangle \langle 0|)^{\otimes N}$, unless otherwise stated.

Training: The training follows the same structure as the training of the RC model. For a given training data set $\{\{\mathbf{x}_j\}_{j=1}^{l}, \{\mathbf{y}_j\}_{j=1}^{l}\}$, the first N_{sync} inputs are used to synchronize the quantum states with the dynamics of the input data, and thereby eliminating any transient dynamics that are a product of the initialization of the quantum states. The synchronization is accomplished by injecting the input into the quantum reservoirs and evolving the quantum systems as defined in Eq. 2.25. The leftover N_{train} steps of the training data set are used to obtain a readout matrix $\mathbf{W}_{\text{out}}$ such that the output time series of the reservoirs approximates the target time series. The systems are sequentially injected with the input time series and evolved N_{train} times. The full output $\mathbf{q}(k)$ is measured and collected in a $\dim(\mathbf{q}(k)) \times N_{\text{train}}$ matrix $\mathbf{Q} = [\mathbf{q}(N_{\text{sync}}+1), \ldots, \mathbf{q}(N_{\text{sync}}+N_{\text{train}})]$ and the desired outputs are collected in a matrix $\mathbf{Y} = [\mathbf{y}_{N_{\text{sync}}+1}, \ldots, \mathbf{y}_{N_{\text{sync}}+N_{\text{train}}]}$. The training is achieved by solving the equation

$$\mathbf{Y} = \mathbf{W}_{\text{out}}\mathbf{Q}. \tag{2.38}$$

As in ”classical” RC (Sec. 2.1) ridge regression is employed:

$$\mathbf{W}_{\text{out}} = \mathbf{Y}\mathbf{Q}^{\text{T}}(\mathbf{Q}\mathbf{Q}^{\text{T}} - \beta\mathbb{1})^{-1}. \tag{2.39}$$

In the original papers on QRC [27, 28] the equation is solved by the Moore-Penrose pseudo-inverse which is equivalent to ridge regression with $\beta = 0$.

Prediction: The temporal machine learning tasks that QRC models are employed to solve in this thesis are the same time series prediction tasks defined for the RC framework. The input and the output time series are given by the d-dimensional discret (training) time series $\mathbf{u}(t)$ $= \{\mathbf{u}(t_0), \mathbf{u}(t_0 + \Delta t), \ldots\} = \{\mathbf{u}_j\}_{j=1}^{L}$. The QRC model is trained to predict the next step in the time series based on the current (quantum) reservoir state, which contains information about the present and recent past of the true time series.

There are L time steps of the time series $\mathbf{u}(t)$ that are used to train the reservoir. The same data split as for the RC model is applied:

$$\begin{aligned} &\text{Syncronization input} : \mathbf{x}_i = \mathbf{u}_i \\ &\text{Training input} : \mathbf{x}_j = \mathbf{u}_j \\ &\text{Training output} : \mathbf{y}_j = \mathbf{u}_{j+1} \end{aligned} \tag{2.40}$$

with $i \in \{1, 2, \ldots, N_{\text{sync}}\},\ j \in \{N_{\text{sync}}+1, N_{\text{sync}}+2, \ldots, N_{\text{train}}\}$ and $N_{\text{sync}} + N_{\text{train}} + 1 = L$.

The output matrix $\mathbf{W}_{\text{out}}$ is trained to map from the output vector $\mathbf{q}(k)$ to $\mathbf{u}_{k+1}$. This means that the prediction of the model $\mathbf{o}_k$ approximates the next step of the time series $\mathbf{u}_{k+1}$ ($\mathbf{u}_{k+1} \approx \mathbf{o}_k$). Once the matrix $\mathbf{W}_{\text{out}}$ is obtained, it remains unchanged and the time series can be continued for an arbitrary amount of steps. To continue the time series an autonomously evolving closed-loop is created (as in RC). In the prediction phase, the external input is replaced by the last prediction of the QRC model (illustrated by Fig. 6):

$$\mathbf{x}_{k+1} = \mathbf{o}_k \text{ for } k > L. \tag{2.41}$$

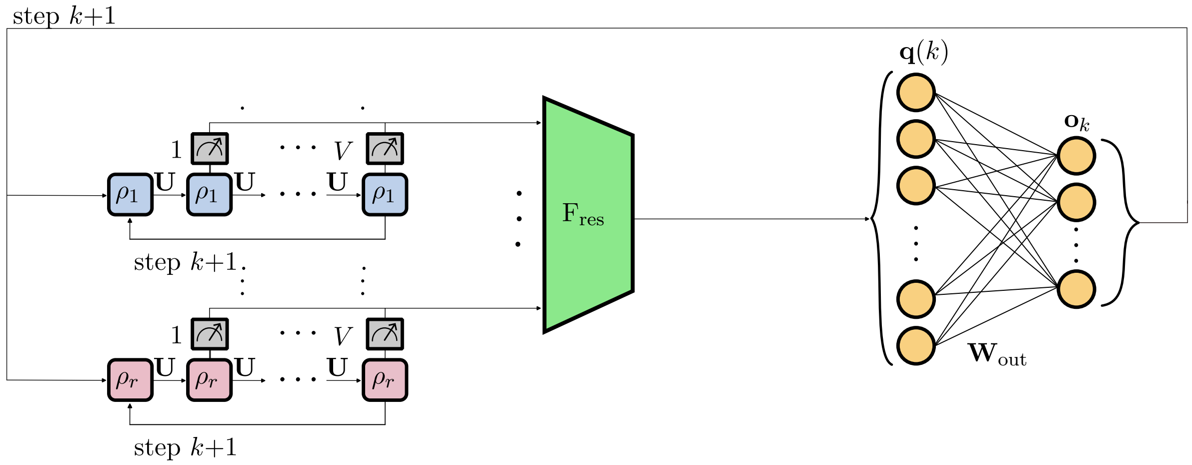


Fig. 6: Schematical illustration of prediction phase of the QRC model.

Chapter 3

Quantum Reservoirs: Unitary Evolution and Model Properties

In Chapter 2, it is described how quantum systems can be exploited for time series forecasting by transitioning from an echo state network to a quantum reservoir. The QRC algorithm is a hybrid quantum algorithm. Some parts of the algorithm utilize quantum hardware, while others are executed on a classical computer. For this thesis all results are obtained by numerical simulation of the quantum systems. In this chapter the details of the simulated quantum systems are defined (Sec. 3.1). Furthermore properties of the quantum system and features of the resulting QRC model are discussed in Secs. 3.2 - 3.5.

3.1 Quantum reservoir simulation

The choice of unitary operator defines the quantum reservoir in the presented model, determining the underlying "network connections". In future real-world applications, the optimal unitary operators will be heavily dependent on the limitations of the available quantum devices. For this work, the unitary time evolution of a quantum system is employed as the unitary evolution of the quantum reservoir.

The simulated models try to capitalize on the real-time dynamics of quantum systems for time series processing. To achieve this, a specific Hamiltonian must be elected. For this thesis, the chosen Hamiltonian is the transverse field Ising model plus onsite disorder, consistent with previous work in the field of QRC [30]. The Hamiltonian of the transverse field Ising model plus onsite disorder is defined as:

$$\mathbf{H} = \sum_{i>j=1}^{N} J_{ij}\sigma_{\mathrm{x}}^{i}\sigma_{\mathrm{x}}^{j} + \frac{1}{2}\sum_{i=1}^{N}(h + D_i)\sigma_{\mathrm{z}}^{i}. \tag{3.1}$$

Where h is the magnetic field , D_i is the onsite disorder uniformly drawn from an interval $[-W, W]$, and J_{ij} is the coupling strength drawn randomly from the interval $[-\frac{J}{2}, \frac{J}{2}]$. All these parameters defining the Hamiltonians are expressed in units of J. To utilize the time evolution of the quantum system, a unit time step size τ (in units $\frac{1}{J}$) is chosen as time between two consecutive inputs. In the following, $J = 1$ is elected for convenience. The observable measurements are carried out V times after letting the reservoirs evolve each time for a time τ/V. The unitary operator ($\hbar$=1) that maps between states is

$$\mathbf{U} = e^{\frac{-i\mathbf{H}\tau}{V}}. \tag{3.2}$$

Fig. 7 sketches the QRC model prediction phase resulting from this choice. This selection of unitary operators may not be the most suitable option for near-term real-world applications. The goal of this thesis is to introduce a framework that provides a good starting point for further research into the use of small quantum systems for time series forecasting with NISQ devices. In the outlook (Chapter 6), the next steps towards real-world applications are discussed. This discussion includes future research on appropriate unitary operators.

3.1.1 Phase diagram

This Hamiltonian was chosen, in this work and in [30], because it exhibits ergodic and localized phases for certain parameter choices. In the mentioned study, the influence of the phases and

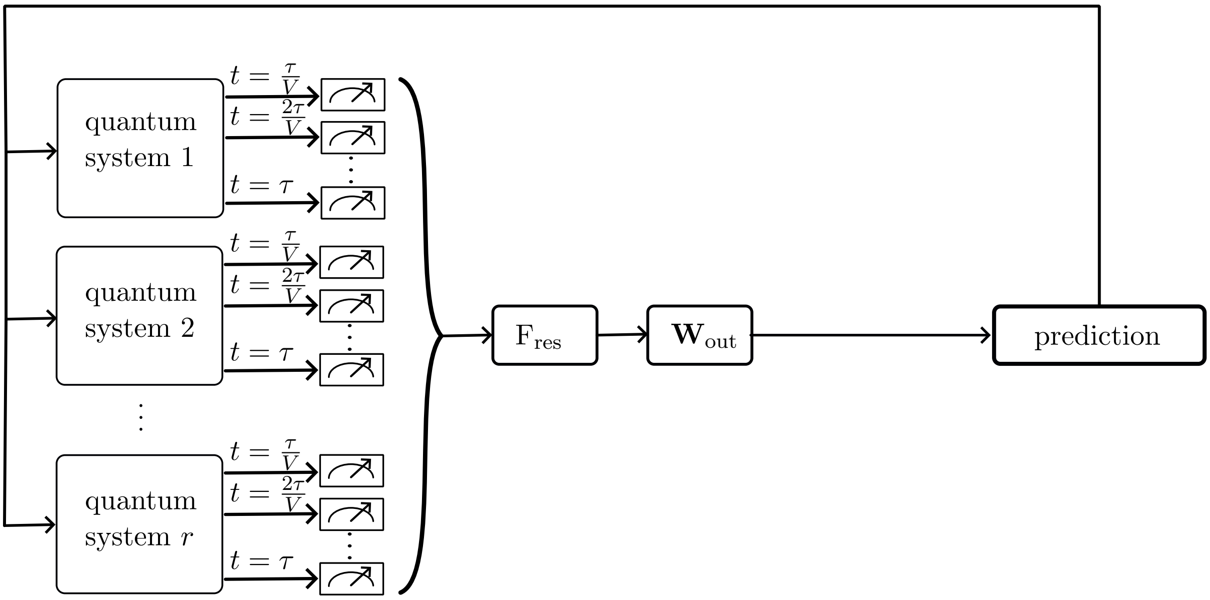


Fig. 7: Schematical illustration of the QRC prediction phase with unitary operator defined by a Hamiltonian and a unit time step between inputs.

the resulting properties of the physical systems on prediction quality is analyzed. It follows an investigation of the phase diagram, which closely aligns with the approach described in [30]. Different phases can be discerned through the analysis of the eigenspectrum using the ratio of adjacent gaps [57]

$$g_n = \frac{\min[\delta_{n+1}, \delta_n]}{\max[\delta_{n+1}, \delta_n]}. \tag{3.3}$$

Here, the gaps δ_n are defined by $\delta_n = E_n - E_{n-1}$ and $\{E_n\}$ is the sorted (in ascending order) list of the eigenvalues of the Hamiltonian (one parameter draw). The level spacing distribution is expected to be different in the ergodic and localized phases according to random matrix theory. The Wigner-Dyson distribution, indicative of level repulsion, characterizes the thermalizing (ergodic) phase. This phase exhibits an average gap ratio ($\langle g \rangle$) of approximately 0.53 [58]. In the many-body localized (non-ergodic) phase, the system has an extensive number of integrals of motion. This allows the eigenvalues from different sectors to be regarded as independent variables, resulting in a Poisson distribution. As a consequence, the average gap ratio in the localized phase is approximately $\langle g \rangle \approx$0.386 [58]. The system (Eq. 3.1) has two parameters (the magnetic field h and the disorder range W) that determine the phase of the system. To study the phase diagram of the quantum system, the average ratio of adjacent gaps is determined and subsequently averaged over numerous Hamiltonian realizations for each parameter combination. The quantum system specified in Eq. 3.1 has a $\mathbb{Z}_2$ symmetry, implying that diagonalization is required only within a single symmetry sector. In Fig. 8 the phase diagram is shown in dependence of the magnetic field and the disorder range. This phase diagram (gridsize=100×100) is obtained with N=10 (10 qubits) and by averaging over 500 realizations for each parameter combination. It shows 2 ergodic (yellow) regions and 2 localized (black) regions. The transitions between the phases strictly occur only in the thermodynamic limit. Nevertheless, signatures of them can be found in finite-size systems. For an in-depth analysis of role of ergodicity within the QRC model and the network's response across various phases, readers are referred to [30]. The subsequent paragraph presents a brief summary of their arguments and conclusions, supplemented with plots specifically generated for this thesis.

During each input step, the qubit corresponding to each dimension of the time series is modified to encode the current state of the time series within the state of that qubit. This encoding

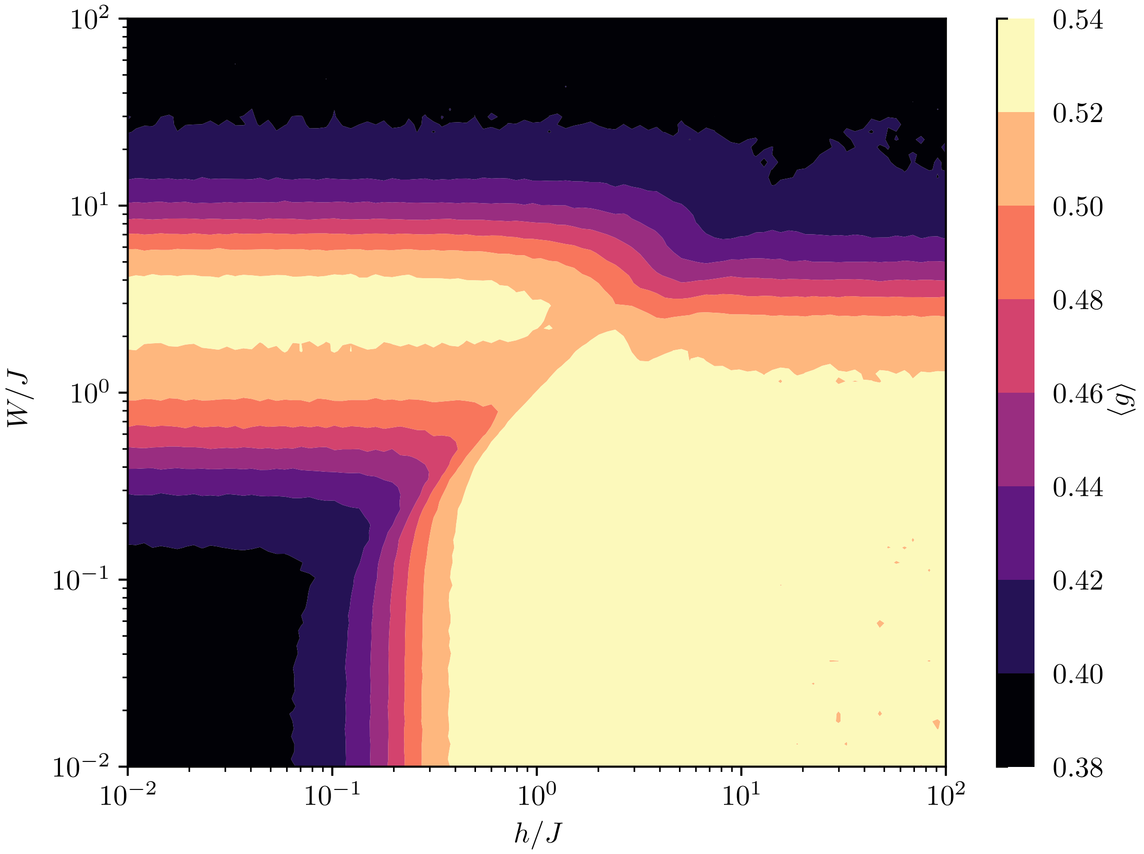


Fig. 8: Heatmap of $\langle g \rangle$ depending on the magnetic field h and the disorder strength W obtained by averaging over 500 realizations of the transverse-field Ising Hamiltonian plus onsite disorder.

process immediately alters the expectation values depending on the modified qubits (e.g., $\langle \mathbf{H} \rangle$). Since the unitary dynamics do not change the conserved quantities of the system (by definition), do these quantities, which are dependent on the modified qubits, become functions of the initial state of the system and the encoded sequence. The total energy $\langle \mathbf{H} \rangle$ and the parity $\langle \mathbf{P} \rangle = \langle \prod_{i=1}^{N} \sigma_z^i \rangle$ are the only conserved quantities in the ergodic regions, each involving all qubits globally. The Eigenstate Thermalization Hypothesis states that local observables primarily depend on these conserved quantities, with deviations constrained to finite-scale fluctuations [59]. This means that local observables (including the ones that are used for training) become functions of the input history. Each input step involves a partial trace and consequently information erasure due to the large amount of entanglement in the system in the ergodic phase. As a result, repetitive inputs induce uniform dynamics of $\langle \mathbf{H} \rangle$ and $\langle \mathbf{P} \rangle$ for different initial states (example shown in Fig. 9). This property is directly related to the emerging echo state property in the ergodic phase presented in Sec. 3.2. The information erasure garanties that the expectation values $\langle \mathbf{H} \rangle$ and $\langle \mathbf{P} \rangle$, along with all local observables, are determined by the input history up to a specific earlier time step, indicating a fading memory effect. This fading memory property is a requirement for a suitable reservoir. In the ergodic phase, the behaviour of the local expectation value depends on the input sequence, as illustrated in Fig. 10. This behaviour is obviously desirable for a quantum system employed in the QRC algorithm. The physical situation is quite different in the localized regime. The flow of information through the system is hindered by local conserved quantities supported by only parts of the reservoir (local integrals of motion). Local observables are strongly impacted by these conserved quantities, whereas the influence from the partial trace is minimal. The reservoir preserves its initial state due to reduced information erasure. This means that the convergence (as shown in Sec. 3.2) needed for forecasting is not given by the reservoir. Hence, the conserved quantities do not exhibit uniform dynamics, exemplified in Fig. 11. Although information is introduced into the system, its propagation to other qubits is severely limited. Local observables, which are insensitive to the qubit employed for encoding, exhibit no discernible dependency on the input sequence. This scenario is exemplified in Fig. 12. Of course, such behaviour is undesirable for quantum systems used in QRC.

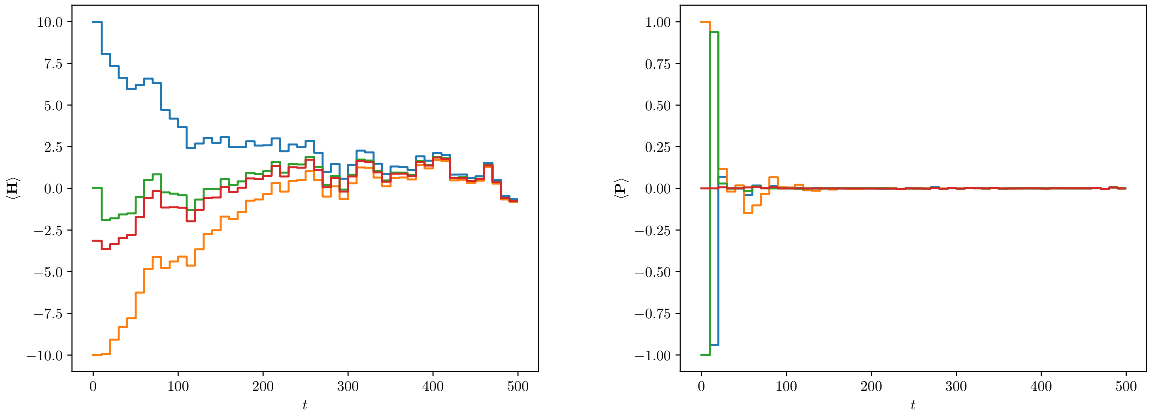


Fig. 9: $\langle\mathbf{H}\rangle$ and $\langle\mathbf{P}\rangle$ in time for 10-qubit systems for different initial states (single realization) for a Hamiltonian chosen in the ergodic phase of the transverse-field Ising Hamiltonian plus onsite disorder ($h = 2$, $W = 0.05$) injected with the same randomly chosen sequence according to Eq. 2.25 with unit time step $\tau = 10$. The different initial conditions are: all qubits in the zero state (blue), all qubits in the zero state (orange), the first half of the qubits in the zero state and first half of the qubits in the one state (green), the first half of the qubits in the zero state and first half of the qubits in the one state in the x-basis (red).

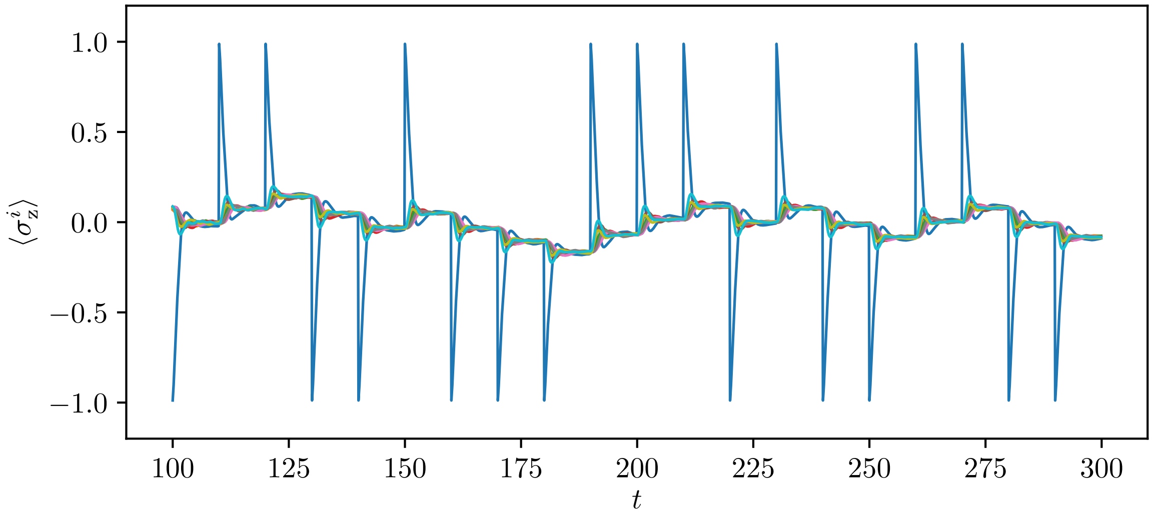


Fig. 10: Dynamics of observables $\langle\sigma_z^i\rangle$ (with $i \in \{1, 2, \ldots, 10\}$) with a binary input for a 10-qubit system (single realization) in the ergodic phase of the transverse-field Ising Hamiltonian plus onsite disorder ($h = 2$, $W = 0.05$) with unit time step $\tau = 10$.

