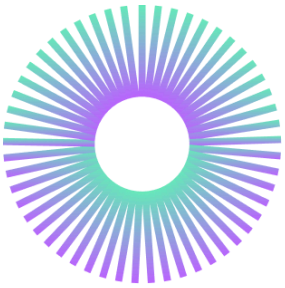
UIMP Universidad Internacional Menéndez Pelayo

ITCL CENTRO TECNOLÓGICO

IFISC Universitat de les Illes Balears CSIC



Master's Degree in Quantum Technologies

Academic year: 2024 - 2025

Tensor Networks for the simulation of Quantum Extreme Learning Machines

Author: Víctor Bayona Marchal
Directors: Gian Luca Giorgi (IFISC)
Roberta Zambrini (IFISC)
Alejandro Mata Ali (ITCL)

Contents

Acronyms

- TN: *Tensor Network*.
- AI : *Artificial Intelligence*.
- ED: *Exact Diagonalization*.
- TDVP: *Time-Dependent Variational Principle*.
- MPS: *Matrix Product State*.
- MPO: *Matrix Product Operator*.
- MPDO: *Matrix Product Density Operator*.
- QML: *Quantum Machine Learning*.
- QELM: *Quantum Extreme Learning Machine*.
- QRC: *Quantum Reservoir Computing*.
- ML: *Machine Learning*.
- QNN: *Quantum Neural Network*.
- VQA: *Variational Quantum Algorithm*.
- MNIST: *Modified National Institute of Standards and Technology*.
- ELM : *Extreme Learning Machine*.
- SLFN : *Single-Layer Feedforward neural Network*.
- RC : *Reservoir Computing*,
- CPTP: *Completely Positive an Trace Preserving*.
- SVD: *Singular Value Decomposition*.

Abstract

This project investigates the potential of a spin chain system as a computational resource in machine learning. In particular, we explore the possibility of using such a quantum system as the substrate for a Quantum Extreme Learning Machine. For that sake, e simulate the time evolution of the system using tensor networks techniques, specifically the two-site Time-Dependent Variational Principle, and benchmark the results against Exact Diagonalization, analyzing the main weaknesses and sources of error of the method. Measured observables — one-, two-, three-, and higher-order correlators— serve as input features for a supervised classification model. The results show that models trained with measures of time-evolved quantum states do not outperform a baseline model without time evolution, indicating that the learning capability primarily depends on the quantum data encoding and measurement scheme, consistent with current Quantum Machine Learning theory. Lastly, the work discusses prospects for implementing tensor network simulations in quantum reservoir computing architectures.

Abstract

En este trabajo se ha estudiado el potencial de una cadena de espines como recurso computacional en aprendizaje automático. En particular, se ha explorado la posibilidad de usar dicho sistema cuántico como sustrato de una *Quantum Extreme Lerning Machine*. Para ello, se ha simulado la evolución del sistema mediante el uso de *Tensor Networks*, empleando el método *Time-Dependent Variational Principle* de dos sitios, y se ha comparado su precisión frente a la diagonalización exacta del Hamiltoniano. De esta manera, se han podido observar las debilidades y fuentes de error del algoritmo de evolución temporal TDVP. Los observables medidos — correladores a uno, dos, tres y más cuerpos— se utilizan como características de entrada para un modelo de clasificación simple. Los resultados muestran que los modelos entrenados con medidas de observables sobre estados evolucionados temporalmente no superan al modelo base sin evolución, lo que sugiere que la capacidad de aprendizaje depende principalmente del esquema de codificación y medida, en línea con la teoría actual de aprendizaje automático cuántico. Finalmente, se discuten distintas perspectivas para el uso de *Tensor Networks* en otras arquitecturas de computación cuántica como *Quantum Reservoir Computing*.

1 Introduction

Over the last few decades, machine learning and artificial intelligence have revolutionized both scientific research and industrial applications, driving breakthroughs in different scientific fields. However, despite their transformative power, classical AI models are often hindered by substantial computational costs, energy consumption, and a lack of interpretability. Training large-scale models such as deep neural networks demands significant computational infrastructure and energy, raising concerns about sustainability and accessibility. Furthermore, the "black-box" nature of many AI models poses a serious challenge for applications where explainability and transparency are critical, such as healthcare or scientific inference.

Quantum Machine Learning has emerged as a promising approach to address some of these limitations by leveraging the inherent properties of quantum mechanics to process information more efficiently. Recent QML paradigms include Variational Quantum Algorithms [1], which use parameterized quantum gates optimized via classical routines. While these models offer theoretical speedups and quantum advantages in specific cases, they are not without drawbacks. For example, recent studies point out that, in general, supervised QML models are actually kernel methods [2], and, thus, the performance of the model depends on the expressivity of the model, mainly determined by the data encoding scheme. For VQAs, the expressivity power increases with the depth of the circuit. However, these same models face the barren plateau problem, where the optimization landscape becomes exponentially flat with system size, hindering the training of deep circuits [3].

To overcome these challenges and better align with the constraints of current quantum hardware, alternative approaches based on unconventional computing have gained interest. In particular, reservoir computing frameworks that utilize quantum systems as untrained dynamical substrates are a possible solution [4]. These frameworks have their own classical counterparts [5]. Within the realm of quantum mechanics, these models exploit the dynamics of quantum many-body systems to perform nonlinear temporal processing of the input data without the need for training the quantum system itself. This approach is especially suitable for the noisy intermediate-scale quantum (NISQ) era, where limited coherence times and gate fidelity restrict the depth and reliability of gate-based models. Reservoir-based quantum models, such as Quantum Extreme Learning Machines (QELM) and Quantum Reservoir Computing (QRC), offer an analog alternative, enabling learning tasks for classification, regression and time series forecasting, leveraging the intrinsic dynamics of physically realizable systems such as spin chains [6] or quantum optical networks [7], [8].

Despite the theoretical potential of quantum computing, its practical realization remains a challenge. One of the most significant obstacles is the fragility of quantum hardware: current quantum processors are susceptible to decoherence, gate noise, and readout errors, which severely limit the depth and reliability of quantum circuits. Therefore, to explore the potential of different quantum computing paradigms, simulation is a fundamental tool nowadays. However, simulating quantum systems on classical computers poses an intrinsic difficulty due to the exponential scaling of Hilbert space with the number of qubits, making exact simulation intractable for all but the

smallest systems. In this context, tensor network (TN) methods. TNs allow for an efficient and scalable representation of quantum many-body states, and provide not only a practical tool for exploring quantum machine learning models on classical hardware but also deeper insight into the entanglement dynamics and memory properties that underlie their computational performance [9]. Also, there has been recent work on the use of quantum-inspired TN methods for QELM [10]. This highlights the importance of further research in this direction.

The primary objective of this project is to explore the machine learning potential of quantum many-body systems by leveraging TN techniques for efficient and scalable simulation. Specifically, we consider a one-dimensional spin chain as a quantum reservoir and analyze how changes in its physical parameters impact its performance on learning tasks. This approach not only provides insights into the interplay between physical dynamics and computational power but also highlights the utility of TN methods in simulating quantum machine learning models and increasing their explainability. For that sake, this work is structured as follows: in section 2 we explain the theoretical foundations of the two main quantum-reservoir-based models: QELM 2.1 and QRC 2.2. In section 3, we introduce the necessary TN tools to simulate the quantum reservoir. In section 4, we present the methodology followed in the analyses of the influence of physical parameters over computational capabilities. Finally, in section 5, the evolution of the quantum system using TN methods and the exact diagonalization of the Hamiltonian are compared 5.1. Also, a study of the performance of the final model when classifying the MNIST dataset, depending on the physical parameters of the Hamiltonian and the evolution time, is carried out 5.2. Lastly, the learning speed in terms of sample efficiency is also examined for different parameters in the Hamiltonian and evolution times 5.3. Additionally, an approach for using TN techniques to simulate QRC will be briefly discussed in 5.4.