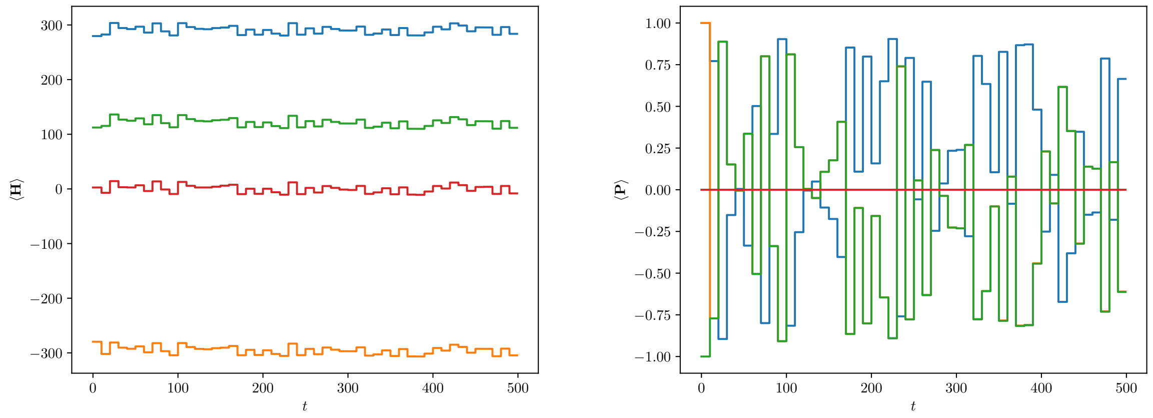


Fig. 11: $\langle \mathbf{H} \rangle$ and $\langle \mathbf{P} \rangle$ in time for 10-qubit systems for different initial states (single realization) for a Hamiltonian chosen in the localized phase of the transverse-field Ising Hamiltonian plus onsite disorder ($h = 50$, $W = 100$) injected with the same randomly chosen sequence according to Eq. 2.25 with unit time step $\tau = 10$. The different initial conditions are: all qubits in the zero state (blue), all qubits in the zero state (orange), the first half of the qubits in the zero state and first half of the qubits in the one state (green), the first half of the qubits in the zero state and first half of the qubits in the one state in the x-basis (red).

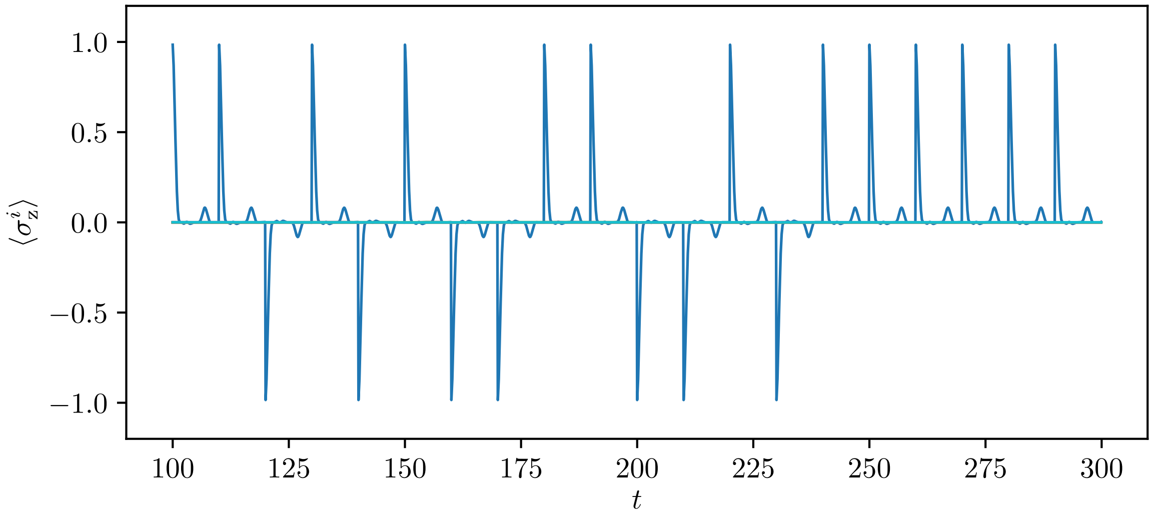


Fig. 12: Dynamics of observables $\langle \sigma_{\mathrm{z}}^{i} \rangle$ (with $i \in \{1, 2, \ldots, 10\}$) with a binary input for a 10-qubit system (single realization) in the localized phase of the transverse-field Ising Hamiltonian plus onsite disorder ($h = 0.05$, $W = 0.05$) with unit time step $\tau = 10$.

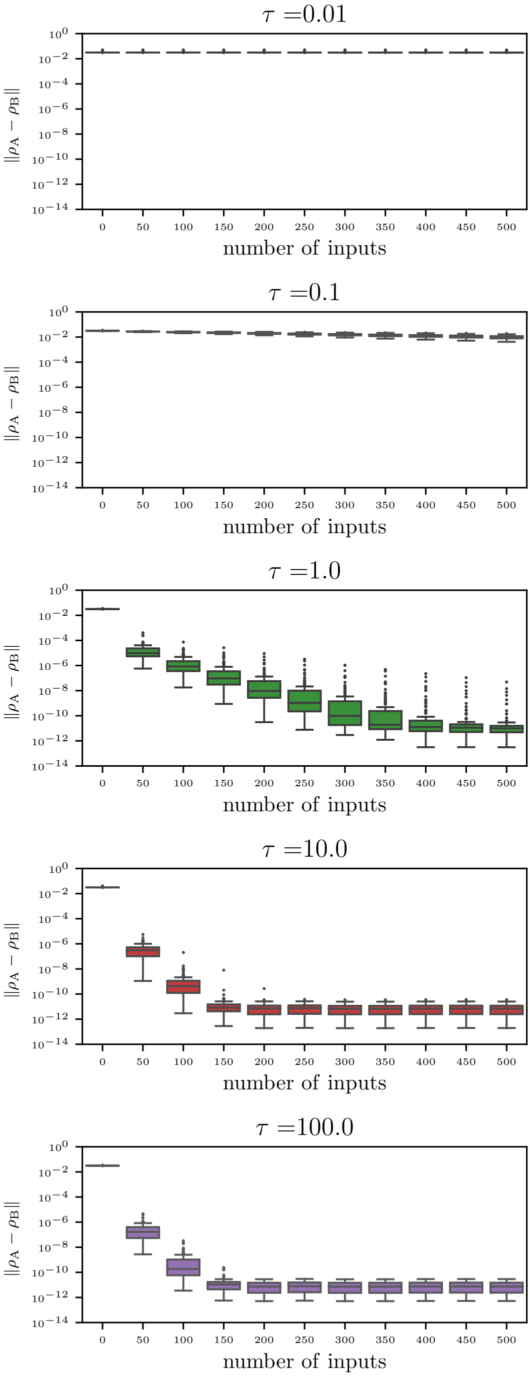
3.2 Echo state property

To be suitable for RC, a network must exhibit the echo state property, also known as the convergence property (which is closely related to the fading-memory) [60]. The reservoir must forget its initial state after repeated injections and subsequent evolutions. A consequence of this is that the output layer only depends on the recently injected input. This echo state property is also important for QRC [30]. The convergence in case of two quantum systems represented by the states ρ_{A} and ρ_{B} (density matrix representation) can be checked by observing the distance $\|\rho_{\mathrm{A}} - \rho_{\mathrm{B}}\|$ (defined by the Frobenius norm) after numerous applications of the previous defined injection and evolution procedure (Eq. 2.25). If the distance goes to zero, or, simply put, if the states converge to each other, the system exhibits the convergence property. The Frobenius norm of a matrix $\mathbf{M}$ is defined by the equation

$$\|\mathbf{M}\| = \sqrt{\mathrm{Tr}(\mathbf{M}^\dagger \mathbf{M})}. \tag{3.4}$$

Whether the system exhibits this property depends on the unit time step size τ between input injections and the phase of the system. The echo state property is evaluated by measuring the trace distance between two different initial states after each is injected according to Eq. 2.25 (using the unitary operator defined in Eq. 3.2). For this purpose, a sequence of uniformly randomly drawn numbers from the interval [0,1] is used. This is done multiple times for each investigated hyperparameter combination (τ, h, and W) to get statistically relevant results. Different random starting states (each with an average distance of 0.03) are used for each of these observed distances. In [30] the connection between the phase of the system and the echo state property for 10-qubit systems is examined.

In the following paragraph, these results are reproduced for 10-qubit systems, followed by the same investigation conducted for 4-qubit systems. The focus is on 4-qubit systems because these are used to produce the results on the prediction performance of QRC using small quantum systems presented in Chapter 5. The results are shown as boxplots to make the statistics more visable. The boxes represent the 25%-75% percentile range of the data and the line in the middle of the box shows the median of the data, i.e. 50% of the observed distances are below this value. The extended lines showcase the largest and smallest observations that fall within a distance of 1.5 times of the IQR (Interquartile range) of the data. Black dotes represent the outliers that are outside of this range. The investigation of the 10-qubit system is started with the ergodic phase of the system. In Fig. 13, boxplots of the convergence are shown for 5 different unit time steps (covering the interval $[10^{-2}, 10^{2}]$) for a 10-qubit system with $h = 2.0$ and $W = 0.05$. Each box represents 100 realizations and the distance is plotted in dependency on the number of injected inputs. This figure shows that the states do not converge for τ =0.01 and τ =0.1. The two states do converge to each other for τ =1, τ =10, and τ =100 after a few hundred inputs. The same investigation is carried out for the localized phase of the system ($h = 50$ and $W = 100$). The resulting boxplots are presented in Fig. 14. They show no convergence for all unit time steps. These results agree with the results of [30]. To fulfill the echo state property, the systems should be chosen in the ergodic phase and an appropriate unit time step τ should be selected. In Figs. 15 and 16, the results of the same numerical experiment with 4 qubits are presented. One can see that a similar behaviour is observed as for 10 qubits: In the ergodic phase and with enough time between successive inputs the states converge, i.e., the echo state property, is fulfilled. Choosing a system in the localized phase or selecting a unit time step τ that is too small results in the opposite behaviour. A key difference that can especially be observed for 4 qubits is that outliers can be observed where the states do not converge. This means that for certain realizations of the Hamiltonian, even in the ergodic phase and with the right unit time step size, the echo state property is not fulfilled.


τ =0.01

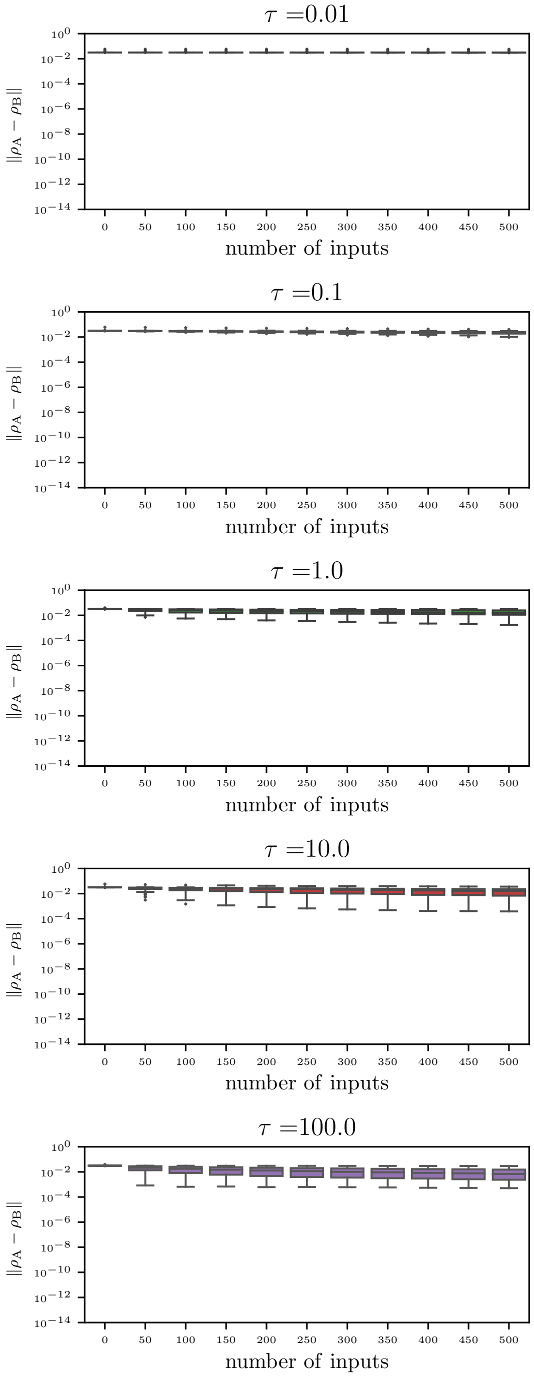
τ =0.1

τ =1.0

τ =10.0

τ =100.0

Fig. 13: Boxplots (100 realizations) of the convergence of two random states for different unit time steps τ dependent on the length of the input sequence for quantum systems of 10 qubits and their interaction described by the transverse-field Ising Hamiltonian (Eq. 3.1) plus onsite disorder with h=2 and W=0.05 (ergodic phase).


τ =0.01

τ =0.1

τ =1.0

τ =10.0

τ =100.0

Fig. 14: Boxplots (100 realizations) of the convergence of two random states for different unit time steps τ dependent on the length of the input sequence for quantum systems of 10 qubits and their interaction described by the transverse-field Ising Hamiltonian (Eq. 3.1) plus onsite disorder with h=50 and W=100 (localized phase).

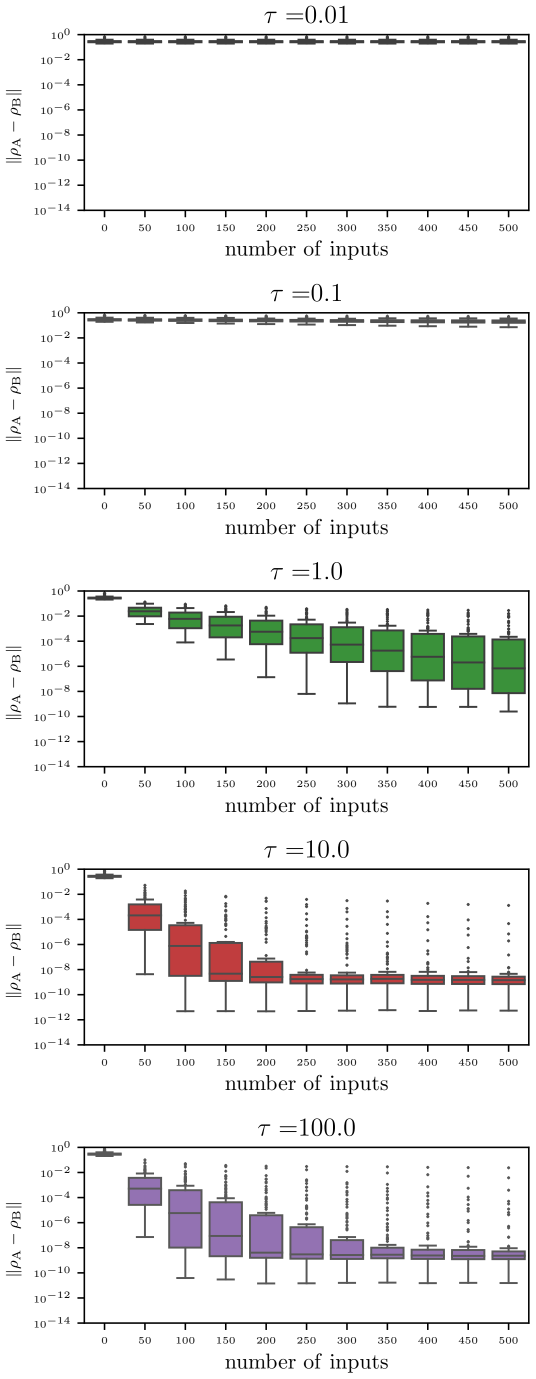


Fig. 15: Boxplots (100 realizations) of the convergence of two random states for different unit time steps τ dependent on the length of the input sequence for quantum systems of 4 qubits and their interaction described by the transverse-field Ising Hamiltonian (Eq. 3.1) plus onsite disorder with h=2 and W=0.05 (ergodic phase).

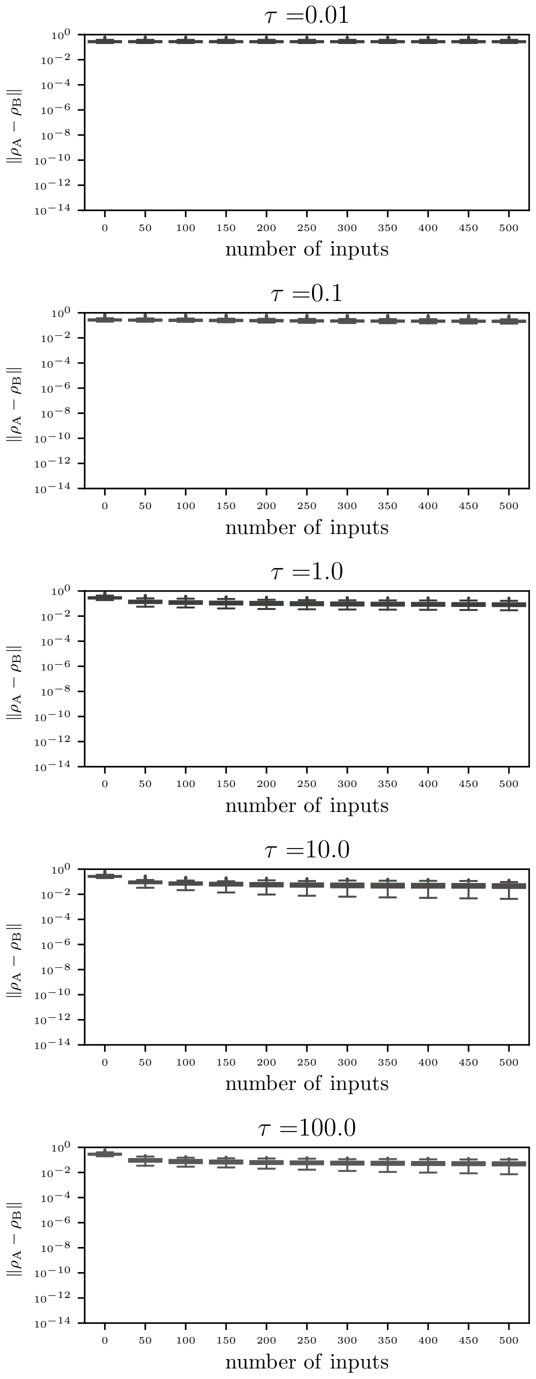


Fig. 16: Boxplots (100 realizations) of the convergence of two random states for different unit time steps τ dependent on the length of the input sequence for quantum systems of 4 qubits and their interaction described by the transverse-field Ising Hamiltonian (Eq. 3.1) plus onsite disorder with h=50 and W=100 (localized phase).

3.3 Comment: Normalization of the time series

As outlined in the algorithm description (Sec. 2.3), the time series must be scaled into the interval $[a, b]$ with 0$\leq a < b \leq 1$. For each dimension of the forecasted time series, one qubit is initialized to represent the corresponding dimension of the current time step. Fig. 17 depicts the encoding space on the Bloch sphere for different parameters a and b. The initialization in a pure state guarantees that the Bloch vectors have length one and therefore extend from the origin to the surface of the sphere. If the maximal interval [0,1] is chosen, the encoding space becomes a line connecting the zero state (top) to the one state (bottom). This line lies on the surface of the Bloch sphere and is characterized by a rotation from 0° ($|0\rangle$) to 180° ($|1\rangle$) about the y-axis passing over the $|+\rangle$ state. Furthermore, the figure indicates that restricting the interval $[a, b]$ limits the encoding space on the Bloch sphere.

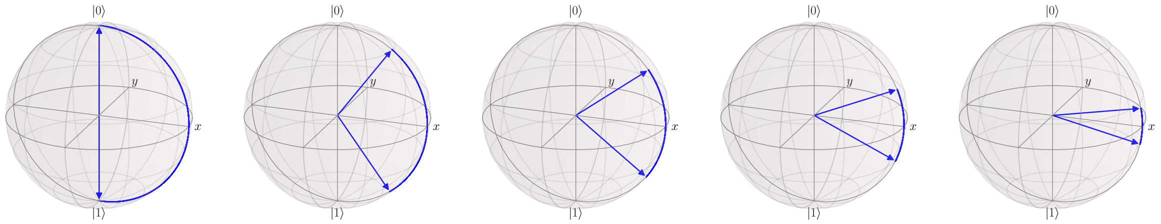


Fig. 17: Bloch spheres showing the encoding space used in Eq. 2.25 for different normalizations (i.e., $[a, b]$ chosen differently). From left to right: [0.00,1.00], [0.10,0.90], [0.20,0.80], [0.30,0.70], and [0.40,0.60]. In the plots, the Bloch vectors of the minimal and the maximal value of the interval are shown. The lines between the vectors represent the space into which the coefficients of the time series are encoded.

3.4 Comment: Quantum reservoir output

The enlargement of the output vector, discussed in this thesis, is a key component of the QRC algorithm. One method to achieve this is through temporal multiplexing. An illustrative example follows to demonstrate method. The determination of true nodes is based on expectation value measurements. In the ensuing example, a quantum system with 5 qubits and 4 evolution and measurement processes is analyzed ($V = 4$). The true nodes are limited to the expectation values $\langle \sigma_z^i \rangle$ where $i \in \{1, 2, 3, 4, 5\}$. Fig. 18 graphically presents one unit time step τ. The various coloured curves correspond to the expectation values plotted against time. At the beginning of the interval, the input (in this case 0) is injected into the first qubit (since the input is one-dimensional). True nodes are determined at times $t \in \{t_0 + \frac{\tau}{4}, t_0 + \frac{2\tau}{4}, t_0 + \frac{3\tau}{4}, t_0 + \tau\}$ and are assembled into one vector. The coloured circles lying on the curves and within the vector illustrate this process.

Fig. 19 sketches the most general construction of the reservoir output vector considered in this thesis. Each coloured circle represents a measured node, with different colours indicating different reservoirs, and the colour gradient represents different true nodes. At step k, each quantum reservoir generates an $n \times V$-dimensional vector. These r vectors are concatenated into a single $r \times n \times V$-dimensional vector $\mathbf{p}(k)$. By including all powers up to G of the coefficients of $\mathbf{p}(k)$ and a bias term, the output vector $\mathbf{q}(k)$ is produced. This is the final output vector of step k and it is a $1 + G \times r \times n \times V$-dimensional vector. Meaning that the number of nodes increases from n to $1 + G \times r \times n \times V$. However, this is not the most general construction of reservoir outputs when considering the combination of temporal multiplexing and spatial multiplexing. The most universal definition would consider the possibility of varying types (and hence amounts) of true nodes n_i as well as varying amounts of evolution and measurement processes V_i for each reservoir. In such a case, $\mathbf{p}(k)$ would be a $\sum_{i=1}^{r} n_i \times V_i$-dimensional vector. In this thesis, the same true nodes and the same number of measurement and evolution processes are assumed for each reservoir to limit the extensive hyperparameter space of the model, thereby making hyperparameter optimization (for small quantum systems) computationally feasible. Additionally, other choices of the non-linear readout function $\mathrm{F}_{\mathrm{res}}$ could be considered, which would also change the dimension of the output vector in a different way.

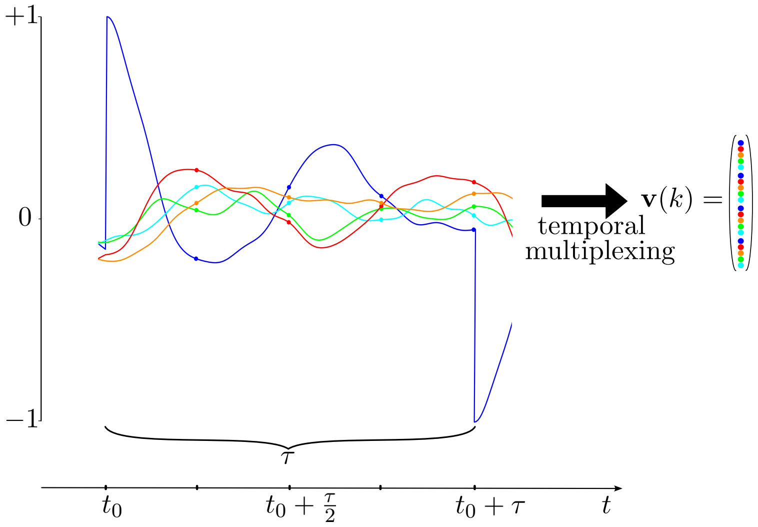


Fig. 18: Schematical illustration of the creation of the output vector by temporal multiplexing.

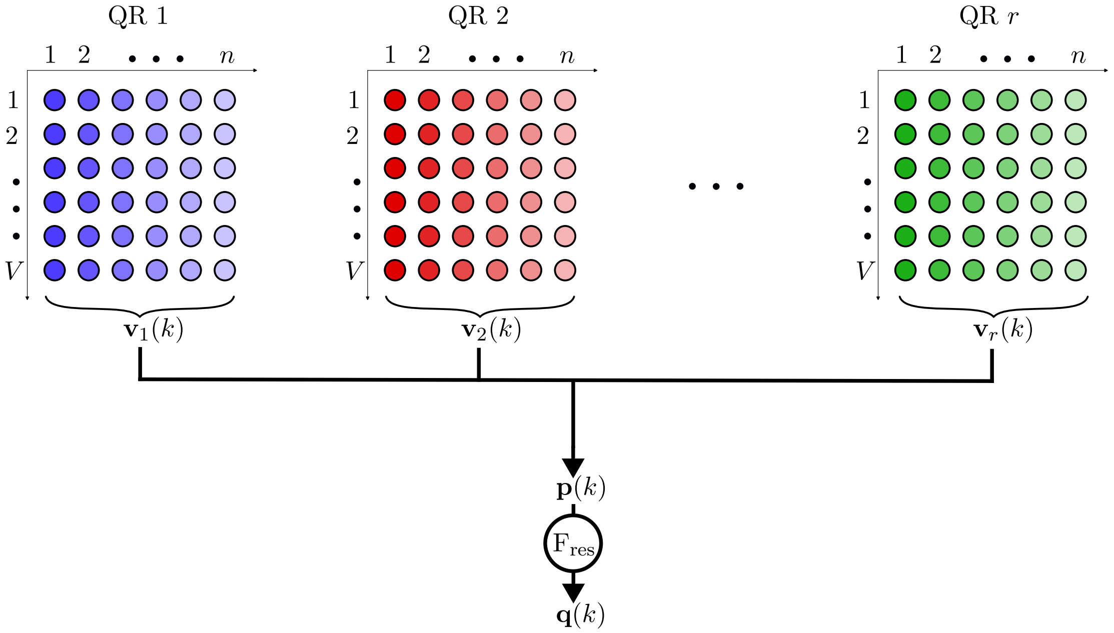


Fig. 19: Schematical illustration of the final output vector creation.

3.5 Comment: Emerging non-linearity

Quantum mechanics is fundamentally a linear theory. Initially, this might suggest that linear dynamics are unsuitable for machine learning tasks, which require non-linearity. How non-linearity enters the model is highlighted using the ensuing example. Suppose a two-dimensional discrete time series $\mathbf{x}(t)$ is encoded into a quantum system of N qubits as outlined by Eq. 2.25, with unitary evolution $\mathbf{U}$ between the encoding steps. Initially, the system is prepared in the state

$$\rho(0) = (|0\rangle\langle 0|)^{\otimes N}. \tag{3.5}$$

The first step of encoding involves initializing two qubits (labeled 1 and 2) in the state defined by Eq. 2.26. This encoding step results in the state

$$\rho_{1.\text{step}} = \begin{pmatrix} 1 - x_1^1 & \sqrt{1 - x_1^1}\sqrt{x_1^1} \\ \sqrt{1 - x_1^1}\sqrt{x_1^1} & x_1^1 \end{pmatrix} \otimes \begin{pmatrix} 1 - x_1^2 & \sqrt{1 - x_1^2}\sqrt{x_1^2} \\ \sqrt{1 - x_1^2}\sqrt{x_1^2} & x_1^2 \end{pmatrix} \otimes \mathrm{Tr}_{1,2}(\rho(0)). \tag{3.6}$$

Here, x_i^j denotes the j-th dimension of the i-th step of the discrete time series $\mathbf{x}(t)$. Afterwards, the unitary evolution is applied, which (if elected correctly, see Sec. 3.1) further spreads the non-linear combinations of the coefficients through the system and leaves the system in state $\rho(1)$. This quantum system and its subsystems exhibit non-linear dependency on the first step of the discrete time series. The second step is encoded into the system by initializing the quantum system in state

$$\rho_{2.\text{step}} = \begin{pmatrix} 1 - x_2^1 & \sqrt{1 - x_2^1}\sqrt{x_2^1} \\ \sqrt{1 - x_2^1}\sqrt{x_2^1} & x_2^1 \end{pmatrix} \otimes \begin{pmatrix} 1 - x_2^2 & \sqrt{1 - x_2^2}\sqrt{x_2^2} \\ \sqrt{1 - x_2^2}\sqrt{x_2^2} & x_2^2 \end{pmatrix} \otimes \mathrm{Tr}_{1,2}(\rho(1)). \tag{3.7}$$

The tensor products create terms (coefficients of the density matrix) that depend non-linearly on the coefficients from the first two steps of the discrete time series. This due to two reasons. The first reason is that the subsystem $\mathrm{Tr}_{1,2}(\rho(1))$ depends non-linearly on the coefficients of first step of the time series. The second reason is that by definition of the tensor product and the chosen states, the initialized qubits (1 and 2) non-linearly encode the information of the second step. This means that the tensor product between the states of the first two qubits and the rest of the system produces a state that depends non-linearly on all encoded steps up to this point. It is now obvious how repeating this process recurrently generates matrix terms that depend non-linearly on the coefficients of the time series up to the encoded step. The expectation values of the operator $\mathbf{O}$, determined by the equation

$$\langle \mathbf{O} \rangle = \mathrm{Tr}(\mathbf{O}\rho), \tag{3.8}$$

thereby exhibit non-linear dependencies on the coefficients of the time series up to the encoded step. This means that the coefficients of the output vector $\mathbf{q}(k)$ (the nodes) also have non-linear dependencies on the coefficients of the time series up to the encoded step.

Chapter 4

Chaotic Systems and Prediction Quality Measures

4.1 Chaotic Dynamical Systems

The QRC framework is designed to forecast dynamical systems from past data. In other words, the models are trained using a (training) dataset consisting of L consecutive time steps from a time series of a dynamical system, with the objective of forecasting the next N_{pred} time steps. The dynamical systems used to benchmark the introduced QRC algorithm are prototypical, synthetic, chaotic dynamical systems. Within this section, fundamental concepts of chaotic dynamical systems relevant to the thesis's context are introduced.

4.1.1 Definition

The following definition is composed of various chapters of [61] and [62]. A dynamical system is a system that evolves in time. Dynamical systems can be stochastic or deterministic. A stochastic system is one that evolves based on a random process (e.g., toss of a coin), and a deterministic system is one that is uniquely determined by its past and specific governing equations. In this thesis, only deterministic systems are considered. Systems can also be distinguished based on whether they evolve discretely or continuously. A dynamical system evolving deterministically in discrete time steps is represented by a d-dimensional vector $\mathbf{u}_k$ of step k, and the evolution of the state is determined by a map $M(\cdot)$ that maps the state of step k to the state of step $k+1$

$$\mathbf{u}_{k+1} = M(\mathbf{u}_k). \tag{4.1}$$

A system that evolves deterministically and continuously in time is described by a d-dimensional vector $\mathbf{u}(t)$ of time t and its evolution is governed by first-order differential equations. The differential equations are defined by a flow $F(\cdot)$:

$$\dot{\mathbf{u}}(\mathrm{t}) = F(\mathbf{u}(\mathrm{t})). \tag{4.2}$$

A deterministic dynamical system is called non-linear if its governing equations ($M(\cdot)$ or $F(\cdot)$) are non-linear in the system's variables. These deterministic, non-linear dynamical systems are often complex in the nature of its dynamics, and they can display chaotic behaviour in some cases. There is no universal definition of chaos. J.C. Sprott [61] states, "chaos is the aperiodic, long-term behavior of a bounded, deterministic system that exhibits sensitive dependence on initial conditions." These complex properties make such dynamical systems the ideal benchmark for the forecasting of dynamical systems and demonstrate the potential of QRC in this area.

4.1.2 Dataset generation: Numerical simulation using RK4

In the case of discrete, deterministic dynamical systems, the evolution of the system (time series) can be generated directly by iterative applications of its defining map (Eq. 4.1) once an initial state $\mathbf{u}_0$ is specified. In the continuous case, the situation is more difficult. Most interesting dynamical systems are defined by first order differential equations that are not analytically solvable. Nevertheless, approximate solutions can be obtained by numerical methods. For this thesis, the fourth-order Runge-Kutta method (RK4) [63, 64, 61] is used to numerically simulate continuous-time dynamical systems. The continuous time is discretized into time steps of size

Δt and the flow 4.2 effectively approximated Eq. 4.2 by a map $M_{\text{rk4}}(\cdot)$ (RK4 iterator) mapping the system state from time step k to time step $k+1$:

$$\mathbf{u}_{k+1} = M_{\text{rk4}}(\mathbf{u}_k). \tag{4.3}$$

The RK4 iterator is defined as:

$$M_{\text{rk4}}(\mathbf{u}_k) = \mathbf{u}_k + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \tag{4.4}$$

with $\mathbf{k}_1 = F(\mathbf{u}_k)\Delta t,\ \mathbf{k}_2 = F(\mathbf{u}_k + \mathbf{k}_1/2)\Delta t,\ \mathbf{k}_3 = F(\mathbf{u}_k + \mathbf{k}_2/2)\Delta t$ and $\mathbf{k}_4 = F(\mathbf{u}_k + \mathbf{k}_3)\Delta t$. Here, the discrete time index $k = 0, 1, \ldots$ corresponds to the real time $t_0 + k\Delta t$. By iteratively applying the RK4 map, the time series can be generated once an initial state is specified.

The datasets used for benchmarking are obtained by specifying an initial state $\mathbf{u}_0$ and applying the map iteratively $1000 + N_{\text{traj}} \cdot N_l$ times. From the resulting discrete time series, the first 1000 steps are discarded, and the remaining time series is split into N_{traj} time series, each of length N_{l}. For this study, the prediction performance of the QRC model is investigated by forecasting eight three-dimensional chaotic systems that are defined in the appendix (Appendix A). The chosen starting points and system parameters are also specified in the appendix. The parameters $N_{\text{traj}} = 1000$ and $N_{\text{l}} = 25000$ are chosen to obtain the trajectories. This means that for all eight chaotic systems, 1000 trajectories, each consisting of 25000 steps, are generated. The second half of these 1000 trajectories is used to calculate the real climate of the systems (to generate values for relative comparison). The results of these calculations are shown in Table 2. The first 500 trajectories are used as benchmark datasets for the studies presented in Chapter 5.

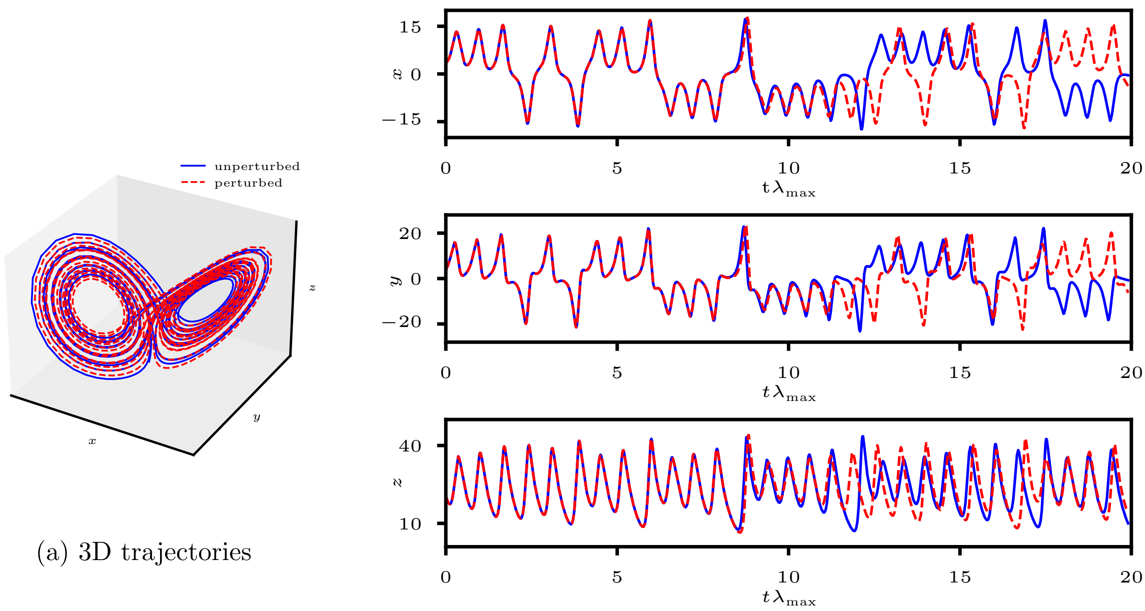
4.1.3 Example of a chaotic system: Lorenz-63

In the following the Lorenz-63 system is presented as a prominent example of a three-dimensional continuous-time system that displays chaotic dynamics for certain parameter ranges. The Lorenz system was introduced to model atmospheric convection in the 1960s by Edward Lorenz [65]. The three-dimensional state $\mathbf{u}(t) = [x(t), y(t), z(t)]^{\text{T}}$ evolves according to the differential equations

$$\begin{gathered}\dot{x}(t) = \sigma(y - x), \\ \dot{y}(t) = x(\rho - z) - y \\ \text{and} \\ \dot{z}(t) = xy - \beta z.\end{gathered} \tag{4.5}$$

For the (typical) parameters $\rho = 28$, $\sigma = 10$, and $\beta = 8/3$ the system exhibits chaotic behaviour. The Lorenz-63 system is simulated (RK4 method) in this thesis with Δt=0.02 and the initial state chosen is $\mathbf{u}_0 = [0, 0.01, 9]^{\text{T}}$. The sensitivity to the initial conditions (for this parameter choice) is illustrated in Fig. 20, which shows the evolution of the Lorenz-63 system for two initially close states. After approximately 10 Lyapunov times (defined in Sec. 4.2.1) have past the difference between the two systems becomes visually noticable and afterwards the systems become completely desynchronized.

The Lorenz-63 system has been presented here in more detail to demonstrate some of the key concepts of chaotic systems and introduce the types of systems that are forecasted in this thesis. The remaining chaotic systems used to evaluate the performance of QRC are defined, and the parameter choices are listed in the appendix (Appendix A).



(a) 3D trajectories

(b) Individual dimensions against time

Fig. 20: Evolution of the Lorenz system for two initially close states (perturbed by a vector $\delta = [10^{-4}, 0, 0]$). a) 3D trajectories of the evolution of the perturbed and unperturbed system for 1100 time steps. b) The individual dimensions of the perturbed and unperturbed system plotted against time for 1100 time steps.

4.2 Prediction performance measures

To evaluate the quality of the predictions different measures are used. These measures are chosen to sufficiently analyze the quality of the exact short-term prediction and the reproduction of the long-term statistical properties (climate) of the systems. Past research in the area of QRC often fell short in evaluating the climate of the predicted system. The measures used for the evaluation follow previous studies [66, 67, 68, 69] investigating ”classical” RC and related approaches.

4.2.1 Forecast horizon

The forecast horizon (also called valid time) is calculated to evaluate the short-term prediction capabilities of the model. It measures the time for which the continued time series matches the true continuation of trajectory very closely. The forecast horizon is the elapsed time while the normalized, time-dependent error $e(t)$ [70] between the continued time series $\mathbf{y}_{\text{pred}}(t) = \{\mathbf{o}_k\}_{k=L+1}^{L+N_{\text{pred}}}$ and the true continuation $\mathbf{y}(t) = \{\mathbf{u}_k\}_{k=L+1}^{L+N_{\text{pred}}}$ is smaller than a threshold value $e_{\max}$. The normalized time-dependent error is defined as

$$e(t) = \frac{\|\mathbf{y}(t) - \mathbf{y}_{\text{pred}}(t)\|}{\langle \|\mathbf{y}(t)\|^2 \rangle^{1/2}}. \tag{4.6}$$

Here, $\langle . \rangle$ denotes the average over all N_{pred} steps and $\|.\|$ is the L2-norm. It is determined for how many consecutive steps s_{v} (starting with the first forecasted state) the relation $e(t) < e_{\max}$ holds. In this thesis the threshold is chosen to be $e_{\max}$=0.4. The forecast horizon t_{v}, in units of the Lyapunov times $1/\lambda_{\max}$ of the dynamical system, is obtained by calculating

$$t_{\text{v}} = \Delta t \cdot s_{\text{v}} \cdot \lambda_{\max}. \tag{4.7}$$

Here, Δt is the time between two successive steps of the discretized time series, and $\lambda_{\max}$ is the largest Lyapunov exponent (defined in Sec. 4.2.3) of the system. The forecast horizon is measured in Lyapunov times to obtain a measure that is more comparable across different dynamical systems.

4.2.2 Correlation dimension

One aspect of the long-term behaviour of a dynamical system is its structural complexity. The correlation dimension is a measure that quantifies structural complexity by measuring the dimensionality of the space populated by the trajectory [71]. This measure is based on the discrete version of the correlation integral

$$C(R) = \lim_{M\to\infty} \frac{1}{M^2} \sum_{i,j=1}^{M} \Theta(R - \|\mathbf{x}_i - \mathbf{x}_j\|). \tag{4.8}$$

Here, Θ represents the Heavyside function. $C(R)$ calculates the mean probability that two states in phase space are closer than a threshold distance R for different time steps. For a self-similar strange attractor the correlation integral in certain range of the threshold R is defined by a power-law relationship:

$$C(R) \propto R^v. \tag{4.9}$$

The scaling exponent v defines the correlation dimension of the attractor. The correlation dimension belongs to the measures of fractal dimensionality introduced by Mandelbrot in 1967 [72]. The advantage of this fractal measure is that it can be determined (without the knowledge of the underlying equations) with a relatively small amount of datapoints available. The Grassberger Procasccia algorithm [73] is used to calculate the correlation dimension from data. The correlation dimension is determined for all eight chaotic systems for 500 different parts (20000 steps) of the attractor. In Table 2 the mean values and the standard deviation of the calculation together with the real literature values are presented. The main point of interest is to make relative comparisons between actual trajectories and the predictions. Therefore, obtaining correlation dimensions that closely match the values found in the literature is not the main priority.

4.2.3 Largest Lyapunov exponent

Another characteristic of the long-term climate of a dynamical system is its temporal complexity. The most appropriate way to quantify the temporal complexity of a dynamical systems is to analyze its Lyapunov exponents, characterizing the system's development in time. A d-dimensional dynamical system has d Lyapunov exponents that determine the average rate of divergence of nearby points in phase space. By measuring the average rate of exponential growth of a small perturbation in each direction in phase space, the Lyapunov spectrum does measure how sensitive the system is to its initial conditions. A dynamical system exhibits chaos if one of its Lyapunov exponents is positive, and the magnitude of the exponent determines the time scale at which the system becomes unpredictable [74, 75]. The largest Lyapunov exponent $\lambda_{\max}$ is linked to the direction in which the divergence occurs most rapidly,

$$d(t) = c \cdot e^{\lambda_{\max} t}. \tag{4.10}$$

In this thesis, measuring the largest Lyapunov exponent suffices, because of its dominant influence over the dynamics. This constraint also has a computational advantage, because $\lambda_{\max}$ can be easily calculated from data using the Rosenstein algorithm [76]. Again, the main interest is not high accuracy (regarding the reproduction of literature values) but in creating a measure for relative comparisons of actual and predicted trajectories of the investigated systems. The largest Lyapunov exponent is calculated for 500 true trajectories (each 20000 steps) for all eight chaotic systems (defined in Appendix A). In Table 2 the mean values and the standard deviations of this calculations are presented together with the true values (taken from literature).

system	$\lambda_{\max}$ (literature)	$\lambda_{\max}$ (calculated)	υ (literature)	υ (calculated)
Lorenz-63	0.9056	0.91 ± 0.02	2.068±0.086	2.052 ± 0.009
Chen	2.0272	2.02± 0.05	2.147±0.117	2.145± 0.008
Chua	0.3271	0.341± 0.007	2.125 ± 0.098	1.75± 0.01
Halvorsen	0.7899	0.78 ± 0.02	2.110 ± 0.095	2.106 ± 0.006
Rössler	0.0714	0.072 ± 0.004	1.991 ± 0.065	1.82 ± 0.02
Rucklidge	0.193	0.194±0.006	2.108 ± 0.095	1.93± 0.02
Thomas	0.0349	0.033 ± 0.001	1.843 ± 0.075	1.76 ± 0.04
WINDMI	0.0755	0.074± 0.004	2.108 ± 0.095	1.88± 0.02

Table 2: Mean values calculated over 500 realizations with standard deviation and the literature values of the maximal Lyapunov exponent and correlation dimension for all eight chaotic systems investigated in this thesis (defined in Appendix A). The literature value of the largest Lyapunov exponent of the Rucklidge system is taken from [77]. All other literature values are taken from [61].

Chapter 5

Forecasting 3D Chaotic Systems with Minimal QRC

One of the variable parameters in the model is the size (number of qubits N) of the quantum system used in the model. This chapter of the thesis explores the predictive power if the smallest theoretically allowed (”minimal”) quantum systems are used as reservoirs. These small quantum reservoirs are particularly interesting due to the significant size restrictions of current and near-future quantum devices. This chapter begins by defining the objectives of the investigation and describing the setup of the numerical experiments (Sec. 5.1). It then presents the results of these experiments (Sec. 5.2) and ends with the conclusions about QRC models that employ small quantum systems, derived from these numerical experiments and their results (Sec. 5.3).

5.1 Objective and methodology

As outlined in Sec. 2.3, a theoretical minimum number of qubits is required to run the model. The dimension (d) of the forecasted time series limits the quantum system size, making it necessary for the number of qubits N to be at least $d+1$. This thesis explores the predictive power of QRC for systems at this minimal size. In the appendix (Appendix C.1), example predictions obtained with bigger quantum systems are shown. The thesis has two primary objectives. The first objective is to demonstrate the impact of introduced changes to the framework (hyperparameters r, G, V, β, and $[a, b]$) on the forecasting performance of the machine learning models when using small quantum systems. The second objective is to test and to optimize the machine learning framework for various synthetic chaotic systems, evaluating its ability to forecast these systems with minimally-sized quantum reservoirs. The study emphasizes ensuring statistical relevance and examines both short- and long-term prediction accuracy, aspects that have often been overlooked in prior QRC studies. The following paragraph outlines the structure of the numerical experiments.