2 Quantum Reservoir Models

Quantum reservoir learning models have emerged as an efficient and flexible alternative in quantum machine learning, particularly effective for temporal tasks, as well as classification and regression problems. Unlike other architectures, which require backpropagation and the training of internal parameters, reservoir-based models rely on a quantum dynamical system, called the reservoir, to embed input data into a feature space. Only the output layer, built up by measurements, is trained, typically via simple linear regression, which significantly reduces training complexity.

The two main quantum reservoir-based models are Quantum Reservoir Computing (QRC) and Quantum Extreme Learning Machines (QELM). In QRC, an input signal is encoded into a quantum system whose time evolution under a fixed Hamiltonian generates feature vectors from the input data. Every data point from the signal is embedded into the quantum system, and then the system is evolved for a certain time step. A set of observables is then measured. Once the whole series has been introduced in the reservoir, a classical readout layer is trained to map all the observables measured after each time step of this evolving quantum state to the desired outputs, while the quantum dynamics themselves remain untrained. The temporal ordering of the introduced data is fundamental, as it induces similar temporal trends in the quantum system that are used to forecast

these sequential series. QELMs follow a very similar approach, but with a simpler scope, targeting static tasks, such as classification: the quantum system is reset and measured only once per input, eliminating the quantum state modification during the temporal evolution. The readout is again classically trained on the measured quantum features. QELMs can be viewed as a special case of QRC, where the corresponding input vector uniquely defines the reservoir's state, and thus, there does not exist dependence of the actual state on previous states.

2.1 Quantum Extreme Learning Machines

Extreme learning machines (ELM), first introduced by Minsky and Papert [11] and recently returned to the literature by Huang [12], are a particular example of single-layer feedforward neural networks, where input weights and biases are chosen randomly, and only output weights are trained using, for example, a linear regression, to fit a certain function. In fact, for any activation function of $\mathcal{C}^\infty$ (space of infinitely derivable functions), the input weight vectors and biases can be chosen randomly from any continuous probability distribution and the output matrix of the hidden layer can be inverted to obtain the output weights that fit the target function [12]. The algorithm behind ELM for a given training set $\mathbf{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) / \boldsymbol{x}_i \in \mathbb{R}^n, \boldsymbol{y}_i \in \mathbb{R}^m, i = 1, ..., N\}$, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are input and output vectors respectively, related by $y = Hx$, for a certain matrix H, an activation function $g(x) \in \mathcal{C}^\infty$, and the number of hidden nodes of $\hat{N}$ is the following:

1. Randomly choose the values for input weight vectors and biases $\boldsymbol{w_j}$, b_j for $j = 1, ..., \hat{N}$.
2. Compute the hidden layer output matrix $\boldsymbol{H}$, where the j-th column of $\boldsymbol{H}$ is given by $[g(\boldsymbol{w_j x_1} + b_j), ..., g(\boldsymbol{w_j x_N} + b_j)]^T$.
3. Compute the estimated output $\hat{\boldsymbol{x}} = \boldsymbol{H}^\dagger \boldsymbol{y}$, where $\boldsymbol{H}^\dagger$ is the Moore-Penrose generalized inverse of $\boldsymbol{H}$ and $\boldsymbol{y} = [\boldsymbol{y_1}, ..., \boldsymbol{y_N}]^T$.

This paradigm is designed to solve some issues regarding the usual SLFN, which include a high computational cost in the training phase and the need for an iterative approach through backpropagation.

A natural extension to ELM is Quantum Extreme Learning Machines (QELM), where the ELM processing is performed with a quantum system acting as a reservoir. Using a quantum hidden layer in ELM can have different advantages: First, QELM can be used to process quantum data [13]. Second, the quantum system evolves in a high-dimensional Hilbert space. This embedding from raw vectors in a training data set to quantum states in a Hilbert space can be seen as a feature map and be useful to capture highly nonlinear correlations in the data. In addition, training a QELM remains a simple optimization problem. Therefore, trainability is guaranteed even on noisy hardware, avoiding noise-induced quantum barren plateaus (which appear, for example, in Variational Quantum Algorithms) [14]. One of the main challenges in Quantum Machine Learning (QML) for classic data tasks and, in particular, in the field of QELM, is the expressivity power depending on the encoding scheme. Some recent studies point out that, in most cases, quantum models can be seen as partial Fourier series in the data, and the predictive capacity of these models

is directly related to the available Fourier frequencies in the model, which depend on the encoding scheme [15].

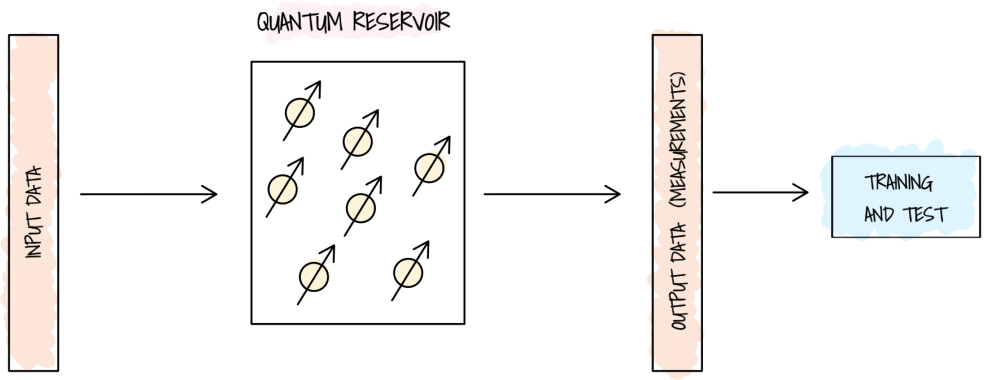


Figure 1: General architecture of QELM: the data is encoded into a quantum state of the quantum reservoir. This is then evolved in time, and then the output data is collected from certain measurements over the system. This output data is finally used to fit a certain function.

The general scheme of QELM can be seen in 1, and consists of the following components:

- Data encoding: the data is encoded into a quantum state. There are several different encoding strategies that can affect QELM performance [16]. This step, thus, transforms a data vector into a quantum state:

$$\boldsymbol{x_i} \longrightarrow \rho(\boldsymbol{x}_i, t=0)\,. \tag{2.1}$$

- Reservoir evolution : The quantum state evolves over time $t = \tau$ according to the dynamics of the reservoir. A review of some common platforms can be found in [4]. The quantum state thus undergoes a transformation given by the time evolution operator:

$$\rho(\boldsymbol{x}_i, t=0) \longrightarrow U(t=\tau)\rho(\boldsymbol{x}_i, t=0)U(t=\tau)^\dagger = \rho(\boldsymbol{x}_i, t=\tau)\,, \tag{2.2}$$

  where $U(t=\tau) = \exp(-iH\tau)$ and H is the Hamiltonian of the quantum reservoir.

- Readout: After the state has evolved, the reservoir is measured on a set of observables $\{O_1, ..., O_M\}$. The expected value of an observable over a quantum state is given by

$$\langle O \rangle_\rho = Tr[O\rho]\,. \tag{2.3}$$

  The election of the observables is also critical, since the geometry of the feature space to which the data are transformed will depend on these.

- Training: After measuring the quantum state, the output data are used to train a simple architecture to fit a certain function f such that $f(\boldsymbol{x}_i) \approx \boldsymbol{y}_i \quad \forall i = 1, ..., N$. This can be used for both classification and regression.