For each of the eight three-dimensional chaotic systems used for benchmarking in these studies (defined in Appendix A), 500 trajectories, each consisting of 25000 discrete, consecutive time steps, are generated (Sec. 4.1.2). These trajectories serve as benchmark datasets for the evaluation of the prediction performance. The time series are standardized before being used as benchmarks. This requires that the predicted continuations be scaled inversely for analysis and comparison with the actual continuations. The results of the following studies are obtained by evaluating the model N_{stat} times for each hyperparameter configuration. During each of these N_{stat} evaluations, different segments of the chaotic attractors (from the generated 500 trajectories) and different Hamiltonian parameter draws are used for training and forecasting to ensure a statistically significant performance evaluation. Each evaluation involves training the model with $N_{\text{sync}} + N_{\text{train}} + 1 = L$ consecutive trajectory steps and subsequently continuing the time series for N_{pred} steps. For each of the forecasted trajectories, the forecast horizon, the largest Lyapunov exponent, and the correlation dimension are calculated as described in Sec. 4.2. These measures are utilized to compare predictions for various hyperparameter configurations for each chaotic system and to assess the model’s performance, focusing on short-term accuracy and climate reproduction. For each numerical experiment, the short-term prediction quality is illustrated using boxplots. The y-axis of these plots represents the forecast horizon of the predictions. Each box in the boxplot corresponds to one of the investigated hyperparameter configurations in the presented study. The boxes indicate the 25%-75% percentile range of the

N_{stat} calculated forecast horizons, with the line in the middle representing the median, i.e., 50% of the forecast horizons fall below this value. The extended lines (whiskers) display the largest and smallest observations within a distance of 1.5 times the interquartile range (IQR) from the edges of the box. Black dots denote outliers outside this range. The distribution of the predicted climate is assessed using scatter plots for each analyzed combination of hyperparameters. For each of the N_{stat} trajectories, the determined maximal Lyapunov exponent and the correlation dimension are plotted as points in the scatter plots. The black ellipses in the plots indicate the three standard deviation errors of the correlation dimension and the largest Lyapunov exponent, derived from simulations of the actual systems (Table 2). Each plot contains a zoomed-in window (shown in the edge of the plot) centered around the mean values of the correlation dimension and largest Lyapunov exponent (x=υ,y=λ_{max}) that are determined from simulations. The windows extends $\pm 15\%\lambda_{max}$ in the y-direction and $\pm 5\%\upsilon$ in the x-direction.

5.2 Results

In this research, three-dimensional chaotic systems are forecasted, necessitating a minimal quantum system of 4 qubits. Consequently, all quantum systems used to generate the presented results consist of 4 qubits. Due to numerical limitations, analyzing the relationship between hyperparameters and performance is challenging, and a grid search over a large portion of the hyperparameter space is not feasible. Even for these small quantum systems, statistically valid and complete evaluations of the hyperparameter space are computationally too expensive. Therefore, some hyperparameters are kept fixed throughout the investigation. Each trained model is synchronized with 100 steps, followed by training on the next 2000 steps of the time series, and then the time series is continued for 20000 steps ($N_{\text{sync}} = 100$, $N_{\text{train}} = 1999$, and $N_{\text{pred}} = 20000$). Additionally, the determined expectation values are not treated as adjustable hyperparameters. In this study, $\langle \sigma_{\text{z}}^{i} \rangle$ and $\langle \sigma_{\text{z}}^{i} \sigma_{\text{z}}^{l} \rangle$ with $i, l \in \{1, 2, ..., N\}$ and $i < l$ are determined at each measurement step (n=10). Following the findings in [30] and the analysis of the system response and the echo state property in Sec. 3.1 and Sec. 3.2, the quantum system parameters W=2.0 and h=0.05, and the unit time step size $\tau = 20$ are selected. The echo state property of exactly this quantum reservoir (three-dimensional input) is analyzed in the appendix (Appendix B.1). The investigation presented in this thesis emphasizes on optimizing how the outputs of these quantum systems are utilized, rather than optimizing the quantum systems themselves. The section begins with an investigation of the effect of the normalization interval $[a, b]$ on the forecasting quality, followed by an analysis of the introduction of ridge regression into the machine learning model. Afterwards, the influences of the output vector enlargement methods are analyzed. These examinations of the effects of the introduced hyperparameters of the QRC model are conducted by forecasting the Lorenz-63 system. The final investigation is an analysis of the prediction quality of the model for all eight chaotic systems (defined in Appendix A) using hyperparameter configurations obtained through hyperparameter optimization.

5.2.1 Normalization

As described in Sec. 2.3 and Sec. 3.3, the time series is scaled to the interval $[a, b]$ where $0 \leq a < b \leq 1$. This implies that the forecasted time series must be inversely scaled and then undergo inverse standardization before being compared to the actual continuation. Subsequently, the effect of normalization on the forecasting ability of the model is analyzed. The hyperparameters $\beta = 10^{-10}$, r=1 and G=1 are kept constant and each combination of the hyperparameters $V \in \{1, 2, 3, 4, 5\}$ and $[a, b] \in \{[0.1, 0.9], [0.2, 0.8], [0.3, 0.7], [0.4, 0.6]\}$ is used to forecast the Lorenz-63 system. For each analyzed hyperparameter combination, $N_{\text{stat}} = 100$ is chosen, and the numerical experiments are carried out as described in Sec. 5.1. The 100 forecasted trajectories for each hyperparameter combination are analyzed by determining the measures defined in Sec. 4.2. In Fig. 21, a boxplot with forecast horizons for all investigated hyperparameter combinations is shown. The statistical distribution (100 realizations of the model) of the forecasted climate is presented for each hyperparameter combination in Fig. 22. It can be observed that increasing V, which leads to a greater dimension of $\mathbf{q}(k)$, increases the forecast horizon. The increase of V also leads to better climate predictions. Meaning that The Lyapunov exponent and the correlation dimension are frequently predicted with high accuracy, within a reasonable margin of deviation. Further investigations into the question how increasing the dimension of $\mathbf{q}(k)$ correlates with the prediction performance can be found in Sec. 5.2.3. The figures show that the normalization interval $[a, b]$ has a significant influence on both long- and short-term prediction quality. The optimal choice of $[a, b]$ is influenced by the remaining hyperparameter configuration. For this forecasting task (Lorenz-63), the short- and long-term prediction quality seem to benefit from a smaller encoding space on the Bloch sphere when the dimension of the output vector is small. The opposite is true for short-term prediction: as the dimension of the output vector increases, a larger encoding space results in longer forecast horizons. The accuracy of the climate prediction saturates for higher-dimensional output vectors, and no difference is visible for the different intervals $[a, b]$.

These results show that incorporating the hyperparameters a and b into the hyperparameter optimization process is a sensible strategy when aiming to find the optimal hyperparameter configuration for a specific forecasting task. Identifying the underlying reasons for these effects and conducting a more comprehensive evaluation of the continuous space $0 \leq a < b \leq 1$ is not within the scope of this thesis.

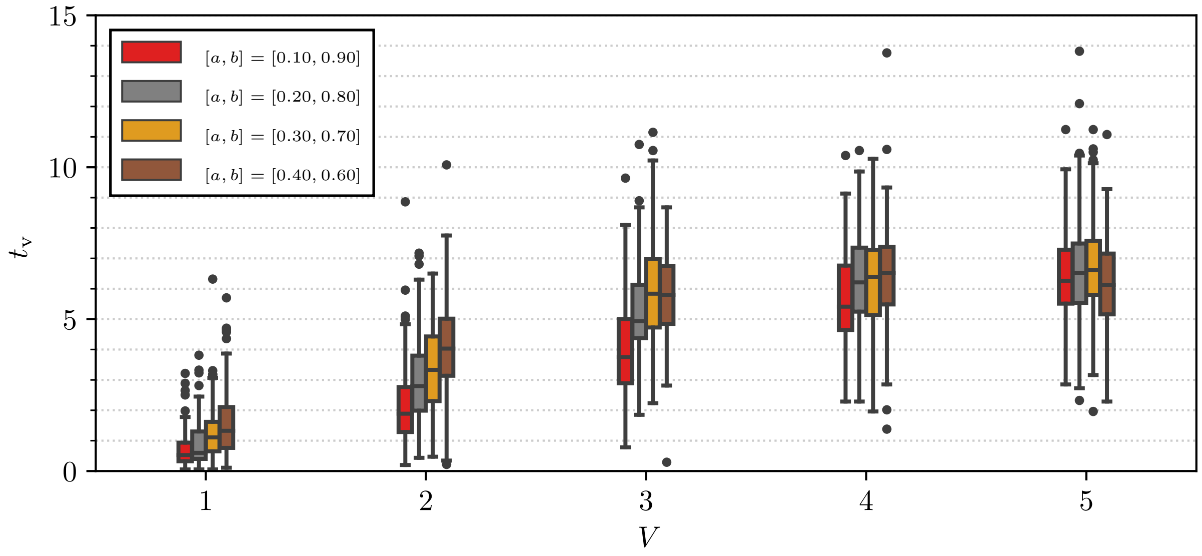


Fig. 21: Boxplot of the forecast horizon in Lyapunov times of 100 forecasted trajectories of the Lorenz-63 system for multiple combinations of the number of evolution and measurements (V$\in$ $\{1, 2, 3, 4, 5\}$) and the normalization interval ($[a, b]\in$ $\{[0.1, 0.9], [0.2, 0.8], [0.3, 0.7], [0.4, 0.6]\}$). The remaining hyperparameters are kept constant at $\beta = 10^{-10}$, r=1, and G=1.

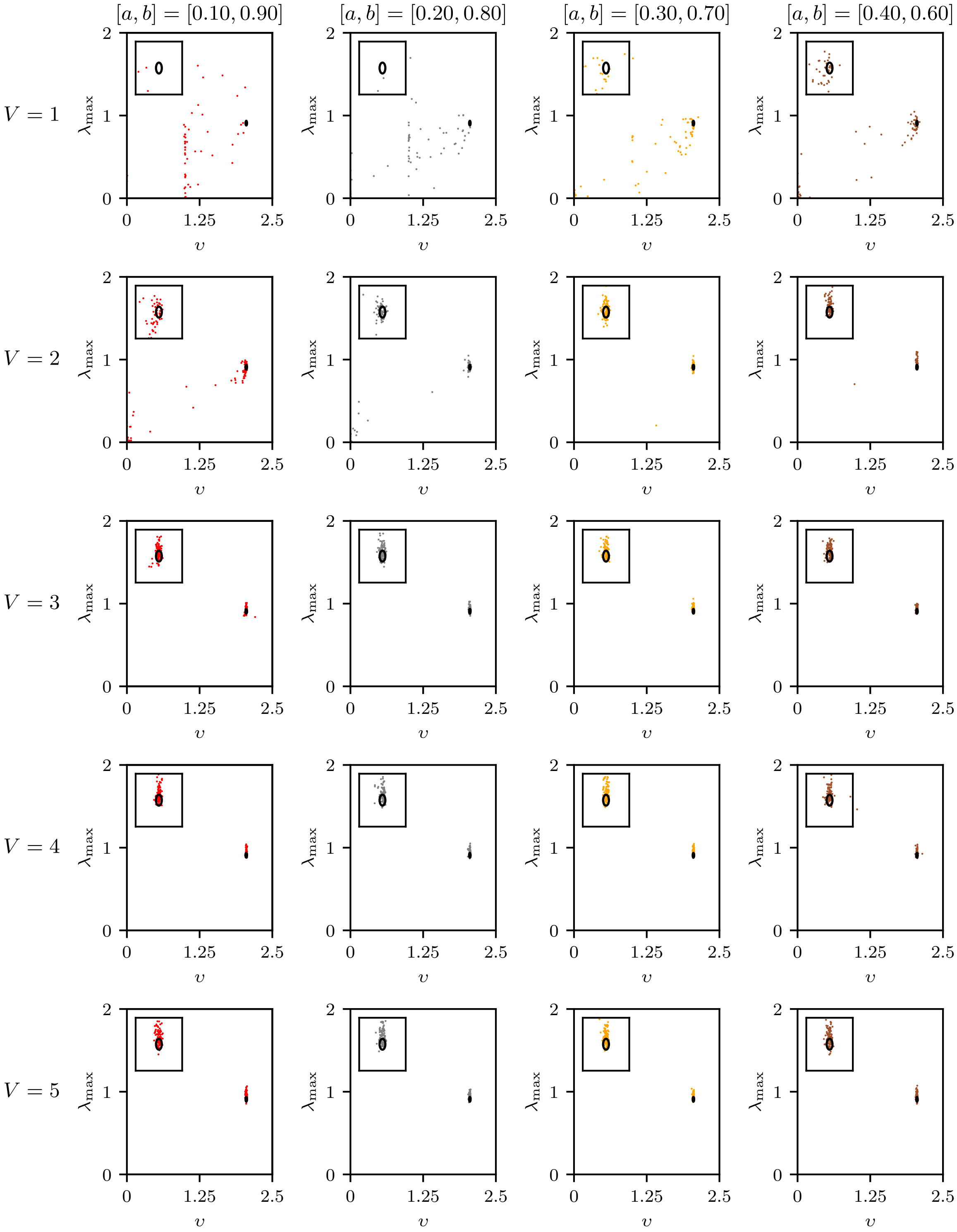


Fig. 22: Scatter plots of the predicted forecast horizons showing the predicted $\lambda_{\max}$ scattered against the predicted υ for each of the 100 realizations for various combinations of normalization intervals ($[a, b] \in \{[0.10, 0.90], [0.20, 0.80], [0.30, 0.70], [0.40, 0.60]\}$) and number of measurement and evolution processes ($V \in \{1, 2, 3, 4, 5\}$). The remaining hyperparameters are kept fixed at $\beta = 10^{-10}$, r=1, and G=1.

5.2.2 Ridge Regression

In the current literature [27, 28, 30], the Moore-Penrose pseudoinverse is commonly applied as the regression method to obtain the readout matrix $\mathbf{W}_{\text{out}}$ in QRC. In this approach, the mean-squared error between the training data and the QRC output $\mathbf{o}_k = \mathbf{W}_{\text{out}}\mathbf{q}(k)$ for $k \in [N_{\text{sync}} + 1, N_{\text{sync}} + 2, ..., N_{\text{sync}} + N_{\text{train}}]$ is minimized. In "classical" RC, the usage of ridge regression is common and makes a huge difference in the forecasting ability of the models. The regression parameter β is a highly influential hyperparameter of these models. In this part of the thesis, the influence of the regression parameter on prediction quality in QRC is investigated, and the relevance of the introduced regression method is examined. To analyze the influence of the regression parameter on the forecasting of the Lorenz-63 system using the QRC framework, the regression parameter β is varied for six different configurations of the hyperparameters V, G, r, and $[a, b]$ (presented in Table 3). This variation of the regression parameter β is performed over the set $\{10^{-25}, 10^{-24}, 10^{-23}, \ldots, 10^{-2}, 10^{-1}\}$. The model is trained for all of the hyperparameter combinations, and the training time series continued for 100 different parts of the attractor (N_{stat}=100). In Fig. 23, the resulting forecast horizons of the sweeps over the regression parameter for all hyperparameter configurations are depicted as boxplots. The predicted climates are evaluated for specific $\beta \in \{10^{-1}, 10^{-3}, 10^{-10}, 10^{-20}\}$ for all six configurations in scatter plots. For all configurations, these scatter plots can be found in Fig. 24. Several observations emerge from the analysis of these figures. The short-term as well as the long-term prediction qualities for configurations with fewer than 100 nodes (configuration 1 and configuration 3) tend to saturate and remain relatively independent of the regression parameter when the regression parameter β is small ($\beta \in [10^{-25}, ..., 10^{-9}]$). With an increase in β beyond this range, the forecasting ability deteriorates. For configuration 4 (dim($\mathbf{q}(k)$)=101), it is evident that there exists an optimal range ($\beta \in [10^{-13}, 10^{-9}]$) where the forecasting quality peaks. Choosing a smaller β often results in effective predictions, but occasionally outliers appear. These outliers fail to accurately capture the system's short- and long-term evolution. Selecting a larger β leads to a decline in prediction quality similar to the decline for large β observed in configuration 1 and configuration 3. Regarding the remaining configurations (configuration 2, configuration 5 and configuration 6), it is clear that the prediction quality is maximized with a specific choice of the regression parameter ($\beta \in [10^{-13}, 10^{-10}]$). Additionally, there is a noticeable decrease in prediction quality when β is set too small. In this case, the model fails to predict the dynamics of the system in many instances. Similarly, larger β values lead to diminished prediction quality, consistent with previous observations across all configurations. All these three configurations have more than 150 nodes (dim($\mathbf{q}(k)$) >150). An analysis of the presented results indicates that the significance of the regression parameter (not saturating for sufficiently small β) increases with the dimension of the output vector $\mathbf{q}(k)$. This intuitively makes sense: a larger output vector implies more nodes for training, which increases the model's susceptibility to overfitting and consequently reduces forecasting accuracy. This investigation highlights the necessity of integrating ridge regression into the model and including the regression parameter in the optimization process to obtain models with stable forecasting ability.

configuration label	V	r	G	$[a, b]$	dim($\mathbf{q}(k)$)
configuration 1	3	1	1	[0.10, 0.90]	31
configuration 2	4	2	3	[0.20, 0.80]	241
configuration 3	5	1	1	[0.20, 0.80]	51
configuration 4	5	1	2	[0.15, 0.85]	101
configuration 5	10	1	3	[0.10, 0.90]	301
configuration 6	4	1	4	[0.25, 0.75]	161

Table 3: Table of all configurations V, G, r, and $[a, b]$ and their labels used in the analysis of the influence of the regression parameter on the forecasting ability.

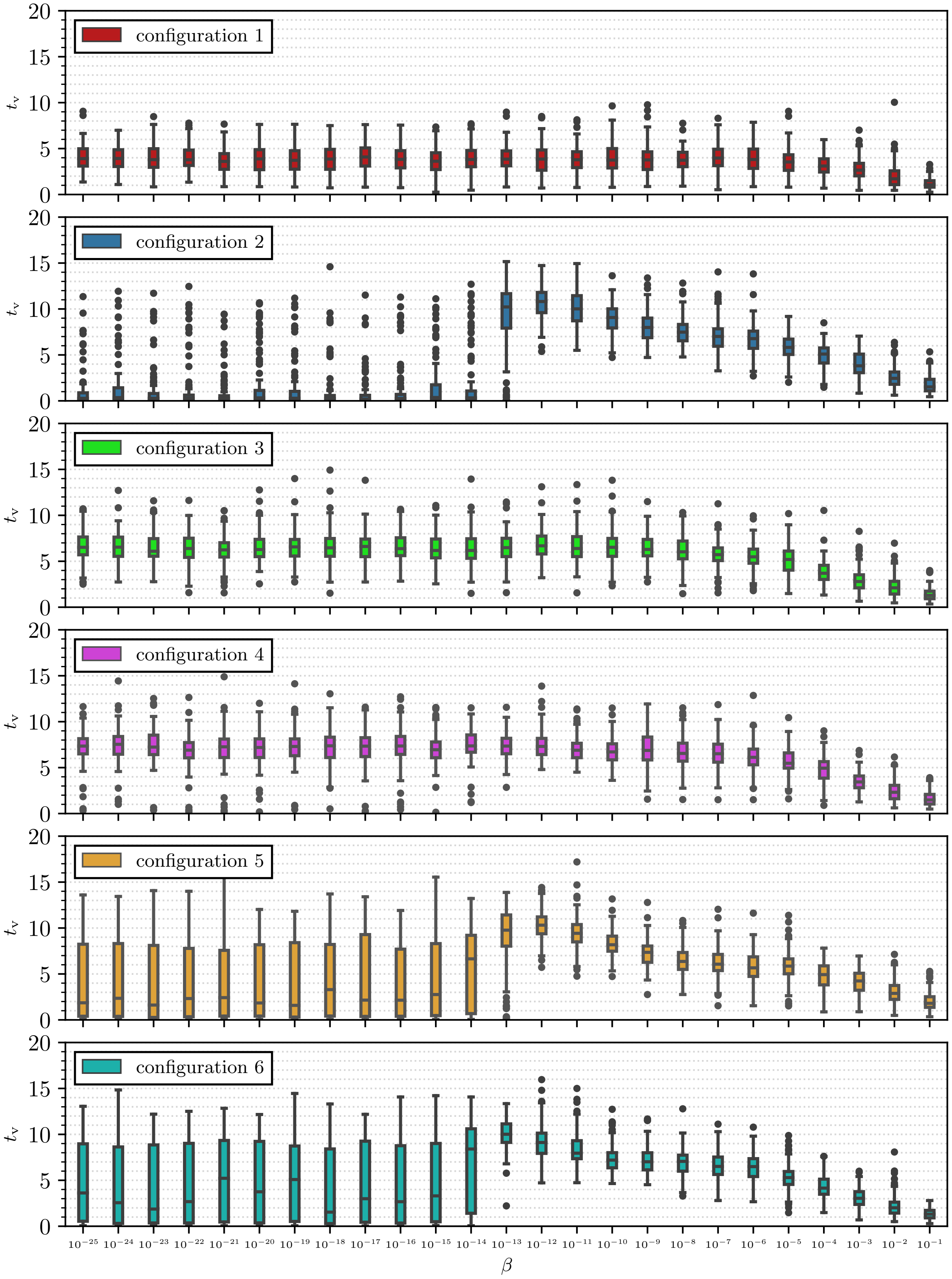


Fig. 23: Boxplots of the forecast horizon in Lyapunov times of 100 forecasted trajectories of the Lorenz-63 system against the regression parameter ($\beta \in \{10^{-1}, 10^{-3}, 10^{-10}, 10^{-20}\}$) for six different hyperparameter configurations of V, G, r, and $[a, b]$ (see Table 3).