2.2 Quantum Reservoir Computing

Classical Reservoir Computing (RC) is a computational technique that deals with dynamical systems (reservoirs) that can map a certain input to a given state vector. Let s_k and $\vec{x}_k$ be such input and state vector, respectively, then the dynamical system aims to map a function f such that

$$\vec{x}_k = f(s_k, \vec{x}_{k-1}) \,, \tag{2.4}$$

where $\vec{x}_{k-1}$ is the previous state of the dynamical system [17, 18]. Therefore, in this situation, unlike for QELM, the state of the reservoir at a certain time, $\vec{x}_k$, depends on its previous state, $\vec{x}_{k-1}$. This means that the current state also depends on the state of every past input, as

$$\vec{x}_k = f(s_k, \vec{x}_{k-1}) = f(s_k, f(s_{k-1}, \vec{x}_{k-2})) = f(s_k, f(s_{k-1}, f(s_k, f(s_{k-2}, \vec{x}_{k-3})))) = \ldots \quad . \tag{2.5}$$

In this way, the state of the reservoir is used as a memory. In addition, the map f should be contractive such that $\vec{x}_k$ only depends on recent dynamics and input history [19, 20, 21]. This ensures independence of distant past: different initial states for the reservoir end up in the same state for the same data inputs (this condition is known as echo state property); and also convergence: the input sequences that coincide during some recent times lead to the same outputs (this condition is known as fading memory). Finally, f should also be surjective, meaning that different inputs are matched to different outputs [18].

After the input data $\{s_k\}$ has been injected and the system evolved accordingly, a function h is optimized (usually through a simple linear regression) such that the output sequence $\{\hat{y}_k = h(\vec{x}_k)\}$ matches the original output sequence $\{y_k\}$. The overall objective of RC is to embed the input s_k into $\vec{x_k}$ in an appropriate feature space and then approximate the desired output y_k.

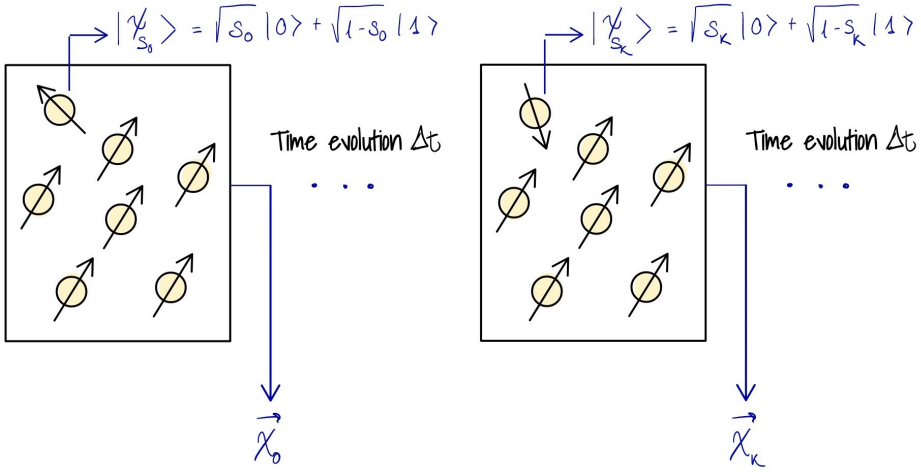


Figure 2: General architecture of QRC: the data is encoded into the state of the first qubit in the reservoir. The whole system is then evolved in time, and then the output data is collected from certain measurements over the system. This output data through all the input points is finally used to fit a certain function.

Again, a natural extension for RC is to consider a quantum system as a reservoir, and harness the disordered quantum dynamics as the feature map f [6]. A graphical description of the procedure can be seen in 2. Although there are different QRC frameworks, we will follow the one discussed in [6]. The quantum reservoir consists of N qubits whose dynamics are given by a certain Hamiltonian H. Let, again, $\{s_k\}$ and $\{y_k\}$ be the input and output sequences. For the sake of simplicity, we will consider $s_k \in [0,1]\,\forall k$, although this idea can also be extended to vector inputs. The scheme of QRC is the following:

- Initialization: Set the initial quantum state in a random mixed state ρ.
- Data Encoding: For the first input s_0, set the state of the first qubit (it could be any other, but we use the first one as a convention) to $|\psi_0\rangle = \sqrt{s_0}\,|0\rangle + \sqrt{1-s_0}\,|1\rangle$ and couple it to the rest of the reservoir
$$\rho_0 \longrightarrow |\psi_0\rangle\langle\psi_0| \otimes Tr_1[\rho]\,, \tag{2.6}$$
where $Tr_1[\rho]$ is the partial trace of the state ρ with respect to the first qubit.
- Time Evolution: Let the system evolve Δt. That is, the system will undergo the transformation
$$\rho_0 \longrightarrow \exp(-iH\Delta t)\rho_0 \exp(iH\Delta t)\,. \tag{2.7}$$
- Measurement: Measure a set of observables $\{B_i\}$ from the set of N-qubit products of Pauli operators $\{I, X, Y, Z\}^{\otimes N}$. Thus, we build a vector $\vec{x}_0$, where the i-th component is the measurement of the i-th operator $x_{0i} = Tr[\rho_0 B_i]$.
- Full data injection: Data encoding, evolution, and measurement steps are repeated for each input s_k. The result is a set of measurement vectors $\{\vec{x}_k\}$ where x_{ki} is the measurement $Tr[\rho_k B_i]$; i.e, the expected value of B_i over the quantum reservoir at time step k.
- Training: A function h is trained through a simple architecture (a linear regression, for example) to match the sequence $\{h(\vec{x}_k)\}$ to $\{y_k\}$, typically a temporal sequence.

As already mentioned, QELM can be seen as a particular case of QRC where the data encoding of a certain input is performed without any dependence on the previous state of the reservoir. The state of the reservoir is reset after each input injection, set with the state corresponding to the data input, evolved in time, and then measured. Although simpler than QRC, QELM cannot perform temporal tasks, as it lacks memory capacities.

3 Tensor Networks

Tensor Networks (TNs) is the term used to refer to the mathematical framework for the representation of high-dimensional data. Originally developed in the context of condensed matter physics, these ideas have now found application in quantum information theory and quantum many-body systems. In particular, TN are a powerful mechanism to describe complex quantum systems [9]. The key idea in TNs is to find efficient tensor structures to represent quantum states and operators.

For example, in a quantum system given by N 1/2-spin particles, a quantum state can be represented by a 2^N dimensional vector. This can be seen as a tensor with N indices, where each index has two possible values, corresponding to the two states of the local systems (in this example, spin "up" and "down"). This tensor can later be decomposed into a set of N interconnected tensors to represent the same state more efficiently; i.e, with fewer coefficients. To find such an efficient representation, we use the Schmidt Decomposition of quantum states, which is based on the Singular Value Decomposition (SVD).

In mathematics, tensors are a well-known algebraic structure that encodes multi-linear relationships between objects associated with a vector space. A usual formal definition for a tensor is the following:
Definition : Let V a vector space such that $dim_{\mathbb{C}}(V) = d$ and V* its dual space. Then, a (p,q)-ranked tensor is a multilinear map

$$T : \underbrace{V^* \times \cdots \times V^*}_{p} \times \underbrace{V \times \cdots \times V}_{q} \longrightarrow \mathbb{C}\,. \tag{3.1}$$

3.1 Schmidt Decomposition. Singular Value Decomposition

Given a quantum state $|\psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$, it is always possible to find orthonormal basis $\{|\alpha\rangle_A\}, \{|\alpha\rangle_B\}$ for the Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively (called Schmidt basis), positive coefficients $\{\lambda_\alpha\}$ (known as Schmidt coefficients) and $\chi \leq \min(\dim(\mathcal{H}_A), \dim(\mathcal{H}_B))$ such that the quantum state can be written as

$$|\psi\rangle_{AB} = \sum_{\alpha=1}^{\chi} \lambda_\alpha |\alpha\rangle_A |\alpha\rangle_B\,. \tag{3.2}$$

This representation can be found using SVD. If we write the quantum state as

$$|\psi\rangle_{AB} = \sum_{i,j} \psi_{ij} |i\rangle_A |j\rangle_B\,, \tag{3.3}$$

where $\{|i\rangle_A\}$ and $\{|j\rangle_B\}$ are arbitrary orthonormal basis in $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively, then, SVD over the tensor ψ_{ij} yields matrices U, Λ, V such that

$$\psi_{ij} = \sum_{\alpha,\beta=1}^{\chi} U_{i\alpha}\Lambda_{\alpha\beta}V^\dagger_{\beta j}\,, \tag{3.4}$$

where the Schmidt basis is encoded in the matrices U and V, and Λ is a $\chi \times \chi$ diagonal matrix containing the singular values (Schmidt coefficients). Equation 3.4 can be expressed in a diagrammatic form as in Figure 3.