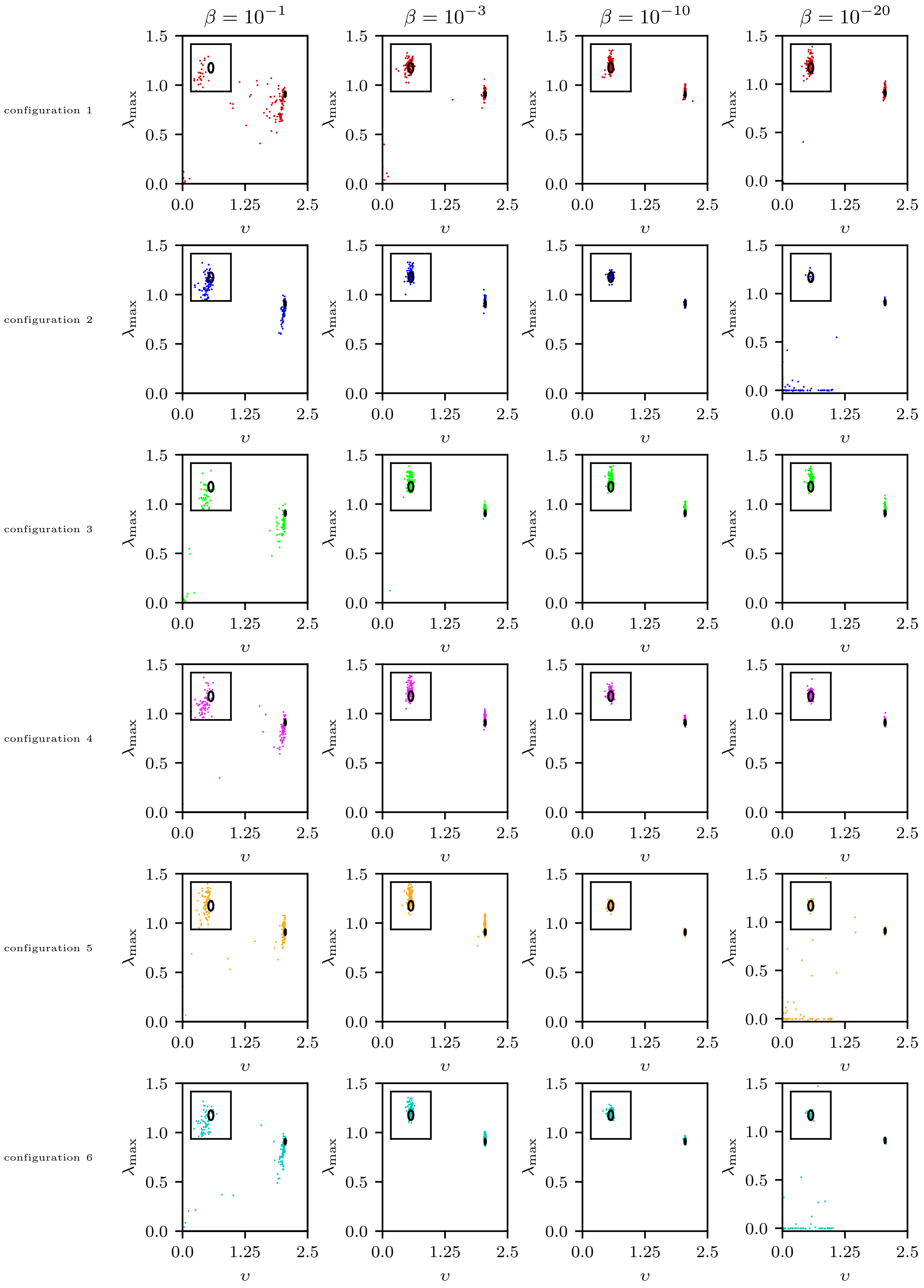


Fig. 24: Scatter plots of the predicted forecast horizons showing the predicted $\lambda_{\max}$ scattered against the predicted υ for each of the 100 realizations for $\beta \in \{10^{-1}, 10^{-3}, 10^{-10}, 10^{-20}\}$ and six different configurations of V, G, r, and $[a, b]$ (see Table 3).

5.2.3 Reservoir output dimension enlargement

Temporal multiplexing, spatial multiplexing, and the incorporation of higher powers of the reservoir response (hyperparameters V, r, and G) are methods that increase the dimension of the output vector $\mathbf{q}(k)$ of the quantum reservoirs at step k. This paragraph describes two numerical experiments focusing on the effect of increasing the dimension of the output vector. In the first experiment, V and r are varied while all other hyperparameters are fixed ($[a, b]$=[0.20,0.80], $\beta = 10^{-10}$, G=1). In the second experiment, V and G are varied and all other parameters are fixed ($[a, b]$=[0.20,0.80], $\beta = 10^{-10}$, r=1). Both experiments involve training models to predict the Lorenz-63 system for each hyperparameter combination for $N_{\text{stat}} = 100$ realizations. For all forecasted trajectories, the short- and long-term measures are calculated.

Fig. 25 and Fig. 26 display the short-term prediction quality measures from the first experiment where the number of quantum reservoirs r, and the number of evolution and measurement processes V, are varied. Fig. 25 is a boxplot illustrating the statistics of the forecast horizon for all hyperparameter combinations. Fig. 26 presents a three-dimensional barplot illustrating the relationship between r, V, and the dimension of the reservoir output vector $\mathbf{q}(k)$. The colour of the bars represents the mean forecast horizon. These figures clearly demonstrate that increasing the dimension leads to improved short-term prediction results. The long-term forecasting quality of the investigated models is analyzed in Fig. 27. Increasing the dimension of $\mathbf{q}(k)$, which means increasing the number of nodes, enhances the accuracy of the climate predictions. Comparing the scatter plots of predicted $\lambda_{\max}$ and υ for r=1 and V=5 (dim($\mathbf{q}(k)$)=51) and for r=2 and V=2 (dim($\mathbf{q}(k)$)=41) reveals differences in long-term prediction quality depending on how the dimension is increased. The Lyapunov exponent is more accurately forecasted in the second hyperparameter combination, indicating that the prediction quality also depends on the type of added node.

The short-term prediction quality for the second experiment, where the dimension of the output vector is enlarged by temporal multiplexing and up to which power the reaservoir response is used, is depicted in Fig. 28 and Fig. 29. Fig. 28 is a boxplot showing the statistics of the forecast horizon for all analyzed combinations of G and V. Fig. 29 is a three-dimensional barplot showing the relationship between G, V, and the dimension of the output vector $\mathbf{q}(k)$. The colour of the bars indicates the mean forecast horizon. These plots demonstrate that increasing the dimension of the output vector results in models that provide accurate forecasting over longer time spans (increased forecast horizon). For each set of analyzed hyperparameters, scatter plots showing the predicted climate are depicted in Fig. 30. A larger output vector dimension leads to more precise climate predictions. The type of added node influences the accuracy, with the Lyapunov exponent being more accurately forecasted for G=4 and V=1 (dim($\mathbf{q}(k)$)=41) than for G=1 and V=5 (dim($\mathbf{q}(k)$)=51).

The results of the experiments confirm that increasing the output vector dimension is an effective way to enhance the forecasting capability of small quantum systems used in the QRC model. Therefore, it is crucial to incorporate the hyperparameters r, V, and G in the hyperparameter optimization process when optimizing the model for forecasting a dynamical system based on its evolution in the past. An interesting observation is that to predict the Lyapunov exponent accurately, multiple methods that increase the dimension of the output vector need to be employed together (see Fig. 27 and Fig. 30). Furthermore, the forecast horizon does not increase infinitely by increasing the output vector dimension using only one method. This is further detailed in the appendix (Appendix B.2). Therefore, it is necessary to explore the entire hyperparameter space and to use multiple methods to enlarge the output vector dimension dim($\mathbf{q}(k)$) to identify optimal hyperparameter configurations for achieving superior forecasting accuracy.

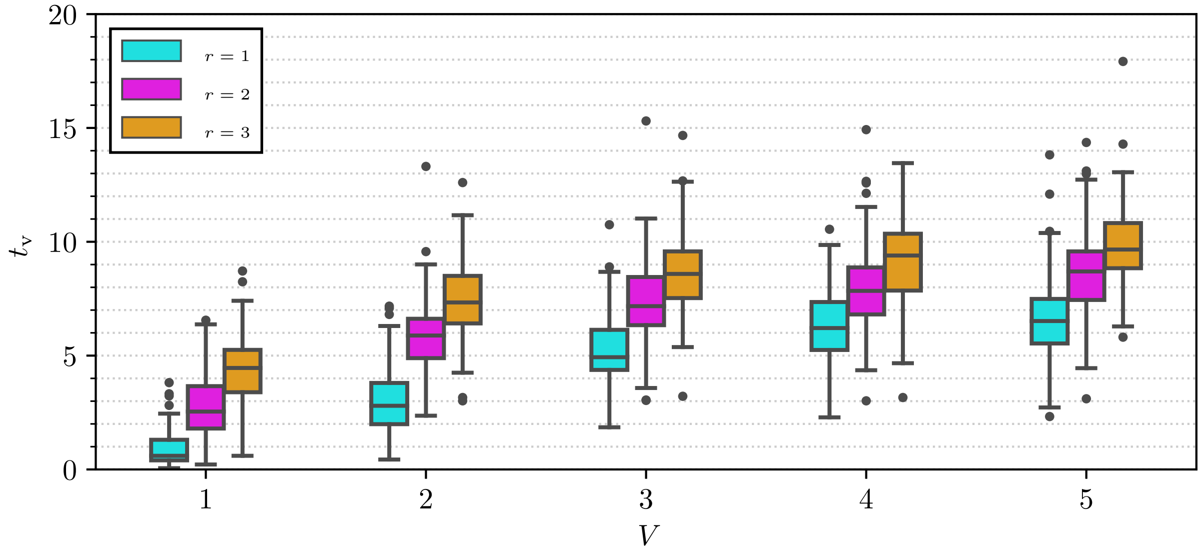


Fig. 25: Boxplot of the forecast horizon in Lyapunov times of 100 forecasted trajectories of the Lorenz-63 system for multiple combinations of the number of evolution and measurements ($V \in \{1,2,3,4,5\}$) and number of employed quantum reservoirs ($r \in \{1,2,3\}$). The remaining hyperparameters are kept constant at $\beta = 10^{-10}$, $[a,b]$=[0.20,0.80], and G=1.

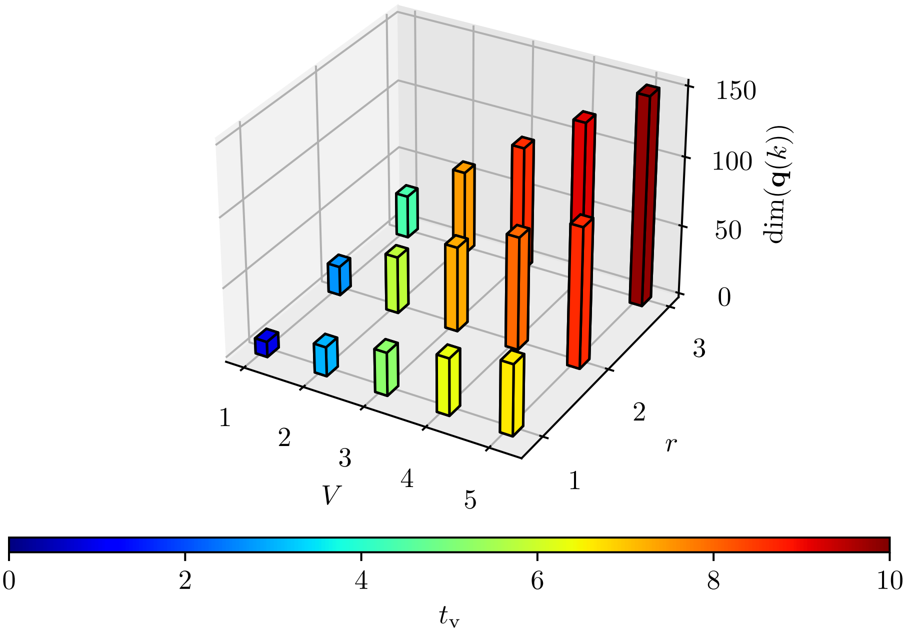


Fig. 26: Three-dimensional barplot with the number of evolution and measurement processes, V, on the x-axis, the number of reservoirs, r, on the y-axis, and the dimension of the output vector $\mathbf{q}(k)$ on the z-axis. The colour of the bars represents the mean forecast horizon ($N_{\text{stat}} = 100$) of the predicted trajectories (Lorenz-63) of the models (with $\beta = 10^{-10}$, $[a,b]$=[0.20,0.80], and G=1).

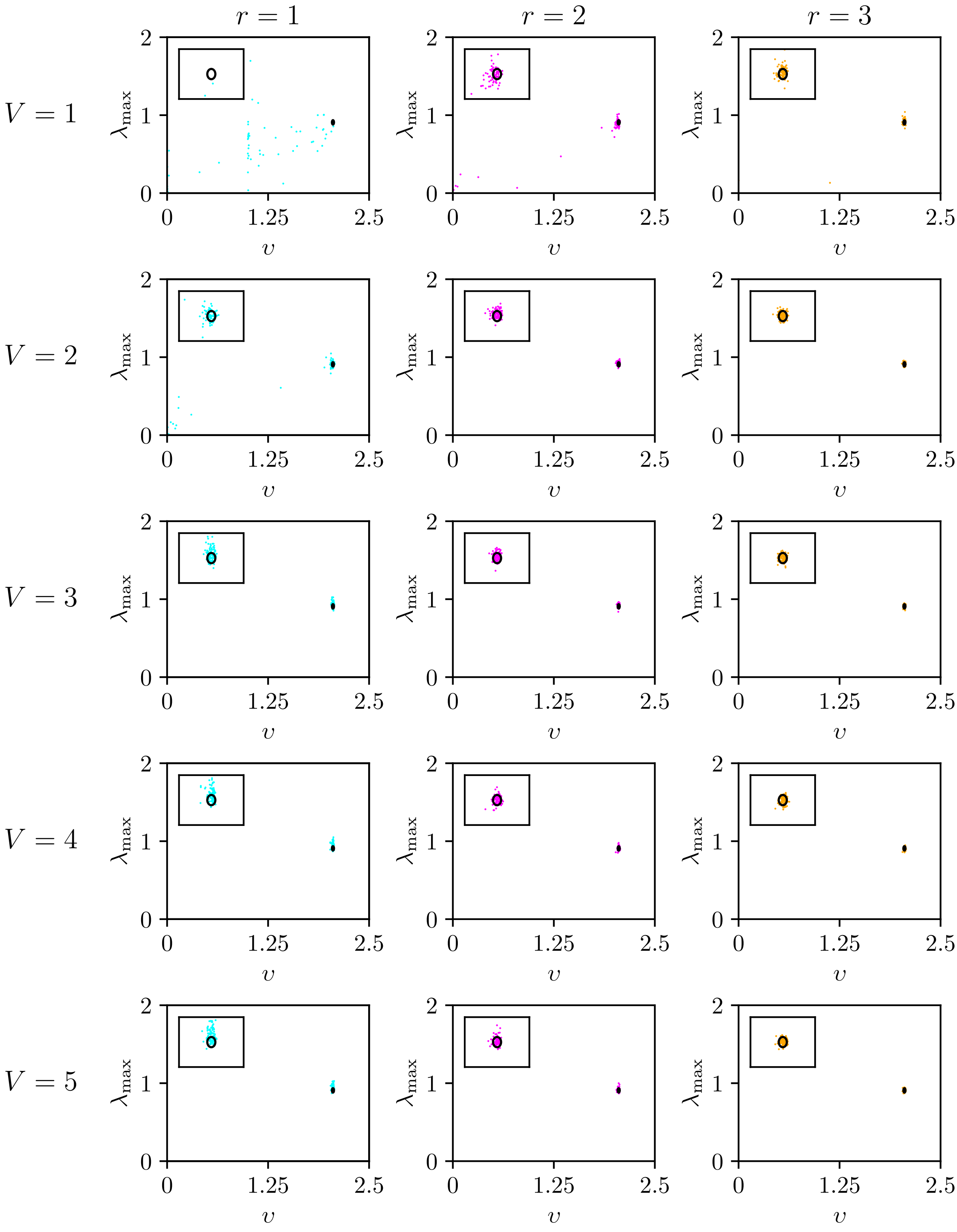


Fig. 27: Scatter plots showing the predicted $\lambda_{\max}$ and the predicted υ for each of the 100 realizations for various combinations of the number of employed quantum reservoirs ($r \in \{1, 2, 3\}$) and number of measurement and evolution processes ($V \in \{1, 2, 3, 4, 5\}$). The remaining hyperparameters are kept fixed at $\beta = 10^{-10}$, $[a, b]$=[0.20, 0.80], and G=1.

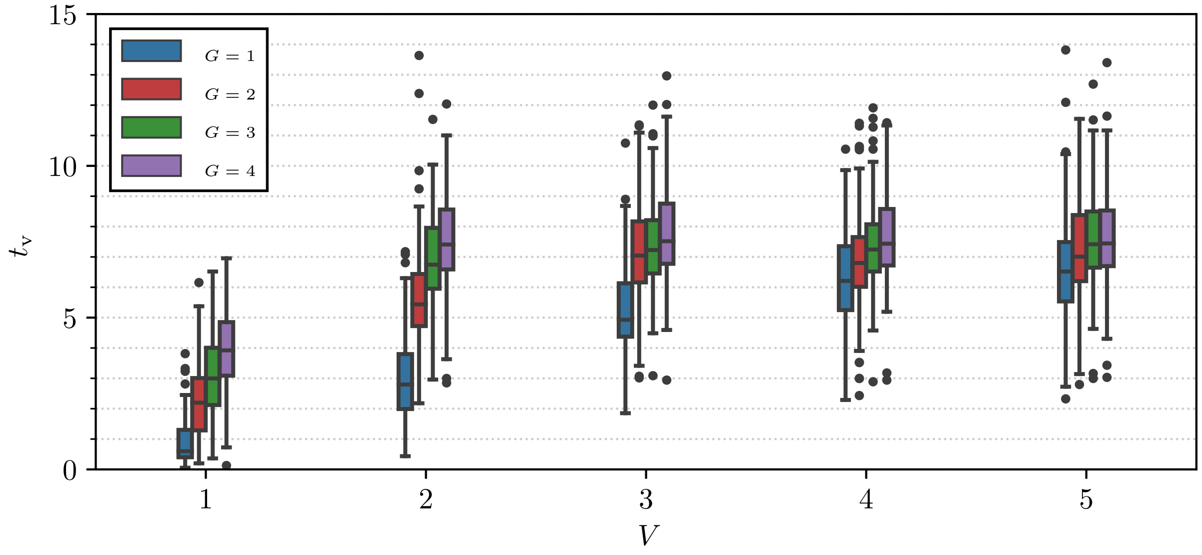


Fig. 28: Boxplot of the forecast horizon in Lyapunov times of 100 forecasted trajectories of the Lorenz-63 system for multiple combinations of the number of evolution and measurements ($V \in \{1, 2, 3, 4, 5\}$) and the employed readout function ($G \in \{1, 2, 3, 4\}$). The remaining hyperparameters are kept constant at $\beta = 10^{-10}$, $[a, b]$=[0.20,0.80], and r=1.

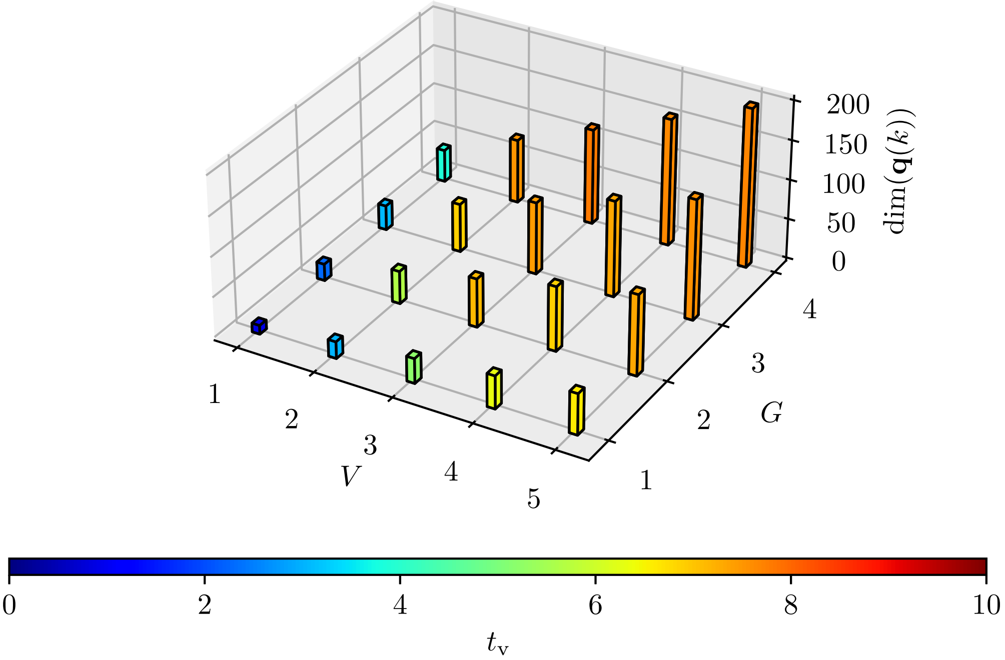


Fig. 29: Three-dimensional barplot with the number of evolution and measurement processes, V, on the x-axis, the type of readout function, G, on the y-axis, and the dimension of the output vector $\mathbf{q}(k)$ on the z-axis. The colour of the bars represents the mean forecast horizon ($N_{\mathrm{stat}} = 100$) of the continuations of the time series (Lorenz-63) obtained with the model (with $\beta = 10^{-10}$, $[a, b]$=[0.20,0.80], and r=1).

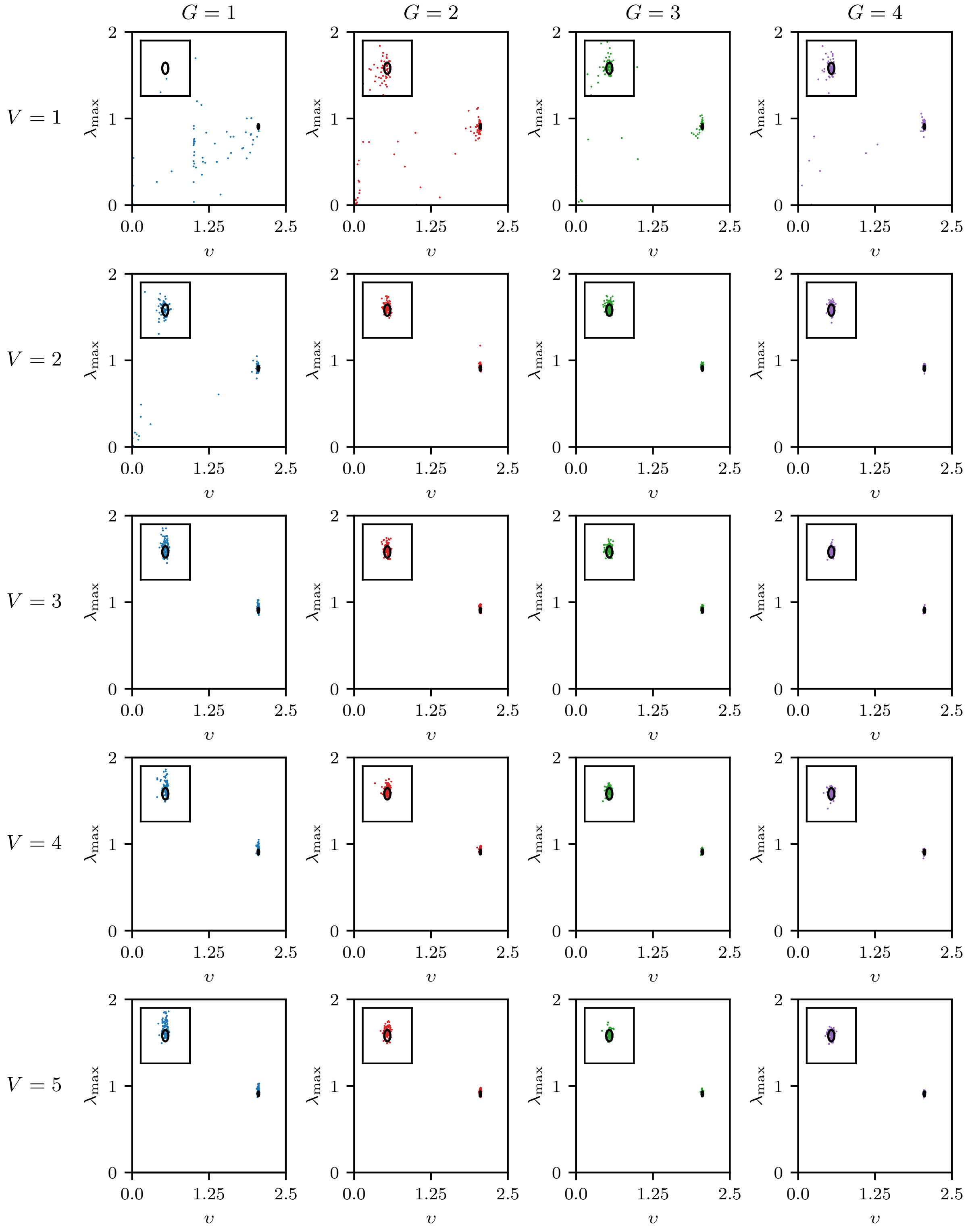


Fig. 30: Scatter plots showing the predicted $\lambda_{\max}$ and the predicted υ for each of the 100 realizations for various combinations of the employed readout function ($G \in \{1, 2, 3, 4\}$) and the number of measurement and evolution processes ($V \in \{1, 2, 3, 4, 5\}$). The remaining hyperparameters are kept fixed at $\beta = 10^{-10}$, $[a, b]$=[0.20,0.80], and r=1.