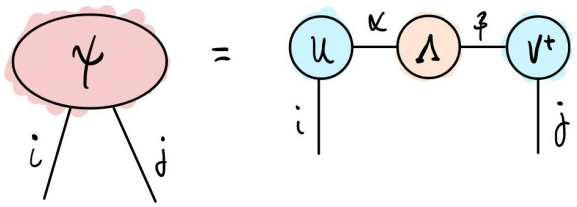


Figure 3: Graphical representation of SVD of the coordinates tensor of a quantum state. Free legs correspond to free indices and linked legs are contracted indices as in Einstein's notation.

In this diagrammatic representation, tensors have "legs" that can be free, which correspond to physical indices and linked, which correspond to contraction indices; i.e, there is a summation over them.

A final result about Schmidt decomposition is that the expansion can be truncated up to a certain precision by keeping only Schmidt coefficients above a threshold. The quantum state entanglement entropy S verifies [9]

$$S = -\sum_{\alpha=1}^{\chi} \lambda_\alpha^2 \log \lambda_\alpha^2 \leq \log \chi \,. \tag{3.5}$$

Therefore, by selecting the highest χ Schmidt coefficients, we are building approximations of quantum states, limiting the maximum value of their entanglement entropy.

3.2 Matrix Product State

A Matrix Product State (MPS) is a TN structure used to represent pure quantum states. Given an N-body quantum system, a quantum state can be written as

$$|\psi\rangle = \sum_{i_1} \sum_{i_2} ... \sum_{i_N} \psi_{i_1,i_2,...,i_N} |i_1\rangle |i_2\rangle ... |i_N\rangle \ , \tag{3.6}$$

where the indices $i_1, ..., i_N \in \{1, ..., \dim(\mathcal{H})\}$, in the assumption that all local Hilbert spaces are equal. Therefore, we can apply the Schmidt decomposition iteratively over this quantum state by reshaping the indices. This process is described in Figure 4

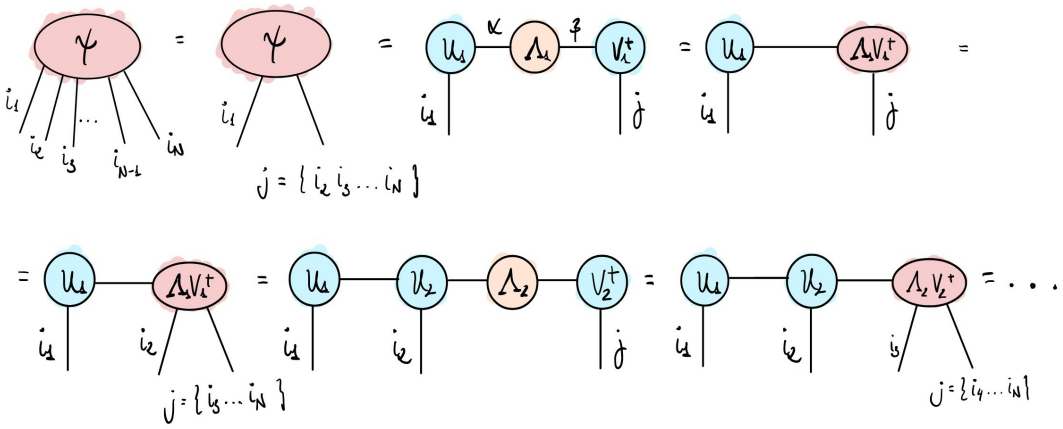


Figure 4: Graphical representation of the method to obtain the MPS of a quantum state. We iteratively apply the schema of Schmidt decomposition pictured in 3 and reshape the indices.

Mathematically, we apply the Schmidt decomposition iteratively:

1. First, we reshape the indices such that $\mathcal{H}_A = \mathcal{H}_1$ and $\mathcal{H}_B = \mathcal{H}_2 \otimes ... \otimes \mathcal{H}_N$.
2. Secondly, we apply the Schmidt decomposition to $|\psi\rangle_{AB}$.
3. Now that the first tensor has already been built, we update $\mathcal{H}_A = \mathcal{H}_2$ and $\mathcal{H}_B = \mathcal{H}_3 \otimes ... \otimes \mathcal{H}_N$ by reshaping again the indices and absorbing the singular values matrix in the right tensor of SVD decomposition.
4. We repeatedly apply steps 2 and 3 until we have achieved all N tensors.

The final result of this process is depicted in Figure 5. The MPS is made up of N tensors, each with one free leg corresponding to one physical index. Furthermore, these tensors are connected by contracted legs. The order of these contracted legs, χ, is called bond dimension, and is a measure of how accurately the MPS represents the original quantum state in terms of entanglement entropy.

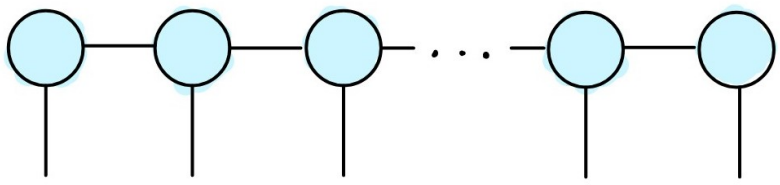


Figure 5: Graphical representation of an MPS tensor network for a quantum state.

The MPS for a separable state, for example, can be built with $\chi = 1$. As the quantum state grows in entanglement, larger values of χ will be required.

3.3 Matrix Product Operator

Matrix Product Operators (MPO) were first introduced as TN ansatz for density matrices [22]. They are also used to represent operators. To find the MPO representation of an object O, either a density matrix or an operator, we write it down as follows:

$$O = \sum_{i_1,...,i_N} \sum_{j_1,...,j_N} O_{i_1,...,i_N,j_1,...,j_N} |i_1\rangle ... |i_N\rangle \langle j_1| ... \langle j_N| \ , \tag{3.7}$$

where $\{i_1, ..., i_N\}$ are "output" indices, referring to elements in the total Hilbert space, and $\{j_1, ..., j_N\}$ are "input" indices, referring to elements in the dual total Hilbert space. Therefore, we can reshape the local indices $\{(i_l, j_l)\}$ into a single composite index k_l, allowing us to apply the same procedure used for MPS.

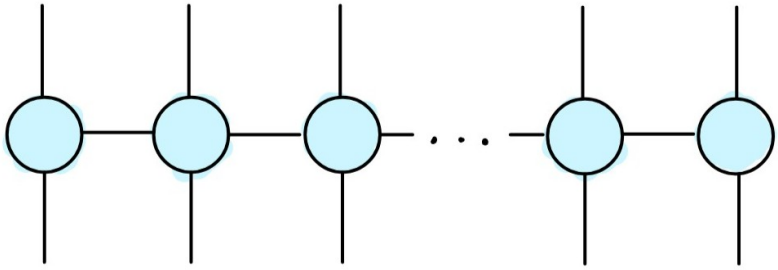


Figure 6: Graphical representation of an MPO tensor network for a quantum state.

We can later unpack the indices such that we end up with a structure like the one depicted in Figure 6. The action of an MPO over an MPS can be understood as shown in 7. The physical indices in the MPS are contracted with the output indices in the MPO, and the bond indices are reshaped so that we end up with an MPS structure, where the new physical indices are the input indices of the MPO.

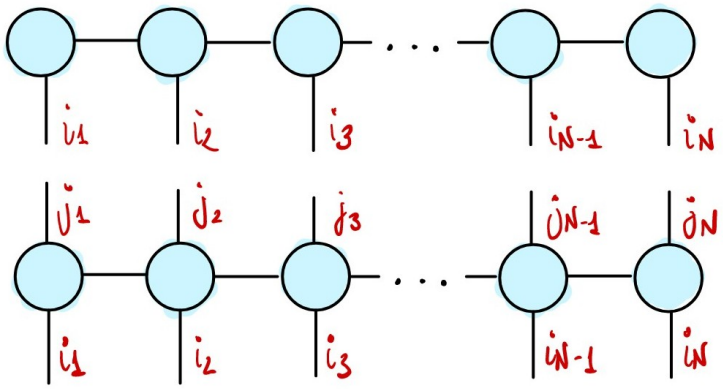


Figure 7: Graphical representation of an MPO acting over an MPS. The MPO legs corresponding to the input indices are contracted with the free legs in the MPS to obtain another MPS.

Other ways to represent mixed states include purification-like structures, known as Matrix Product Density Operators (MPDO) [23]. An ancillary Hilbert space is used to describe the quantum state as a pure state. Then, the MPS of the new quantum state is found, distinguishing between the ancilla and physical indices. To operate with this new MPS, only the physical legs of the TN are manipulated.

3.4 Time Evolution

After introducing the usual TN structures as an efficient representation of states and operators, the next key step is to address time evolution. Quantum states evolve in time according to the Schrödinger equation:

$$i\frac{\partial}{\partial t}\left|\psi\right\rangle = H\left|\psi\right\rangle \ . \tag{3.8}$$

This equation only admits an exact solution for a limited set of cases. Therefore, numerical methods are required. TN methods are a recent addition to this list. Although there are different time evolution TN algorithms, in this work, we will focus on the Time Dependent Variational Principle (TDVP) [24] [25].

We can use the MPS representation of the quantum state in the Schrödinger equation as

$$i\frac{\partial}{\partial t}\left|\psi(U)\right\rangle = H\left|\psi(U)\right\rangle \ , \tag{3.9}$$

where

$$\left|\psi(U)\right\rangle = \sum_{i_1}...\sum_{i_N} U(1)^{\alpha_1}_{i_1} U(2)^{\alpha_1\alpha_2}_{i_2}...U(N)^{\alpha_{N-1}}_{i_N}\left|i_1\right\rangle\left|i_2\right\rangle...\left|i_N\right\rangle \ , \tag{3.10}$$

and we are using the Einstein convention of summation over repeated indices. The MPS $\left|\psi(U)\right\rangle$ belongs to a manifold in the total Hilbert space $\mathcal{M}_\chi \subset \mathcal{H}$. This manifold is defined as the one containing MPS with bond dimension χ. However, applying a Hamiltonian described as an MPO to an MPS will increase the bond dimension. The idea behind TDVP is to restrict the unitary time evolution within the manifold $\mathcal{M}_\chi$. In other words, the maximum value for χ is set during time evolution. Two variants of TDVP arise now: single-site TDVP, where χ_{max} is constant during time evolution, and two-site TDVP, where χ_{max} can grow during time evolution to better enclose the structure of the quantum state [25].

To effectively keep the time evolution of the MPS within the manifold $\mathcal{M}_\chi$, we project the right side of the Schrödinger equation onto the tangent space $\mathbb{T}_{\psi(U)}\mathcal{M}$ defined at the point $\left|\psi(U)\right\rangle$. That is

$$i\frac{\partial}{\partial t}\left|\psi(U)\right\rangle = \hat{P}^{[i]}_{\mathbb{T}_{\psi(U)}\mathcal{M}_\chi} H\left|\psi(U)\right\rangle \ , \tag{3.11}$$

where $\hat{P}^{[i]}_{\mathbb{T}_{\psi(U)}\mathcal{M}_\chi}$ is the projector onto $\mathbb{T}_{\psi(U)}\mathcal{M}_\chi$ subspace. The exact form of $\hat{P}^{[i]}_{\mathbb{T}_{\psi(U)}\mathcal{M}_\chi}$ can be found in [24] [25]. The superscript i refers to whether we are using single-site TDVP or two-site TDVP. The resulting Schrödinger equation is not exactly solvable, but can be variationally approximated using a Lie-Trotter decomposition of the operators and evolving in Δt time steps [26].

The TDVP method encounters 3 types of errors. The first one is related to the limited expressivity

of the MPS for a fixed bond dimension χ. The second one concerns the solution of the differential equations arising from the Lie-Trotter decomposition of the projected Schrödinger equation. This can be amended by reducing the time step Δt in the Lie-Trotter decomposition, as the arising error is $O(\Delta t^2)$. Finally, TN often involve the multiplication of many high-precision numbers, leading to cumulative truncation errors due to finite numerical precision in computations. However, this error scales with the number of tensors in the structure, as well as with the bond dimensions. For reasonably small systems, this does not constitute a main source of error and can be neglected. For larger systems, this must be taken into account.
The computational complexity of the TDVP algorithm scales asymptotically as $O(\chi^3)$. Further information on the method and its application can be found in [27].

4 Methodology

The objective of this work is to study the performance capabilities of a 1/2-spin network described by the transverse-field Ising Hamiltonian plus onsite disorder:

$$H = \sum_{i>j}^{N} J_{ij}\sigma_i^x\sigma_j^x + \frac{1}{2}\sum_{i=1}^{N}(h + D_i)\sigma_i^z \,, \tag{4.1}$$

where N is the number of spins in the all-to-all network, h is the magnetic field, D_i is the onsite disorder, $\sigma^x\,, \sigma^z$ and σ^y are the Pauli matrices and J_{ij} are the spin-spin couplings, randomly, uniformly sampled from the interval $[-J_s/2, J_s/2]$ and D_i is also drawn randomly, uniformly from the interval $[-W, W]$, wehere W is the disorder strength. For convenience, all parameters will be expressed in terms of J_s.

This system will be used as a reservoir for QELM, where the task will be MNIST classification [28]. This system presents different dynamical phases depending on the values of the Hamiltonian [29]. Using the usual ratio of adjacent gaps

$$r_n = \frac{\min\{\delta_{n+1}\,, \delta_n\}}{\max\{\delta_{n+1}\,, \delta_n\}} \,, \tag{4.2}$$

where $\delta_n = E_n - E_{n-1}$ and E_n is the list of ascending eigenvalues of the Hamiltonian, the phase diagram of the system displays four different phases, as shown in 8.

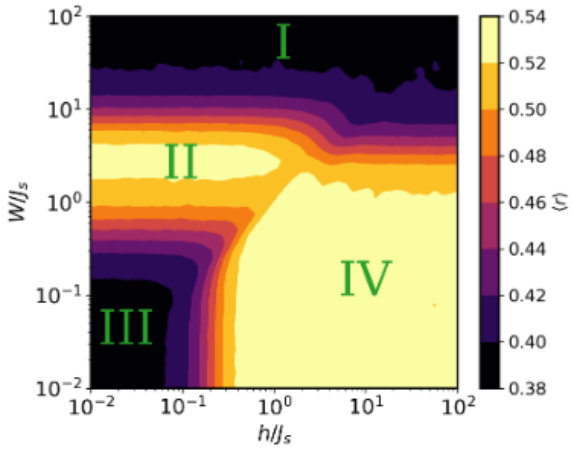


Figure 8: Heat map of $\langle r \rangle$ for different values of the magnetic field h and te onsite disorder strength W in units of J_s. The system has four different phases. Two localization areas (regions I and III) and two ergodicity areas (regions II and IV) [29].