5.2.4 Predicting chaos with hyperparameter optimized minimal QRC

The results of the investigation presented thus far have demonstrated that the modifications made to the QRC algorithm are both useful and necessary to achieve accurate forecasting of dynamical systems with QRC when small quantum systems are employed. The conclusion is that choosing the quantum system so that the echo state property is fulfilled, and then optimizing r, V, G, β, and $[a, b]$, is a sensible approach to optimize the model for specific forecasting tasks. In this paragraph, this approach is used to forecast a variety of three-dimensional synthetic chaotic systems using 4-qubit quantum reservoirs. This is done to further demonstrate the predictive power of the QRC framework introduced in this thesis when utilizing small quantum systems. Furthermore, the prediction is evaluated with statistical relevance. As before, the evaluation focuses on short- and long-term accuracy. The goal is to showcase that the QRC models have significant potential (even for small quantum systems) in the field of dynamical systems prediction.

Despite having fixed parameters, the hyperparameter space of the framework is too large (considering numerical limitations) to perform a fine grid search for finding the optimal combination of hyperparameters (r, V, G, β, and $[a, b]$) for each of the eight chaotic systems. Instead, a Bayesian hyperparameter optimization using the Python package Optuna [78] over the hyperparameter space section presented in Table 4 is used to find the best-performing hyperparameter configuration for each system. For all eight investigated chaotic systems, the optimal configuration of hyperparameters is obtained by maximizing the mean forecast horizon of the forecasted trajectories received from the trained models (N_{stat}=100). The best-performing hyperparameter configuration (found through Bayesian hyperparameter optimization) for each system is listed in Table 5. With these best-performing hyperparameter configurations, 500 models for each chaotic system are trained (with different parts of the attractor, N_{stat}=500). For each of the trained models, the trajectories are continued for N_{pred}=20000 steps to evaluate the short- and long-term prediction efficiency of the models for all eight chaotic systems. In the following, the forecasted trajectories for all chaotic systems (examples can be found in Appendix C.2) are analyzed, and their short- and long-term prediction accuracies are evaluated.

Parameter	Parameter range
Number of evolutions	$V \in \{1, 2, ..., 15\}$
Number of reservoirs:	$r \in \{1, 2, 3\}$
Regression parameter:	$10^{-20} < \beta < 10^{3}$
Readout function:	$G \in \{1, 2, 3, 4\}$
Normalization interval:	$[a, b] \in \{[0.05, 0.95], [0.10, 0.90], [0.15, 0.85], [0.20, 0.80], [0.25, 0.75], [0.30, 0.70], [0.35, 0.65], [0.40, 0.60], [0.45, 0.55]\}$

Table 4: Hyperparameter space that is examined by the Bayesian hyperparameter search.

System	V	r	β	G	$[a, b]$
Lorenz-63	9	3	$1.41{\cdot}10^{-12}$	3	[0.15, 0.85]
Chen	8	3	$1.09{\cdot}10^{-12}$	3	[0.30, 0.70]
Chuas-Circuit	14	3	0.000269	2	[0.10, 0.90]
Halvorsen	8	3	$1.41{\cdot}10^{-12}$	3	[0.20, 0.80]
Rössler	9	3	$2.10{\cdot}10^{-12}$	3	[0.20, 0.80]
Rucklidge	7	3	$1.25{\cdot}10^{-12}$	4	[0.15, 0.85]
Thomas	15	3	$1.89\ {\cdot}10^{-10}$	4	[0.05, 0.95]
WINDMI	10	3	$9.13{\cdot}10^{-12}$	4	[0.05, 0.95]

Table 5: Best-performing (maximized forecast horizon) hyperparameter configuration for each chaotic system.

In many applications, it is important that the machine learning model is able to very accurately forecast a time series of a dynamical system for as long as possible. The best-performing hyperparameter configurations are obtained by maximizing this ability. The mean forecast horizon of the 500 trained models can be found for each of the chaotic systems in Table 6, and the distributions are further investigated in a boxplot shown in Fig. 31. A comparison of these results with those of [69] shows that the mean forecast horizon is, in all cases, at least comparable to that of the "classical" RC approach. In some cases, the QRC models even outperform some "classical" RC and Hybrid-RC approaches (models with prior knowledge about the physics of the underlying equations of the chaotic systems). Fig. 31 shows that the whiskers of

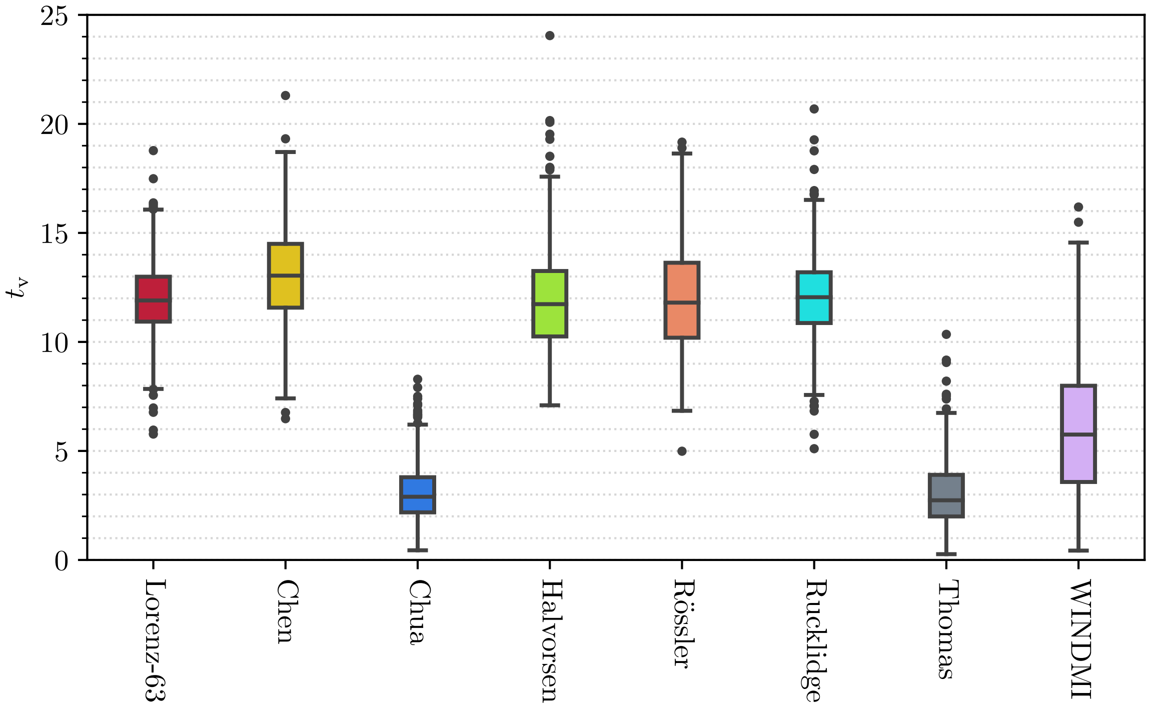


Fig. 31: Boxplot of the forecast horizon in Lyapunov times for all eight forecasted chaotic systems. The initilized hyperparmeters for each of the chaotic systems are denoted in Table 5. For each chaotic system, 500 time series are forecasted. The mean and standard deviations of the distributions can be found in Table 6.

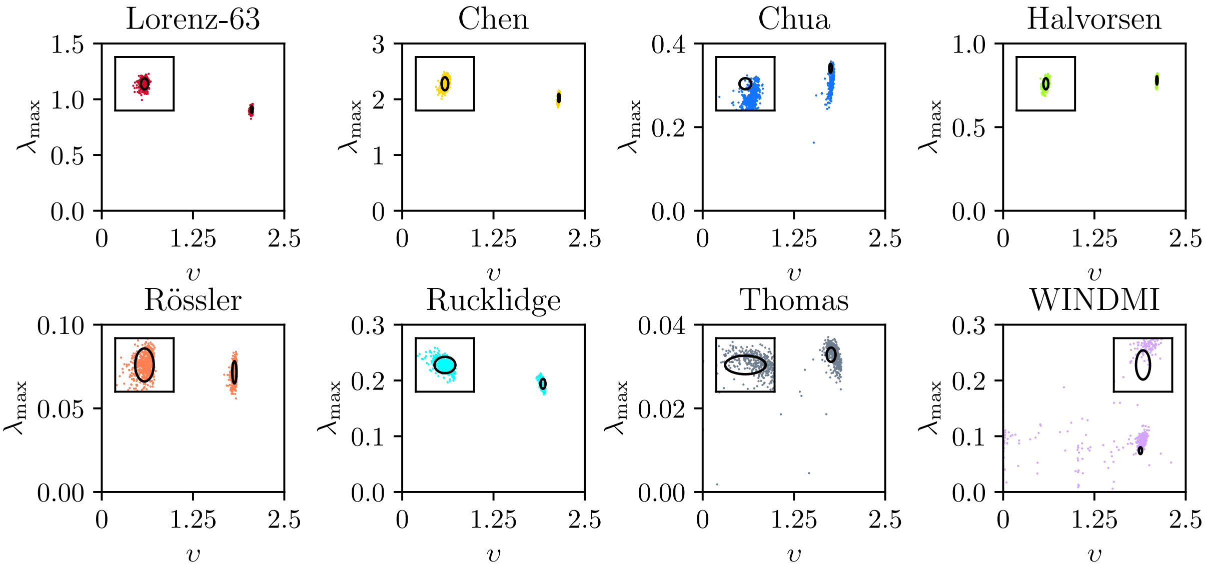


Fig. 32: Largest Lyapunov exponent scattered against correlation dimension for all eight chaotic systems for each of the 500 forecasted trajectories for all eight systems. The used hyperparameters for each of the chaotic systems are presented in Table 5.

system	t_v	predicted λ_max	true λ_max	predicted υ	true υ
Lorenz-63	11.9 ± 1.7	0.91 ± 0.02	0.91 ± 0.02	2.053 ± 0.008	2.052 ± 0.009
Chen	13.0 ± 2.2	2.02 ± 0.05	2.02± 0.05	2.146 ± 0.008	2.145± 0.008
Chua	3.1 ± 1.4	0.31 ± 0.02	0.341± 0.007	1.77 ± 0.02	1.75± 0.01
Halvorsen	11.9± 2.3	0.78 ± 0.02	0.78 ± 0.02	2.106 ± 0.006	2.106 ± 0.006
Rössler	11.9 ± 2.4	0.071 ± 0.005	0.072 ± 0.004	1.82 ± 0.02	1.82 ± 0.02
Rucklidge	12.1± 2.0	0.194 ± 0.006	0.194±0.006	1.93 ± 0.02	1.93± 0.02
Thomas	3.0 ± 1.5	0.032 ± 0.004	0.033 ± 0.001	1.8 ± 0.2	1.76 ± 0.04
WINDMI	5.9 ± 2.9	0.08± 0.04	0.074± 0.004	1.5 ± 0.8	1.88± 0.02

Table 6: Mean values (500 realizations) with standard deviations for all three forecast evaluation measures are provided for the chosen hyperparameter configuration across all eight chaotic systems. For comparison, the results from calculating the climate measures on true trajectories are also included (from Table 2).

the boxplots span a large range, meaning that the models perform very differently for different realizations of the Hamiltonians and different parts of the attractor being forecasted. This is also visible in the standard deviation shown in Table 6. Futher investigation of Table 6 and Fig. 31 shows that three of the eight chaotic systems are predicted accurately on a much shorter time scale (fewer Lyapunov times) than the others. Interestingly, these systems are also relatively poorly forecasted using conventional RC methods. It is an important and obvious question to determine the cause of the systematic differences in performance among the forecasted systems. However, this research topic is beyond the scope of this work.

In some scenarios, the focus might not be on a very accurate short-term prediction of the dynamical system's state but rather on whether a dynamical system's long-term climate can be correctly reproduced. For each chaotic system, the forecasted trajectories, each consisting of 20000 steps, are used to calculate the mean largest Lyapunov exponent and the correlation dimension. The results are presented in Table 6 together with the values obtained by the same calculation with true trajectories (see Sec. 4.2.2 and Sec. 4.2.3). In Fig. 32, the distributions of the predicted climates are shown. The spatial and temporal statistical properties of the five chaotic systems that are forecasted accurately for long time scales are predicted extremely well. In the sample, there are no single outliers with large deviations for the Lyapuov exponent or the correlation dimension. All realizations are within a 5 σ error range of the two measured quantities. For the remaining three systems, the climate of the systems is reproduced well in some cases but some forecasted trajectories exhibit long-term behaviour that is far from the statistical fluctuations of the true data. These findings suggest that the (minimal) QRC setup does not only learn patterns of the time series by heart, leading to good short-term predictions but rather correctly learns the dynamics of the underlying chaotic systems, enabling correct climate reproduction. It is crucial to emphasize that this achievement was attained despite the use of relatively short training time series.

5.3 Conclusions

The experiments detailed in this chapter yield several significant insights into the QRC model. Firstly, optimizing the normalization interval $[a, b]$ is crucial as it strongly influences the model's ability to forecast dynamical systems. Secondly, short-term prediction accuracy and the climate reproduction is further enhanced by incorporating ridge regression and selecting an appropriate regression parameter, β. Thirdly, due to the small size of the quantum systems, increasing the dimension of the output vector through various methods becomes necessary. This enlargement of the output vector dimension introduces hyperparameters (r, V, and G) that also require optimization. Collectively, these findings indicate that the modifications and optimizations investigated substantially impact forecasting outcomes, improving the stability and quality of predictions achievable with very small quantum systems. In some cases, they are essential for developing functional models.

The fourth and final conclusion is that using an intelligent hyperparameter optimization technique (e.g., Bayesian optimization) for each specific forecasting task allows for the development of models that yield strong prediction results. This demonstrates the approach's potential for forecasting dynamical systems. The trained QRC models successfully reproduce climatic patterns and accurately predict the short-term evolution of chaotic systems using relatively short training time series. For the WINDMI, Chua, and Thomas systems, instances where the trained model fails or performs significantly worse than the bulk of the other models have been observed. Future research should investigate these failing predictions to identify necessary modifications to the model, its hyperparameters, and preprocessing steps to enhance prediction accuracy. As noted at the beginning of this chapter, due to numerical limitations, this thesis does not include a full optimization of all hyperparameters. The following numerical experiment demonstrates the potential for further improvements when the quantum systems (the unitary operator) are optimized for forecasting tasks. This experiment focuses on forecasting the WINDMI system. The model is trained with the same hyperparameters as previously used (see Table 5) and is repeatedly trained with different Hamiltonian configurations until a model initialization is achieved that extends the forecast horizon beyond six Lyapunov times. Subsequently, 500 trajectories of the WINDMI system, each trained with different parts of the attractors, are forecasted using the Hamiltonians that achieved $t_\mathrm{v} > 6$ Lyapunov times. In Fig. 33 and Fig. 34, the short- and long-term prediction results are compared to the original results, which utilized randomly drawn Hamiltonians. The results clearly indicate that selecting appropriate unitary evolution enhances prediction performance and eliminates completely failing predictions. This shows the importance of optimizing the evolution of the quantum systems. Future research in this direction is discussed in the outlook (Chapter 6).

In summary, the numerical experiments indicate that it is feasible to forecast time series from past evolution using QRC with the theoretically smallest possible quantum systems, provided the right hyperparameters are chosen. The promising nature of this conclusion suggests that near-term practical applications using upcoming quantum hardware are becoming more likely.

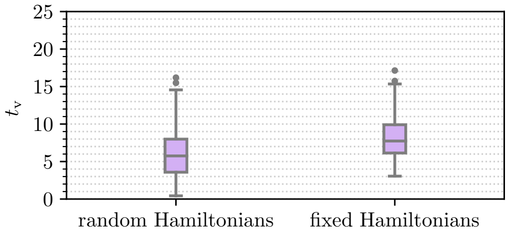


Fig. 33: Forecast horizon for randomly drawn Hamiltonians and fixed (well-performing) Hamiltonians of the WINDMI system forecasting (500 trajectories each) with the hyperparameter configuration denoted in Table 5.

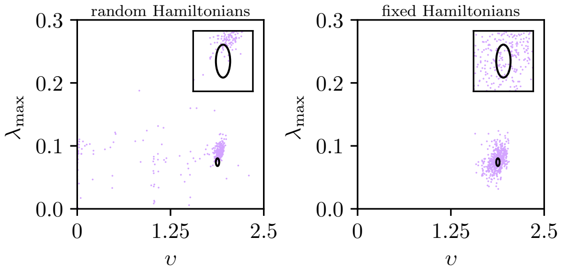


Fig. 34: Predicted climate for each of the 500 forecasted trajectories of the WINDMI system as presented in Fig. 32 (random Hamiltonians) and the same plot with fixed (well-performing) Hamiltonians for the same hyperparameter configuration (see Table 5).

Chapter 6

Summary and Outlook

To summarize, this thesis explored a modified version of the Quantum Reservoir Computing (QRC) algorithm and focused on optimizing the algorithm for small quantum systems. This modified framework was introduced in Chapter 2. In Chapter 3, the employed quantum systems were analyzed, with an emphasis on reproducing the results of previous studies [30] that demonstrate the relationship between the phase of the quantum systems and the algorithm's ability to produce quantum states that exhibit the echo state property. The echo state property is a necessary feature for models that are able to forecast time series. Fundamental concepts of chaotic systems and the performance measures used throughout the evaluations of the framework were defined in Chapter 4. In Chapter 5, the results of Chapter 3 were utilized to analyze the introduced QRC algorithm for minimal quantum systems, beginning with a statistical evaluation of the introduced algorithmic changes. The Lorenz-63 system was used as a benchmark for forecasting. These evaluations showed that the modifications introduced in this thesis significantly improved forecasting quality. This chapter concluded with the presentation of models obtained through Bayesian hyperparameter optimization, which were trained to forecast eight different chaotic systems. A similar statistical evaluation demonstrated that these models could successfully reproduce the climate of the synthetic chaotic systems in most cases and accurately forecast their short-term evolution. It is fascinating that, even though such small quantum systems are employed and only short training time series are used in this modified QRC framework, such accurate predictions can still be achieved. Particularly noteworthy is the accurate reproduction of the climate, which represents a significant accomplishment.

The results of the investigations into minimal quantum reservoir computing are encouraging. In numerical simulations, the discussed QRC architecture is capable of forecasting chaotic systems with very high accuracy using only the smallest quantum systems that are theoretically allowed by the algorithm. However, many open questions and avenues for exploration remain. The preprocessing of the time series (training data) can be further optimized. The current investigation into the normalization interval $[a, b]$ was simplistic, and it remains to be seen whether performance can be enhanced by optimizing the hyperparameters over the continuous space 0 $\leq a < b \leq 1$. From classical reservoir computing (RC), it is known that adding artificial noise to the training data can mitigate overfitting and improve forecasting results. Future research should explore the inclusion of artificial noise in the optimization of QRC models. Furthermore, it is unclear whether ridge regression is the optimal method for training the readout layer (readout matrix $\mathbf{W}_{\text{out}}$). Other regression methods, such as Tree regression [79], might yield better performance and should be investigated. Higher powers (G) of the reservoir response and other readout functions should also be considered to possibly improve the prediction results even further. A critical question for future studies is determining what defines an optimal quantum reservoir. As discussed in Sec. 5.3, different unitary operators achieve varying levels of forecasting quality. This means that it should be investigated which physical properties the unitary operator must have to be optimal for the QRC framework or a specific forecasting task. A promising starting point would be to examine the relationship between the echo state property and prediction accuracy in more detail. As shown in Sec. 3.2, certain realizations of the Hamiltonians, even if parameters are drawn from appropriate intervals, do not result in converging quantum states. This could explain the strong fluctuations in prediction quality and occasional prediction failures. When constructing optimal unitary operators, it is crucial to consider the limitations of near-term quantum hardware (e.g., coherence times, connectivity, etc.) to advance towards practical applications. Especially interesting is the effect of noise, as a small amount of noise could even help the training by preventing overfitting. Another avenue for exploration is modifying the algorithms to lift the restriction $N \geq d + 1$. This would be interesting for tackling more complex problems with QRC using small quantum systems. Such changes would require adjustments to the encoding of the time series.

In conclusion, the results of this thesis are highly promising, demonstrating the potential power of the QRC framework in predicting dynamical systems. The use of very small quantum systems in this study further highlights the promising nature of this approach, even within the NISQ era of quantum computing. This suggests that QRC has the potential to significantly impact various disciplines and drive advancements in these fields in the near future.

Appendix A Chaotic systems

As mentioned in the main text of this thesis, multiple synthetic chaotic systems are employed to demonstrate the capability of QRC to forecast dynamical systems based on past data. In this section of the Appendix, all systems used for this demonstration are defined. A dynamical system that continuously evolves in time must be at least three-dimensional to exhibit chaos, whereas discretely evolving systems can display chaotic behaviour even in one dimension [61]. In Appendix A.1-A.8, eight three-dimensional chaotic systems that evolve continuously in time are defined. These definitions include the system's defining differential equations, the chosen system parameters, and a figure showing the attractor in three-dimensional space as well as the individual dimensions against Lyapunov time.

A.1 Lorenz-63 attractor

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{A.1}$$

system parameter	Δt	initial state
$\rho = 28$, $\sigma = 10$, $\beta = 8/3$	0.02	[0, 0.01, 9]

Table 7: Lorenz-63 system parameter choice.

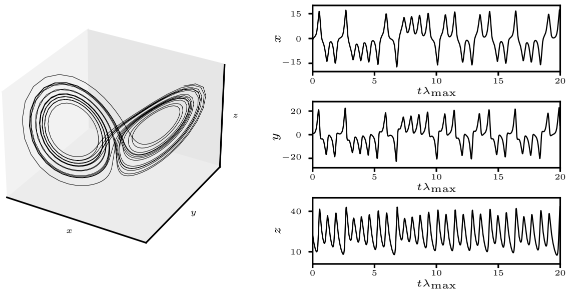


Fig. 35: Left: 3D trajectory of the Lorenz attractor. Right: Individual dimensions of the Lorenz attractor plotted against Lyapunov time.

A.2 Chen's system

$$\begin{aligned} \dot{x} &= a(y - x) \\ \dot{y} &= (c - a)x - xz + cy \\ \dot{z} &= xy - bz \end{aligned} \tag{A.2}$$

system parameter	Δt	initial state
$a = 35$, $b = 3$, $c = 28$	0.02	[10, 0, 37]

Table 8: Chen's system parameter choice.

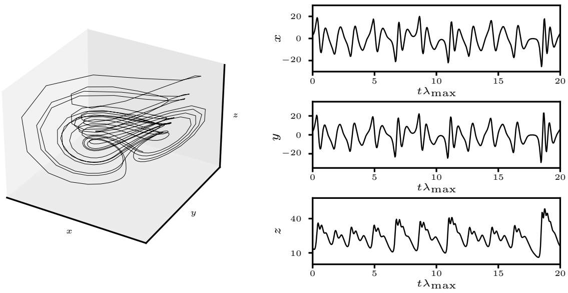


Fig. 36: Left: 3D trajectory of the Chen attractor. Right: Individual dimensions of the Chen attractor plotted against Lyapunov time.

A.3 Chua's circuit

$$\begin{aligned}\dot{x} &= \alpha[y - x + bx + 0.5(a-b)(|x+1| - |x-1|)] \\ \dot{y} &= x - y + z \\ \dot{z} &= -\beta y\end{aligned} \tag{A.3}$$

system parameter	Δt	initial state
$\alpha = 9$, $\beta = 100/7$, $a = 8/7$, $b = 5/7$	0.1	[0, 0, 0.6]

Table 9: Chua's system parameter choice.

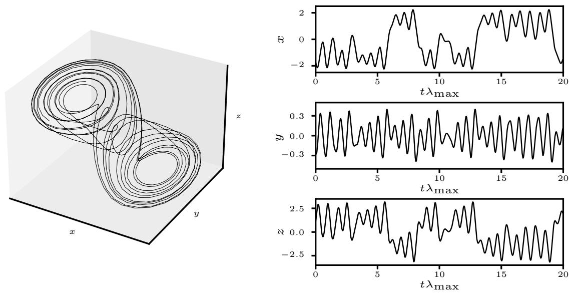


Fig. 37: Left: 3D trajectory of the Chua attractor. Right: Individual dimensions of the Chua attractor plotted against Lyapunov time.

A.4 Halvorsen's cyclically symmetric attractor

$$\begin{aligned}\dot{x} &= -ax - 4y - 4z - y^2 \\ \dot{y} &= -ay - 4z - 4x - z^2 \\ \dot{z} &= -az - 4x - 4y - x^2\end{aligned} \tag{A.4}$$

system parameter	Δt	initial state
$a = 1.27$	0.05	[5, 0, 0]

Table 10: Halvorsen's system parameter choice.

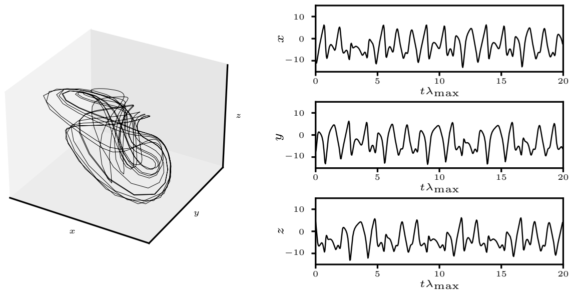


Fig. 38: Left: 3D trajectory of the Halvorsen attractor. Right: Individual dimensions of the Halvorsen attractor plotted against Lyapunov time.

A.5 Rössler attractor

$$\begin{aligned}\dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c)\end{aligned} \tag{A.5}$$

system parameter	Δt	initial state
$a = b = 0.2$, $c = 5.7$	0.1	[9, 0, 0]

Table 11: Rössler system parameter choice.

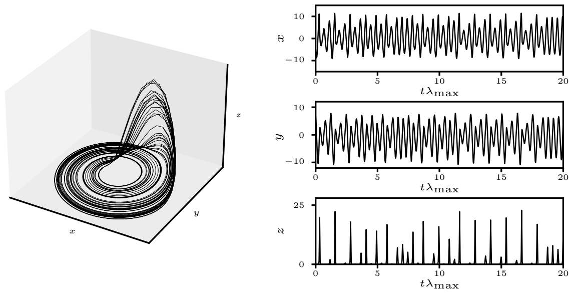


Fig. 39: Left: 3D trajectory of the Rössler attractor. Right: Individual dimensions of the Rössler attractor plotted against Lyapunov time.

A.6 Rucklidge attractor

$$\begin{aligned} \dot{x} &= -\kappa x + \lambda y - yz \\ \dot{y} &= x \\ \dot{z} &= -z + y^2 \end{aligned} \tag{A.6}$$

system parameter	Δt	initial state
$\kappa = 2.0$, $\lambda = 6.7$	0.1	[1, 0, 4.5]

Table 12: Rucklidge system parameter choice.

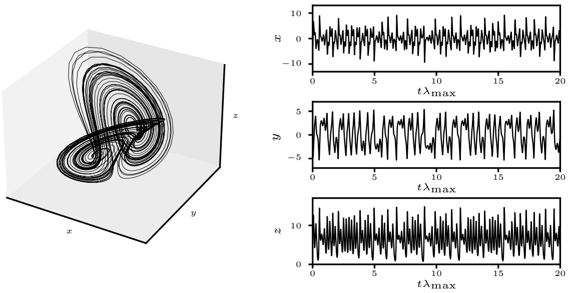


Fig. 40: Left: 3D trajectory of the Rucklidge attractor. Right: Individual dimensions of the Rucklidge attractor plotted against Lyapunov time.

A.7 Thomas' cyclically symmetric attractor

$$\begin{aligned} \dot{x} &= -bx + \sin(y) \\ \dot{y} &= -by + \sin(z) \\ \dot{z} &= -bz + \sin(x) \end{aligned} \tag{A.7}$$

system parameter	Δt	initial state
$b = 0.18$	0.3	[0.1, 0, 0]

Table 13: Thomas' system parameter choice.

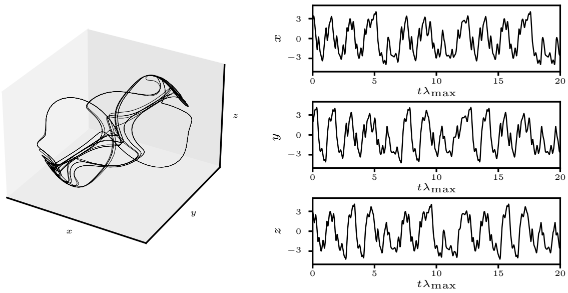


Fig. 41: Left: 3D trajectory of the Thomas attractor. Right: Individual dimensions of the Thomas attractor plotted against Lyapunov time.

A.8 WINDMI attractor

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= z \\ \dot{z} &= -az - y + b - \exp(x) \end{aligned} \tag{A.8}$$

system parameter	Δt	initial state
$a = 0.7$, $b = 2.5$	0.2	[0, 0.8, 0]

Table 14: WINDMI system parameter choice.

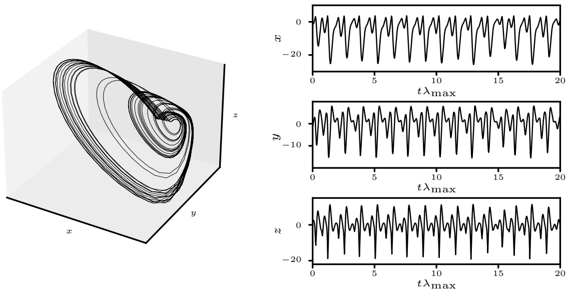


Fig. 42: Left: 3D trajectory of the WINDMI attractor. Right: Individual dimensions of the WINDMI attractor plotted against Lyapunov time.

Appendix B QRC properties

B.1 Echo state property of 4-qubit systems with 3D input

In Sec. 3.2, the echo state property of the employed quantum reservoirs is examined. The experiment is set up as follows: the trace distance between two different initial states is measured after evolving them according to Eq. 2.25 with the unitary operator defined in Eq. 3.2. This evolution is driven by a sequence of uniformly random numbers drawn from the interval [0,1], effectively creating a one-dimensional time series. This process is repeated for multiple times for any investigated combination of τ, h, and W. In Chapter 5, 4-qubit systems with unitary operators defined by h=2, W=0.05, and τ=20 are used to forecast three-dimensional chaotic systems. To evaluate convergence for these specific quantum system configurations (h=2, W=0.05, τ=20), sequences of three-dimensional vectors (with each coefficient uniformly drawn from the interval [0,1]) are used instead of one-dimensional sequences. The outcomes of these studies are presented in Fig. C.2 as a boxplot based on 500 different evaluations. Comparing Fig. 43 and Fig. 15 indicates that using a three-dimensional input, as anticipated, leads to much faster convergence compared to a one-dimensional input. It is observed that, even with three-dimensional inputs, there are outliers where convergence either does not occur or happens significantly slower than for the majority of realizations. The impact of the echo state property, particularly the speed of convergence (”the speed that the reservoir forgets”), warrants further investigation.

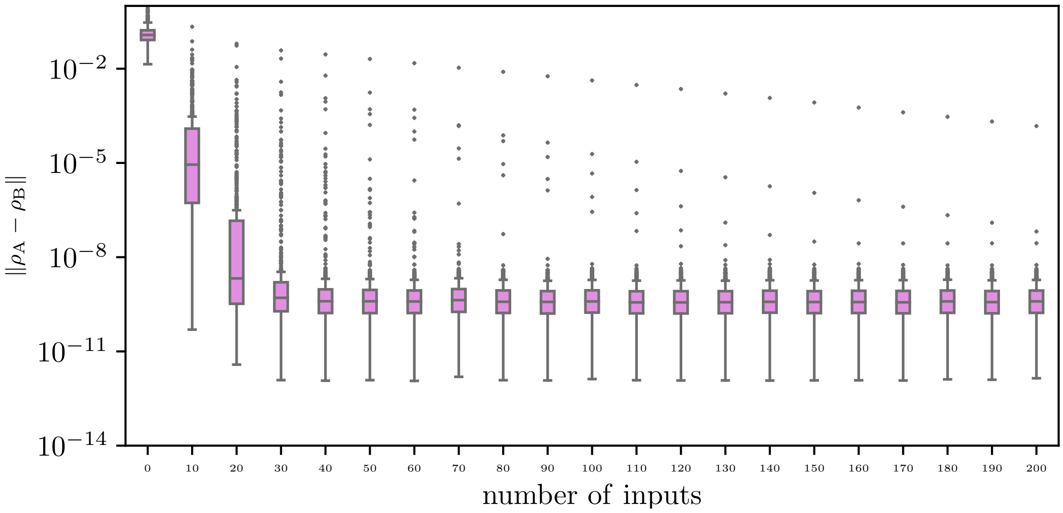


Fig. 43: Boxplots (500 realizations) of the convergence of two random states for τ = 20 dependent on the length of the three-dimensional input sequence for quantum systems of 4 qubits and their interaction described by the transverse-field Ising Hamiltonian (Eq. 3.1) plus onsite disorder with h=2 and W=0.05 (ergodic phase).

B.2 Saturation of short-term prediction quality

In Fig. 44, the short-term prediction results are presented in Fig. 25 and Fig. 26 extended for $V \in \{1, 2, 3, ..., 10\}$. Increasing the number of measurements and evolution processes does not always extend the forecast horizon. The forecast horizon saturates for this configuration at approximately V=5. This illustrates the importance of using multiple methods to increase the output vector dimension as a combination.

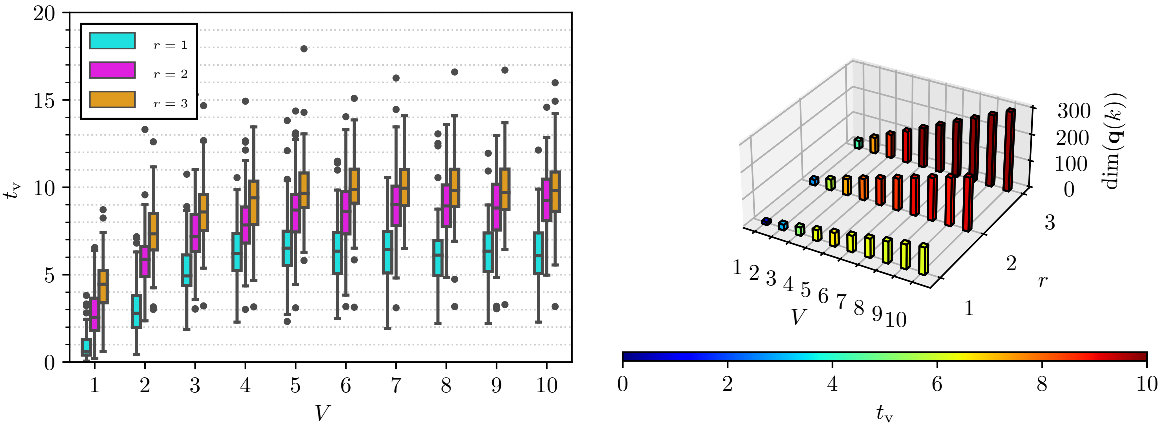


Fig. 44: Fig. 25 and Fig. 26 extended for V∈ $\{1, 2, 3, ..., 10\}$.

Appendix C Example trajectories

C.1 Predicting chaos with one 10 qubit system

Examples of predicted trajectories obtained with a QRC model employing 10 qubits are presented in Fig. 45. The model (r=1, $\beta = 10^{-12}$, G=4, V=4, and $[a, b]$=[0.10,0.90]) is trained with N_{sync}=100 and N_{train}=1999 steps, and the time series is continued for N_{pred}=2000 steps. The selected hyperparameters are not optimized. The resulting trajectory is displayed on the left in Fig. 45. The used hyperparameters are not optimized. Afterwards, another part of the attractor is selected, and N_{sync}=400 subsequent steps are used to synchronize the trained model with this part of the attractor. Then the time series is continued for N_{pred}=2000 steps (same $\mathbf{W}_{\text{out}}$ as before). The results of this are presented on the right in Fig. 45.

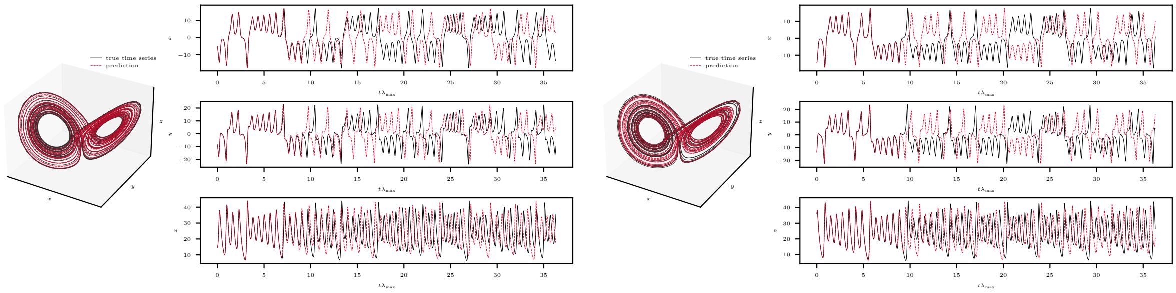


Fig. 45: Predicted trajectories obtained with a QRC model employing 10 qubits. Left: The forecasted trajectory resulting from training with N_{sync}=100 and N_{train}=1999, and continuing for N_{pred}=2000 steps. Right: The forecasted trajectory from another part of the attractor, 400 steps are used to synchronize the trained model with this part of the attractor and the same readout matrix is used to forecast 2000 steps of the attractor.

C.2 Examples of forecasted trajectories with 4 qubits

In Figs. 46-53, examples of forecasted trajectories and the real continuation of the trajectories presented each for 40 Lyapunov times after the training has ended. On the left side of the figures, the predicted attractor and the real attractor (3D trajectory) are shown, while on the right side, each of the individual dimensions is plotted against the Lyapunov time.

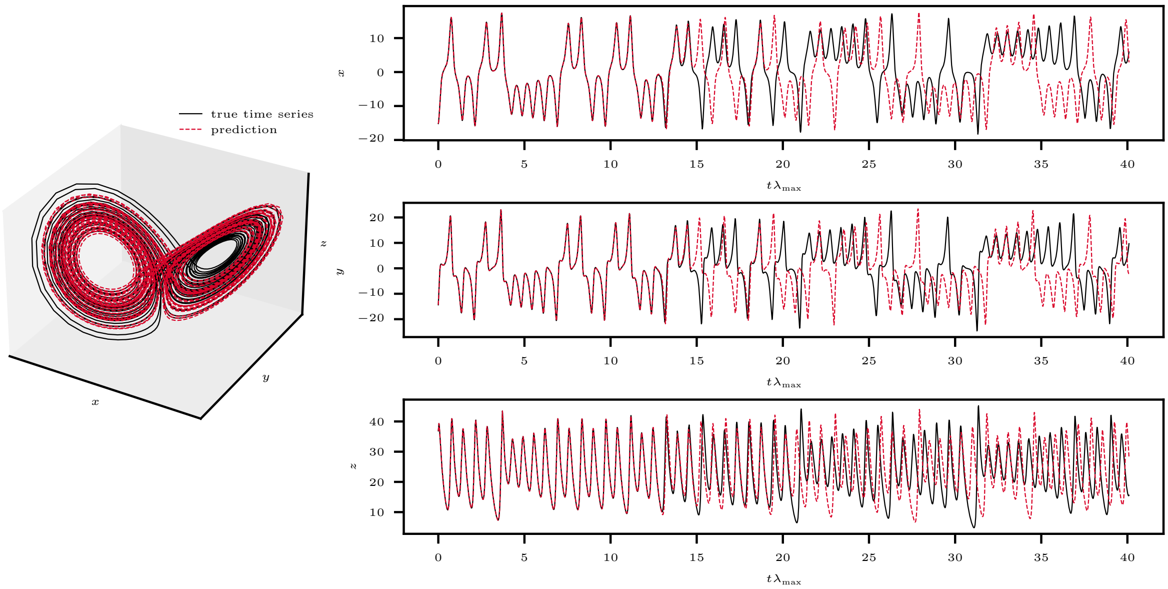


Fig. 46: Example of predicted and real continuation of the Lorenz-63 system. The predicted trajectory is forecasted with the best-performing hyperparameter configuration (see Table 5) obtained by Bayesian hyperparameter optimization.

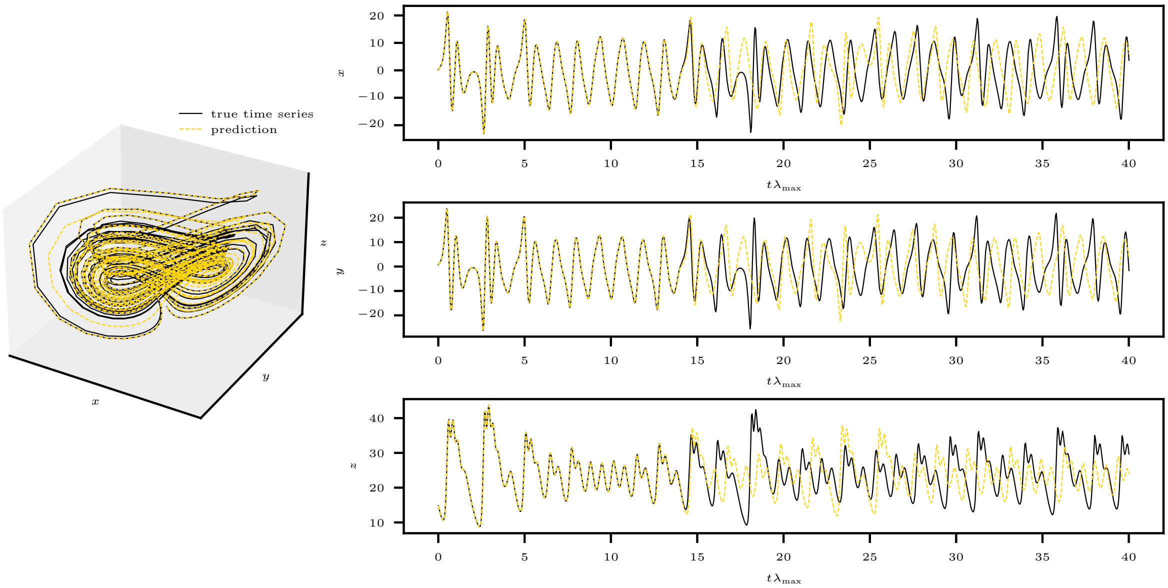


Fig. 47: Example of predicted and real continuation of the Chen system. The predicted trajectory is forecasted with the best-performing hyperparameter configuration (see Table 5) obtained by Bayesian hyperparameter optimization.

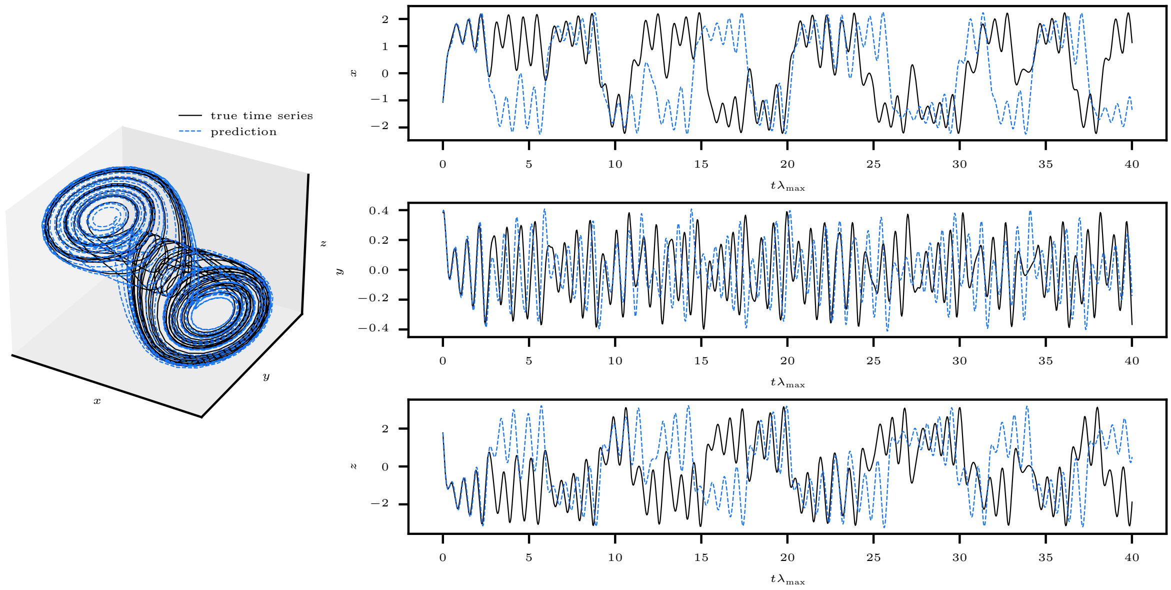


Fig. 48: Example of predicted and real continuation of the Chua system. The predicted trajectory is forecasted with the best-performing hyperparameter configuration (see Table 5) obtained by Bayesian hyperparameter optimization.