The diagram displays four phases depending on the values of the parameters in the Hamiltonian. There are two localization areas and two ergodic areas (regions I and III, and II and IV, respectively). The simulation of this system and its evolution in time will be performed using TN. The methodology followed in this study is the following:

- First, compare the evolution in time of the system using exact diagonalization techniques with TN methods. For that sake, we have used QuTiP [30] and TeNPy [31] software packages.
- Second, evaluate the performance of QELM in the different regions (localization and ergodicity areas) of the phase transition diagram in 8 evolving for different times. Thus, we can see which is the optimal regime for the system to operate as a QELM.
- Also, we can study the "learning speed" of the system in the different phases. That is, check the performance of the model for increasing training data set sizes and verify which requires shorter processing to achieve optimal performance.
- Finally, an approach towards using TN for QRC simulation will be briefly developed.

The data encoding in the system has been performed using amplitude encoding, as it is the only encoding method that allows the use of the full 28×28 image as a quantum state without dimensionality reduction in a preprocessing step. Also, the whole dataset has not been used. To save time and computational resources, only 10000 images from the training set and 5000 from the test set have been processed out of the MNIST. It is to be understood, therefore, that the performance of the QELM models could be improved using more images.

5 Results

In the following section, we display the results of this work. We will consider different points in the phase diagram 8. The chosen values are shown in 1. We have considered points in the two localization areas (I and III), in the two ergodicity ares (II and IV), including two different points in IV to check numerical stability and in all the transition areas between these four regions (I to II, II to III, III to IV, IV to I). This section is structured as follows: first, we check the accuracy of the TN simulation of the system versus the exact diagonalization method. Secondly, we evaluate the accuracy of the models in time, and compare it with the baseline model, which consists of encoding the images in a quantum state and measuring the observables without performing time evolution. Finally, we examine the learning speed, in terms of accuracy versus training data set size, of the models for the different times. Additionally, we will discuss the possibilities of using TN for the simulation of the QRC framework explained in 2.2. As mentioned in [29], all the parameters are given in units of J_s. In our case, we have fixed $J_s = 20$, so that the time evolution is performed for $J_s t \in [0, 20]$. This choice is guided by [32], where the evolution of the system is done for $\Delta t = 20$. To build the classifier, we have measured the following Pauli words over the evolved quantum states:

- **One-body correlators:**

$$\langle \sigma_i^z \rangle \quad \text{for } i = 0, \dots, 9$$

- **Two-body correlators:**

$$\langle \sigma_i^z \sigma_{i+r}^z \rangle \quad \text{for } i = 0, \dots, 9 - r, \quad r = 1, \dots, 9$$

- **Three-body correlators:**
  - Consecutive:

$$\langle \sigma_i^z \sigma_{i+1}^z \sigma_{i+2}^z \rangle$$

  - Non-consecutive:

$$\langle \sigma_i^z \sigma_{i+\delta_1}^z \sigma_{i+\delta_2}^z \rangle \quad \text{for } \delta_1, \delta_2 \in \{1, 2, \dots, 8\}$$

- **Four or more-body correlators:**

$$\langle \sigma_i^z \sigma_{i+1}^z \cdots \sigma_{i+\ell}^z \rangle \quad \text{for } \ell = 3, \dots, 9$$

In total, the number of variables N_V used to train a logistic regression as a classifier is $N_V = 184$. In other studies, for example [32], the whole evolved quantum state was used as input variable ($N_V = 1024$, as they use ten qubits), or [10], where different models using from $N = 10$ to $N = 50$ qubits, and measuring one- and two-body correlators are studied. We have considered longer correlations to include the longer-range dynamics in our Hamiltonian. Also, it is important to notice that, since all the measured observables are commuting operators, only one physical experiment is necessary. Using sets of commuting operators would require carrying out different

experiments in parallel. The obtained results with TDVP are compared with ED when possible. Processing the considered training and test data sets with ED would require an unaffordable computational cost with the current capabilities.

Phase zone	h	W
I	20	2000
II	2	100
III	0.2	0.2
IV	2000	0.2
IV alternative	20	0.2
I to II	2	200
II to III	2	10
III to IV	10	2
IV to I	200	200

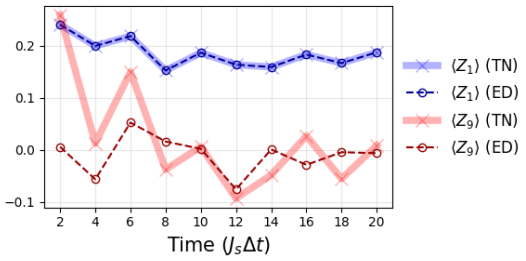
Table 1: Hamiltonian parameters h y W used to perform the study in the different zones in the phase diagram.

5.1 TN simulation versus Exact Diagonalization

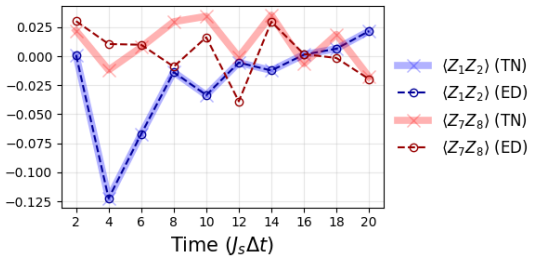
In this part, we present the results of simulating the system using tensor networks compared to exact diagonalization. The two-site Time-Dependent Variational Principle (TDVP) method was employed. Although this method is more computationally demanding, it generally offers improved accuracy. We report the results for one-body, two-body, three-body, and four-or-more-body correlators. To assess the performance of the TN method across different regimes, we selected three representative regions of the phase diagram: one where the method performs well, one where it performs poorly, and one with intermediate performance. For each region, and each class of correlators (1-, 2-, 3-, and 4-or-more-body), we show both a successful and a failed case, if possible. There are cases where performance is always good and others where performance is poor. This allows us to systematically identify and analyze the strengths and limitations of the method.

Firstly, we display the results of zone II. This is a localization area. As can be seen in figure 9, the agreement between the TN method and exact diagonalization is significantly better for observables involving qubits up to and slightly beyond the midpoint of the spin network. This is natural in the context of TN time evolution methods. On the one hand, these methods work better to approximate the evolution of short-range interactions, explaining why the results for the first correlators are better than the worse correlators. In our situation, we have an all-to-all interacting spin network. On the other hand, TDVP methods are variational methods and thus can get stuck in local minima. These local minima could be understood as a very good solution for the first qubits and a poorer one for the last ones. Also, the optimization is carried out in "sweeps", starting and ending in the first qubit. Thus, between the time reference of the results and the time reference of the last qubits update, there is a larger amount of time. Four- and more-body correlations display a high resemblance. Therefore, the general dynamics involving several different qubits seem to be well captured. Finally, it is necessary to notice that, although the different correlator values might not be exact, their temporal trends are quite similar. This phenomenon is also pointed out in [10] for one-site TDVP and correlators involving the first qubits. In that situation, yet, the explanation

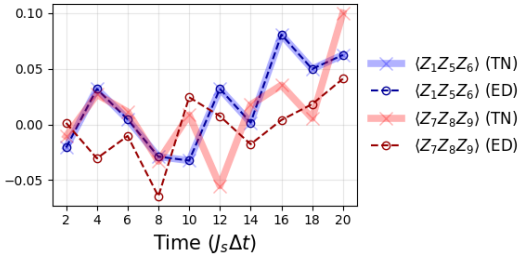
for the errors could be the lack of expressivity of the fixed bond dimension MPS for the time evolution.



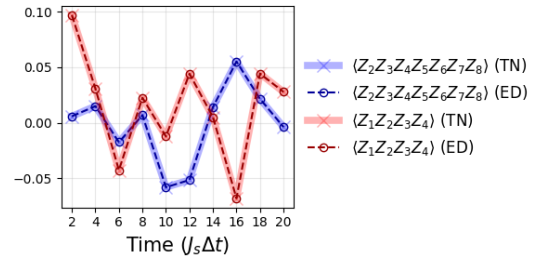
(a) Expected values for the one-body correlators at the beginning of the spin chain ($\langle Z_1 \rangle$) and the end of the spin chain ($\langle Z_9 \rangle$).



(b) Expected values for the two-body correlators at the beginning of the spin chain ($\langle Z_1 Z_2 \rangle$) and at the end of the spin chain ($\langle Z_7 Z_8 \rangle$).



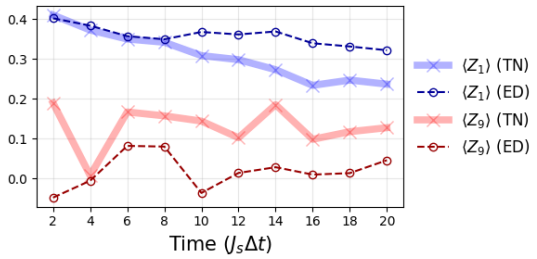
(c) Expected values for the three-body correlators including qubits at the beginning and the middle of the spin chain ($\langle Z_1 Z_5 Z_6 \rangle$) and at the end of the spin chain ($\langle Z_7 Z_8 Z_9 \rangle$).



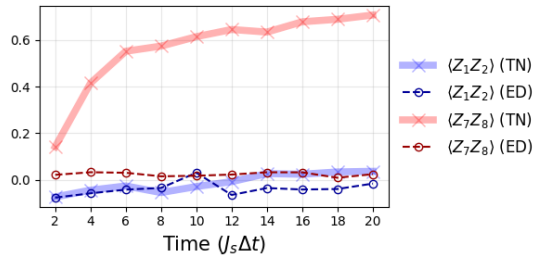
(d) Four-body (and more) correlators in the spin chain.

Figure 9: Comparison between TN and exact diagonalization (ED) methods for different body correlators in zone II. The dashed line corresponds to the exact diagonalization method, whereas the continuous line corresponds to the values obtained using the TN time evolution two-site TDVP.

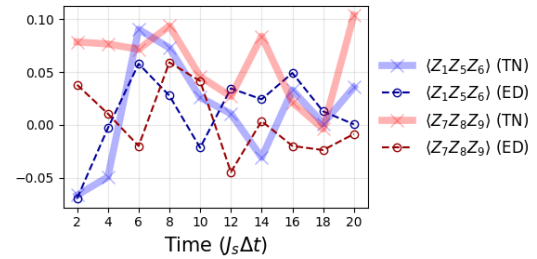
Next, we show the same results for zone I to II , which is a phase transition between a localization (I) and an ergodicity (II) area. At this point, the TN method performed worse with respect to exact diagonalization. In figure 10, we can see that the agreement between TN and ED methods for one- and two-body correlators for the first qubits in the spin chain is high for short evolution times. As we try to approximate longer evolution times, results start to diverge, although the temporal trends are respected. For results involving the last qubits in the spin chain, the results obtained via TN time evolution and ED are quite different. However, during certain times, the temporal trends look alike. For three- and four- or more-body correlations, results are completely different, and we can only find similarity in the temporal series during certain times. The main hypothesis behind the mismatch between the two methods is the presence of large values in the Hamiltonian parameters. Having fixed $J_s = 20$ means that navigating through the phase map 8 requires high values of W and h. At this point, we are dealing with $W = 2 \cdot 10^2$. As the TDVP requires a Lie-Trotter decomposition, high values of parameters in the Hamiltonian can significantly degrade the accuracy of the results. The accuracy of the TDVP method can be increased by using smaller time steps in the evolution. However, to explore the same evolution time, thus, we ought to either extend the simulation or increase the value of J_s. However, the computational cost of the simulation is already high, and increasing the value of J_s can lead to the same effect over the Lie-Trotter decomposition.



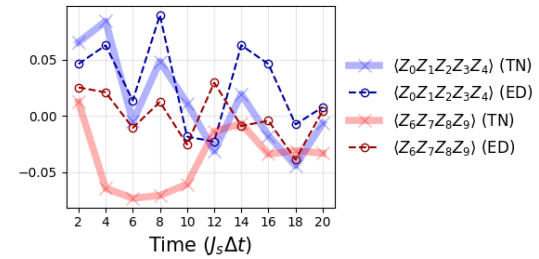
(a) Expected values for the one-body correlators $\langle Z_1 \rangle$ and $\langle Z_9 \rangle$.



(b) Expected values for the two-body correlators $\langle Z_1 Z_2 \rangle$ and $\langle Z_7 Z_8 \rangle$.



(c) Expected values for the three-body correlators $\langle Z_1 Z_5 Z_6 \rangle$ and $\langle Z_7 Z_8 Z_9 \rangle$.

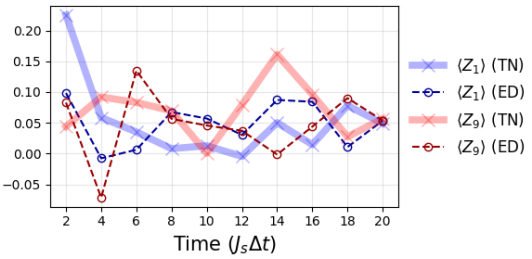


(d) Some examples of five-body correlators in the spin chain.

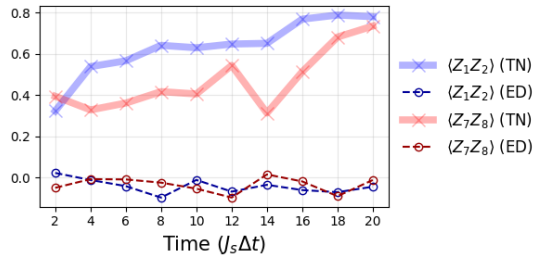
Figure 10: Comparison between TN and exact diagonalization (ED) methods for different body correlators in the transition between zone I and zone II.

For four- or more-body correlators, results are no longer as valid as in zone II. Although we are including several qubit dynamics, the general behaviour is not captured as precisely as in zone II.

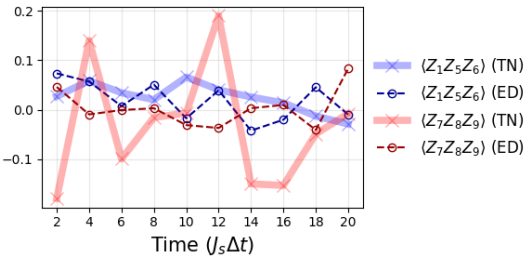
To further explore the performance of the TN time evolution method with respect to exact diagonalization, we now present the results for two points in zone IV (points IV and IV alternative). As shown in 11, the agreement between the two methods is none. The hypothesis is again that the high values in the parameters of the Hamiltonian. For this point, we are considering $h = 2 \cdot 10^3$. Again, this can compromise the Lie-Trotter decomposition in TDVP, leading to a divergent solution. To check this hypothesis, we have tried another point in zone IV, with lower values for the Hamiltonian parameters: IV alternative. Results displayed in figure 12 show a similar agreement as the results in 9. In both situations, the TN time evolution method seems to capture perfectly the individual behaviour of the qubits up to and slightly beyond the midpoint of the spin chain, and worse, the dynamics of the last qubits in the chain. However, the temporal trends of the correlators and the general dynamics are well represented with TN.



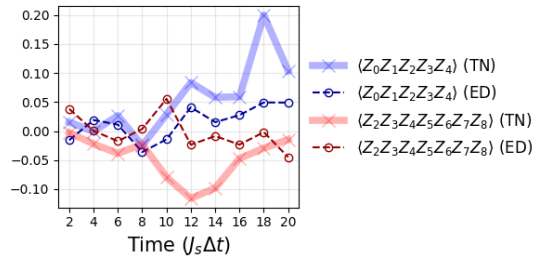
(a) Expected values for the one-body correlators $\langle Z_1 \rangle$ and $\langle Z_9 \rangle$.



(b) Expected values for the two-body correlators $\langle Z_1 Z_2 \rangle$ and $\langle Z_7 Z_8 \rangle$.