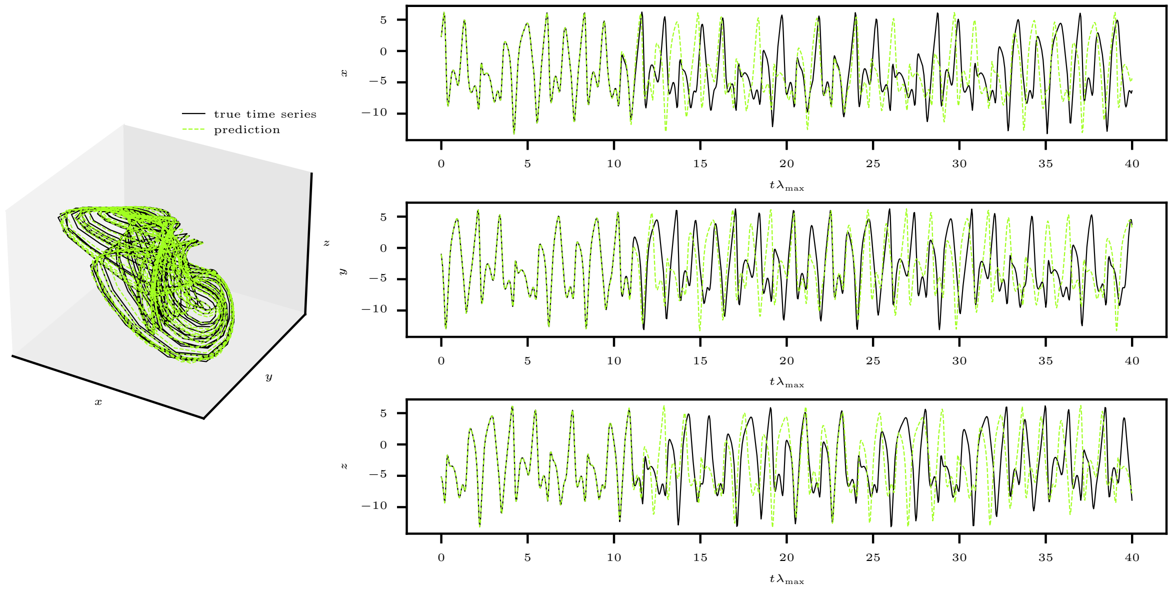


Fig. 49: Example of predicted and real continuation of the Halvorsen system. The predicted trajectory is forecasted with the best-performing hyperparameter configuration (see Table 5) obtained by Bayesian hyperparameter optimization.

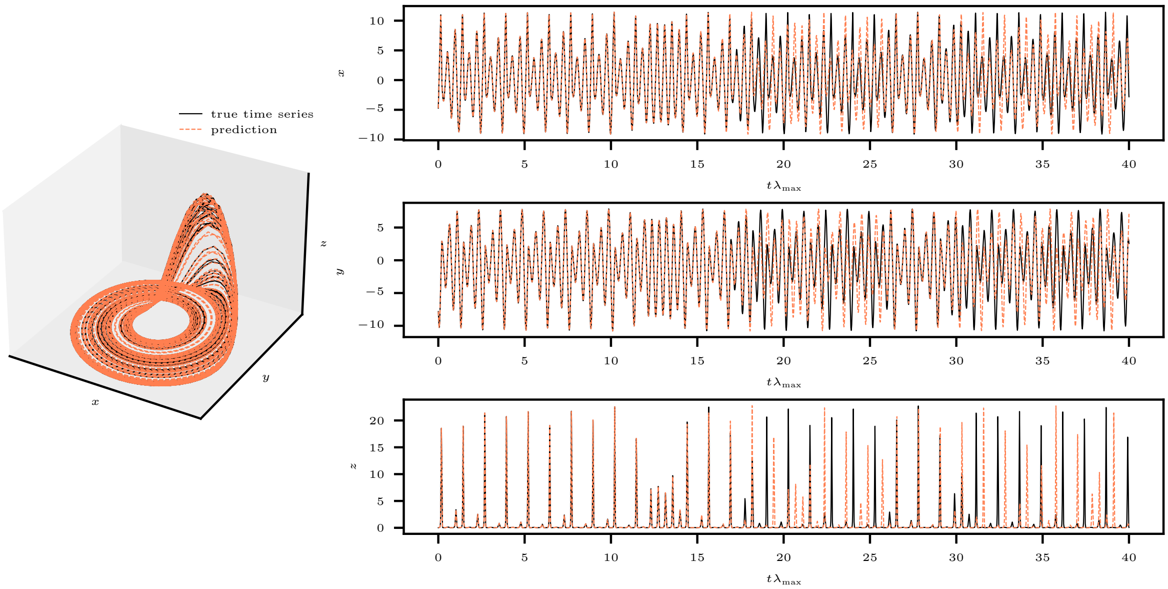


Fig. 50: Example of predicted and real continuation of the Rössler system. The predicted trajectory is forecasted with the best-performing hyperparameter configuration (see Table 5) obtained by Bayesian hyperparameter optimization.

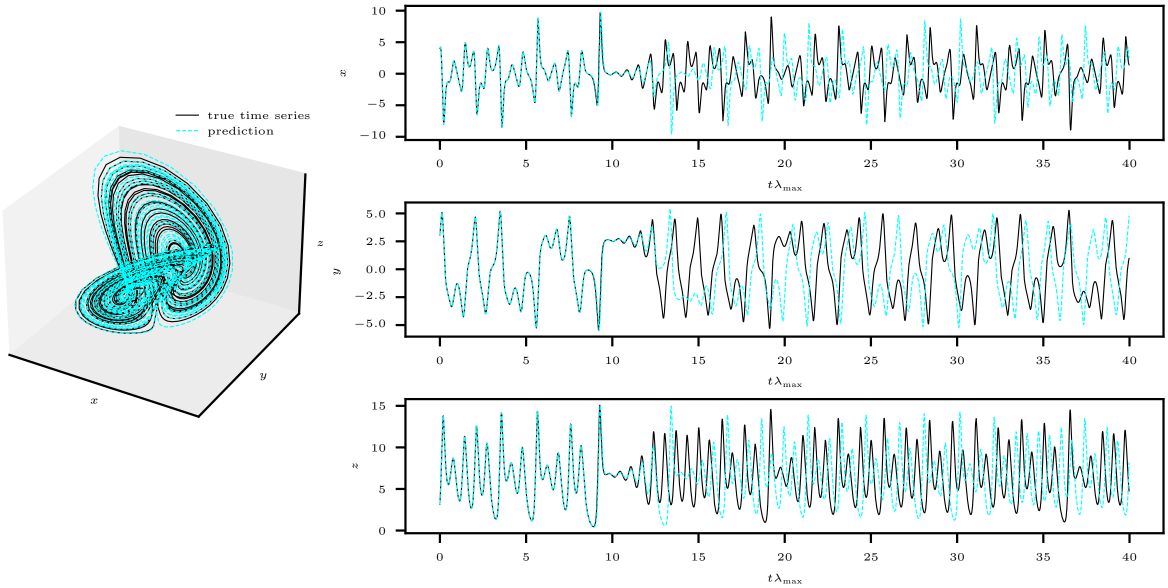


Fig. 51: Example of predicted and real continuation of the Rucklidge system. The predicted trajectory is forecasted with the best-performing hyperparameter configuration (see Table 5) obtained by Bayesian hyperparameter optimization.

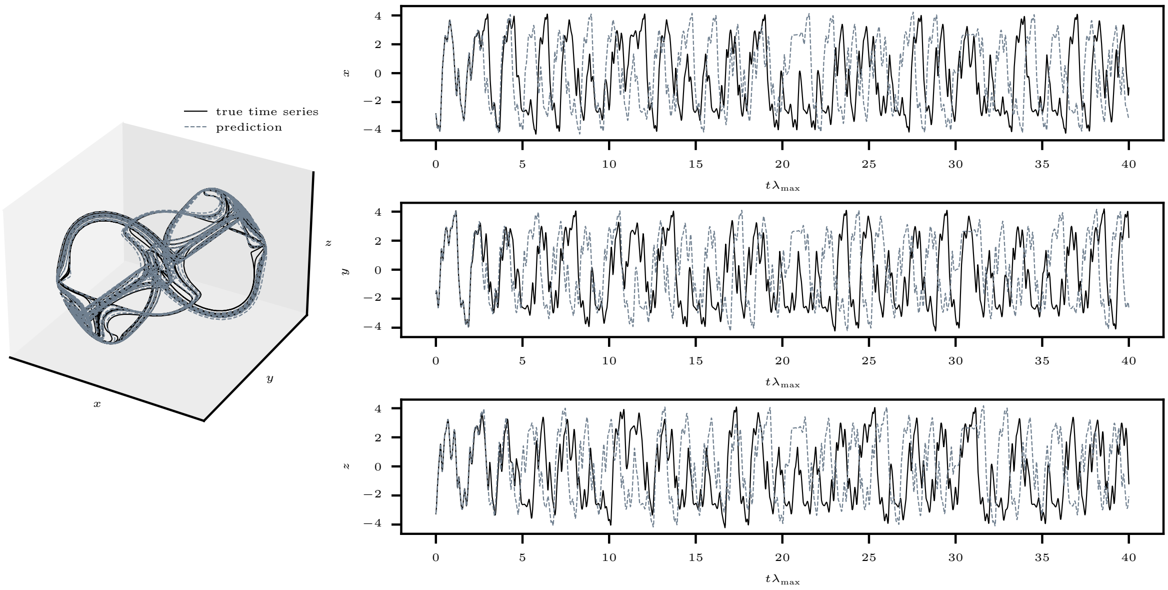


Fig. 52: Example of predicted and real continuation of the Thomas system. The predicted trajectory is forecasted with the best-performing hyperparameter configuration (see Table 5) obtained by Bayesian hyperparameter optimization.

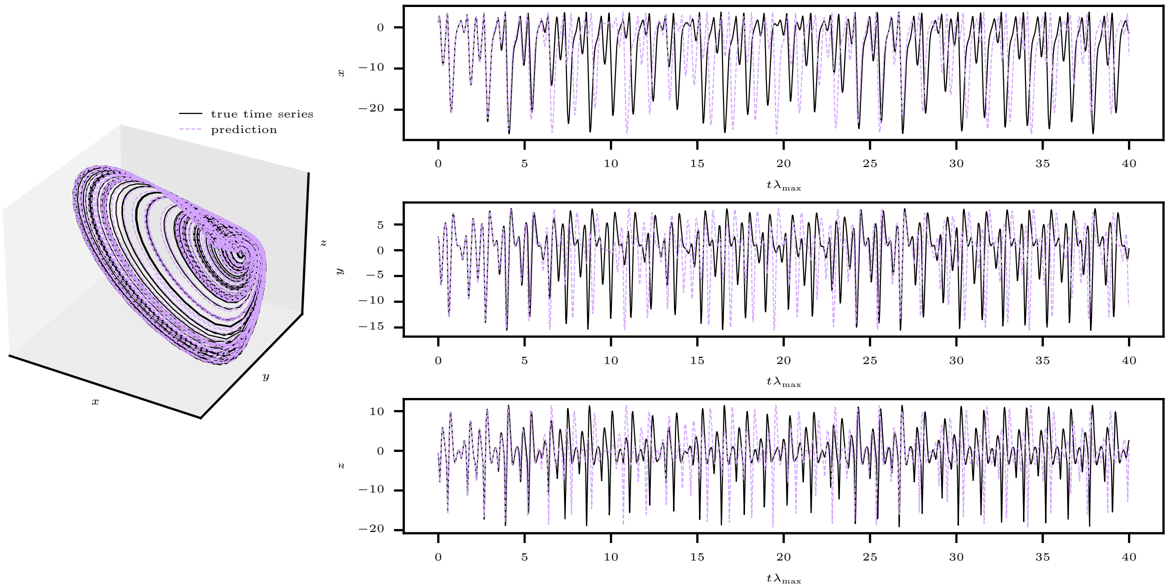


Fig. 53: Example of predicted and real continuation of the WINDMI system. The predicted trajectory is forecasted with the best-performing hyperparameter configuration (see Table 5) obtained by Bayesian hyperparameter optimization.

References

[1] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge University Press, 2019.

[2] Zhixin Lu, Jaideep Pathak, Brian Hunt, Michelle Girvan, Roger Brockett, and Edward Ott. Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(4):041102, 04 2017.

[3] Jaideep Pathak, Zhixin Lu, Brian Hunt, Michelle Girvan, and Edward Ott. Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27, 10 2017.

[4] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120:024102, Jan 2018.

[5] Zhixin Lu, Brian R. Hunt, and Edward Ott. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):061104, 06 2018.

[6] Roland S. Zimmermann and Ulrich Parlitz. Observing spatio-temporal dynamics of excitable media using reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(4):043118, 04 2018.

[7] Ling-Wei Kong, Gene Brewer, and Ying-Cheng Lai. Reservoir-computing based associative memory and itinerancy for complex dynamical attractors. *Nature Communications*, 15, 06 2024.

[8] Xin Li, Qunxi Zhu, Chengli Zhao, Xiaojun Duan, Bolin Zhao, Xue Zhang, Huan-Fei Ma, Jie Sun, and Wei Lin. Higher-order granger reservoir computing: Simultaneously achieving scalable complex structures inference and accurate dynamics prediction. *Nature Communications*, 15:2506, 03 2024.

[9] Seol-Hyun Noh. Analysis of gradient vanishing of rnns and performance comparison. *Information*, 12(11), 2021.

[10] Herbert Jaeger. The"echo state"approach to analysing and training recurrent neural networks. 2001.

[11] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, 14(11):2531–2560, 11 2002.

[12] Gordon E. Moore. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff. *IEEE Solid-State Circuits Society Newsletter*, 11(3):33–35, 2006.

[13] Thomas N. Theis and H.-S. Philip Wong. The end of moore's law: A new beginning for information technology. *Computing in Science Engineering*, 19(2):41–50, 2017.

[14] David Deutsch and Roger Penrose. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[15] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

[16] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1), February 2022.

[17] Jonathan Lau, Kian Lim, Harshank Shrotriya, and Leong Kwek. Nisq computing: where are we and where do we go? *AAPPS Bulletin*, 32, 09 2022.

[18] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, July 2019.

[19] L. Appeltant, M.C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C.R. Mirasso, and I. Fischer. Information processing using a single dynamical node as complex system. *Nature Communications*, 2(1):1–6, 2011.

[20] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express*, 20(3):3241–3249, Jan 2012.

[21] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar. Optoelectronic reservoir computing. *Scientific Reports*, 2(1), February 2012.

[22] Daniel Brunner, Miguel C Soriano, Claudio R Mirasso, and Ingo Fischer. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature communications*, 4:1364, 2013.

[23] Damien Woods and Thomas J. Naughton. Optical computing: Photonic neural networks. *Nature Physics*, 8(4):257–259, Apr 2012.

[24] K. Vandoorne, Pauline Mechet, Thomas Van Vaerenbergh, Martin Fiers, G. Morthier, David Verstraeten, Benjamin Schrauwen, J. Dambre, and Peter Bienstman. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature communications*, 5:3541, 03 2014.

[25] Guy Van der Sande, Daniel Brunner, and Miguel C. Soriano. Advances in photonic reservoir computing. *Nanophotonics*, 6(3):561–576, 2017.

[26] Adam Z. Stieg, Audrius V. Avizienis, Henry O. Sillin, Cristina Martin-Olmos, Masakazu Aono, and James K. Gimzewski. Emergent criticality in complex turing b-type atomic switch networks. *Advanced Materials*, 24(2):286–293, 2012.

[27] Keisuke Fujii and Kohei Nakajima. Harnessing disordered-ensemble quantum dynamics for machine learning. *Physical Review Applied*, 8(2), aug 2017.

[28] Keisuke Fujii and Kohei Nakajima. Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices, 2020.

[29] Kohei Nakajima, Keisuke Fujii, Makoto Negoro, Kosuke Mitarai, and Masahiro Kitagawa. Boosting computational power through spatial multiplexing in quantum reservoir computing. *Physical Review Applied*, 11(3), mar 2019.

[30] Rodrigo Martínez-Peña, Gian Luca Giorgi, Johannes Nokkala, Miguel C. Soriano, and Roberta Zambrini. Dynamical phase transitions in quantum reservoir computing. *Physical Review Letters*, 127(10), aug 2021.

[31] Jiayin Chen and Hendra I. Nurdin. Learning nonlinear input–output maps with dissipative quantum systems. *Quantum Information Processing*, 18(7), May 2019.

[32] R. Martínez-Peña, J. Nokkala, G. L. Giorgi, R. Zambrini, and M. C. Soriano. Information processing capacity of spin-based quantum reservoir computing systems. *Cognitive Computation*, oct 2020.

[33] Quoc Hoan Tran and Kohei Nakajima. Higher-order quantum reservoir computing, 2020.

[34] L. Domingo, M. Grande, G. Carlo, F. Borondo, and J. Borondo. Optimal quantum reservoir computing for market forecasting: An application to fight food price crises. 11 2023.

[35] Sanjib Ghosh, Andrzej Opala, Michał Matuszewski, Tomasz Paterek, and Timothy C. H. Liew. Quantum reservoir processing. *npj Quantum Information*, 5(1), apr 2019.

[36] Sanjib Ghosh, Tomasz Paterek, and Timothy C. H. Liew. Quantum neuromorphic platform for quantum state preparation. *Physical review letters*, 123 26:260404, 2019.

[37] Jiayin Chen, Hendra I. Nurdin, and Naoki Yamamoto. Temporal information processing on noisy quantum computers. *Physical Review Applied*, 14(2), aug 2020.

[38] Makoto Negoro, Kosuke Mitarai, Keisuke Fujii, Kohei Nakajima, and Masahiro Kitagawa. Machine learning with controllable quantum dynamics of a nuclear spin ensemble in a solid, 2018.

[39] Pere Mujal, Rodrigo Martínez-Peña, Johannes Nokkala, Jorge García-Beni, Gian Luca Giorgi, Miguel C. Soriano, and Roberta Zambrini. Opportunities in quantum reservoir computing and extreme learning machines. *Advanced Quantum Technologies*, 4(8):2100027, jun 2021.

[40] Tanjung Krisnanda, Tomasz Paterek, Mauro Paternostro, and Timothy C. H. Liew. Quantum neuromorphic approach to efficient sensing of gravity-induced entanglement. *Physical Review D*, 107(8), apr 2023.

[41] Johannes Nokkala, Rodrigo Martínez-Peña, Gian Luca Giorgi, Valentina Parigi, Miguel C. Soriano, and Roberta Zambrini. Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing. *Communications Physics*, 4, 2021.

[42] 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. *Physical Review Research*, 3(1), jan 2021.

[43] Sanjib Ghosh, Tanjung Krisnanda, Tomasz Paterek, and Timothy C. H. Liew. Realising and compressing quantum circuits with quantum reservoir computing. *Communications Physics*, 4(1), may 2021.

[44] Akitada Sakurai, Marta P. Estarellas, William J. Munro, and Kae Nemoto. Quantum extreme reservoir computation utilizing scale-free networks. *Phys. Rev. Appl.*, 17:064044, Jun 2022.

[45] Mantas Lukoeviius, Herbert Jaeger, and Benjamin Schrauwen. Reservoir computing trends. *KI - Künstliche Intelligenz*, 26:365–371, 2012.

[46] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[47] Mantas Lukoševičius. *A Practical Guide to Applying Echo State Networks*, pages 659–686. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[48] Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019.

[49] Lyudmila Grigoryeva and Juan-Pablo Ortega. Echo state networks are universal, 2018.

[50] Zhixin Lu, Jaideep Pathak, Brian Hunt, Michelle Girvan, Roger Brockett, and Edward Ott. Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(4):041102, 04 2017.

[51] Alexander Haluszczynski and Christoph Räth. Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing, 2019.

[52] T. L. Carroll and L. M. Pecora. Network structure effects in reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(8):083130, aug 2019.

[53] Aaron Griffith, Andrew Pomerance, and Daniel J. Gauthier. Forecasting chaotic systems with very low connectivity reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, dec 2019.

[54] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[55] Zhixin Lu, Jaideep Pathak, Brian Hunt, Michelle Girvan, Roger Brockett, and Edward Ott. Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(4):041102, 04 2017.

[56] Haochun Ma, Davide Prosperino, and C. Räth. A novel approach to minimal reservoir computing. *Scientific Reports*, 13, 08 2023.

[57] Vadim Oganesyan and David A. Huse. Localization of interacting fermions at high temperature. *Phys. Rev. B*, 75:155111, Apr 2007.

[58] Y. Y. Atas, E. Bogomolny, O. Giraud, and G. Roux. Distribution of the ratio of consecutive level spacings in random matrix ensembles. *Physical Review Letters*, 110(8), February 2013.

[59] Dmitry A. Abanin, Ehud Altman, Immanuel Bloch, and Maksym Serbyn. Colloquium : Many-body localization, thermalization, and entanglement. *Reviews of Modern Physics*, 91(2), May 2019.

[60] Lyudmila Grigoryeva and Juan-Pablo Ortega. Echo state networks are universal. *CoRR*, abs/1806.00797, 2018.

[61] J.C. Sprott. *Chaos and Time-series Analysis.* Oxford University Press, 2003.

[62] G.C. Layek. *An Introduction to Dynamical Systems and Chaos.* 02 2024.

[63] C. Runge. Ueber die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 46:167–178, 1895.

[64] W. Kutta. Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Zeit. Math. Phys.*, 46:435–53, 1901.

[65] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130 – 141, 1963.

[66] Alexander Haluszczynski and Christoph Räth. Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(10):103143, 10 2019.

[67] Alexander Haluszczynski, Jonas Aumeier, Joschka Herteux, and Christoph Räth. Reducing network size and improving prediction stability of reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(6):063136, 06 2020.

[68] Joschka Herteux and Christoph Räth. Breaking symmetries of the reservoir equations in echo state networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(12):123142, 12 2020.

[69] Dennis Duncan and Christoph Räth. Optimizing the combination of data-driven and model-based elements in hybrid reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(10):103109, 10 2023.

[70] Jaideep Pathak, Alexander Wikner, Rebeckah Fussell, Sarthak Chandra, Brian R. Hunt, Michelle Girvan, and Edward Ott. Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(4):041101, 04 2018.

[71] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1):189–208, 1983.

[72] Benoit Mandelbrot. How long is the coast of britain? statistical self-similarity and fractional dimension. *Science*, 156(3775):636–638, 1967.

[73] Peter Grassberger. Generalized dimensions of strange attractors. *Physics Letters A*, 97(6):227–230, 1983.

[74] Alan Wolf, Jack Swift, Harry L. Swinney, and John A. Vastano. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16:285–317, 1985.

[75] Robert Shaw. Strange attractors, chaotic behavior, and information flow. *Zeitschrift für Naturforschung A*, 36(1):80–112, 1981.

[76] Michael T. Rosenstein, James J. Collins, and Carlo J. De Luca. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65(1):117–134, 1993.

[77] Ihsan Pehlivan, Yılmaz Uyaroğlu, and M. Yoğun. Chaotic oscillator design and realizations of the rucklidge attractor and its synchronization and masking simulations. *Scientific Research and Essays*, 5:2210–2219, 08 2010.

[78] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

[79] Adam Giammarese, Kamal Rana, Erik M. Bollt, and Nishant Malik. Tree-based learning for high-fidelity prediction of chaos, 2024.

Acknowledgements

I would like to express my gratitude to my supervisor, Christoph, for his guidance, his insightful feedback throughout the course of this research, his encouragement to follow my own ideas, and for the opportunity to work on such a fascinating topic.

My heartfelt thanks also go to my family. To my parents, Thomas and Stefanie, your endless support, love and encouragement have been a great source of strength for me. To my sister, Philine, I deeply appreciate you always being there for me.

Ehrenwörtliche Erklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet zu haben.

Declaration of authorship

I hereby declare that this thesis is my own work, and that I have not used any sources and aids other than those stated in the thesis.

Location, date	Joel Steinegger