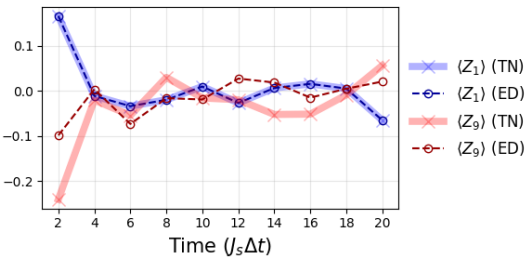
(c) Expected values for the three-body correlators $\langle Z_1 Z_5 Z_6 \rangle$ and $\langle Z_7 Z_8 Z_9 \rangle$.



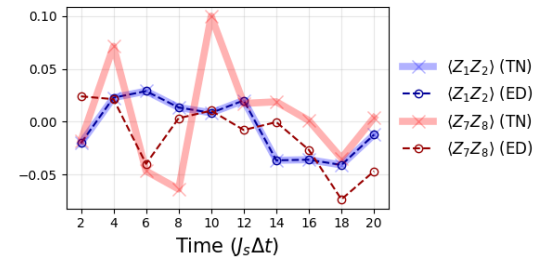
(d) Some examples of five-body correlators in the spin chain.

Figure 11: Comparison between TN and exact diagonalization (ED) methods for different body correlators in zone IV.

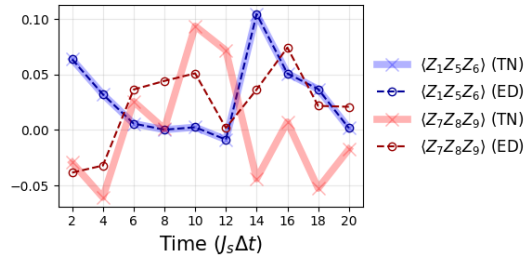
From these two points, we can conclude that there is a clear effect of the particular values of the Hamiltonian on the accuracy of TDVP. Probably, a fine-tuning of the hyperparameters of TDVP and a reduced step for the time evolution could lead to a complete solution.



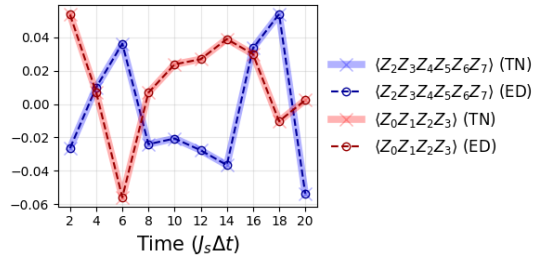
(a) Expected values for the one-body correlators $\langle Z_1 \rangle$ and $\langle Z_9 \rangle$.



(b) Expected values for the two-body correlators $\langle Z_1 Z_2 \rangle$ and $\langle Z_7 Z_8 \rangle$.



(c) Expected values for the three-body correlators $\langle Z_1 Z_5 Z_6 \rangle$ and $\langle Z_7 Z_8 Z_9 \rangle$.



(d) Some examples of five-body correlators in the spin chain.

Figure 12: Comparison between TN and exact diagonalization (ED) methods for different body correlators in zone IV, with lower values of h and W.

The two methods present good agreement in points II, III, IV (alternative), III to II, III to IV, II to III, and I to II. The discrepancies appear in points I, IV, and IV to I. The condition of convergence for values $h, W \lesssim 10^2$ is satisfied in these situations.

5.2 Time Evolution of the performance

In the following subsection, we present the results of training different models for each selected point in the phase map 1. For each point, we have encoded the images (10000 for training and 5000 for test [28]) into quantum states and evolved up to $J_s\Delta t = 20$, in steps of $J_s\Delta t = 2$ using two-site TDVP. Thus, we can evaluate the time evolution of the performance of the models built in the different zones in the phase diagram. The number of qubits used is $N = 10$. Compared with [10], we obtain a similar performance to their results using a Rydberg Hamiltonian made up by $N = 12$ qubits, and evolving with two-site TDVP. Also, their results using one-site TDVP, for $N = 10$ qubits and only 1000 test images, are similar to our results, which consider a test set five times larger. Considering these differences, both results look compatible.
For this analysis, the comparison with ED method could not be performed due to computational costs, as already mentioned.

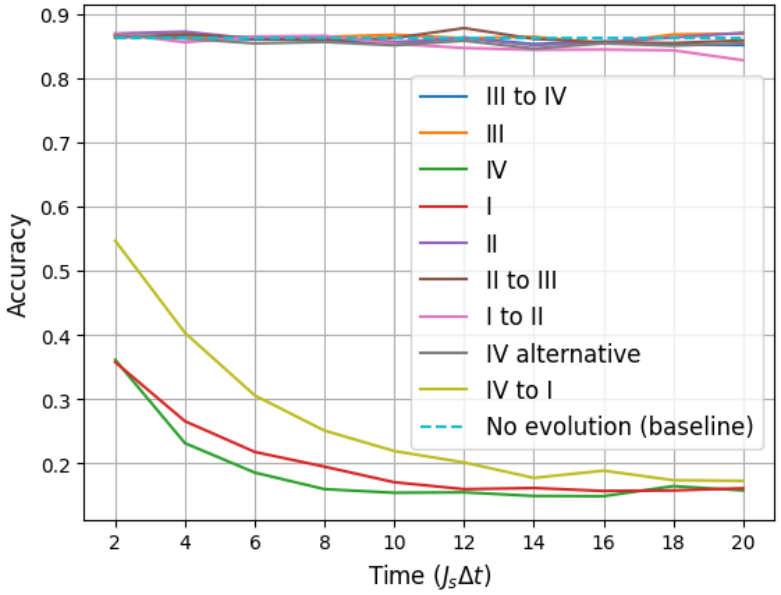


Figure 13: Accuracy of all the different models trained with time evolutions with $J_s\Delta t \in [2, 20]$, for the different points chosen from the phase diagram.

The results displayed in 13 show three divergent models. These, however, correspond to the zones where TDVP and ED methods are not consistent. In fact, once again, two different points selected from IV present different results. Therefore, this divergence can be attributed to numerical error in the analysis. To better examine the performance over time of the other models, we display the convergent ones in 14. First, we can notice that the accuracy of the model trained with a Hamiltonian in the zone I to II is degraded from $J_s\Delta t = 8$. Indeed, we showed in the previous section 5.1 that the simulations with TN and ED matched during a certain time, and then started to diverge. Therefore, the idea that the poor performance presented here has to do with the numerical

simulation seems consistent.

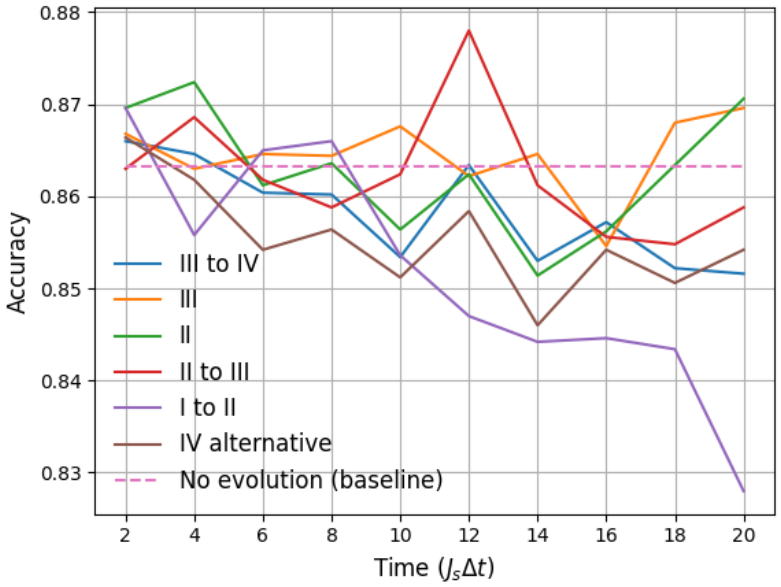


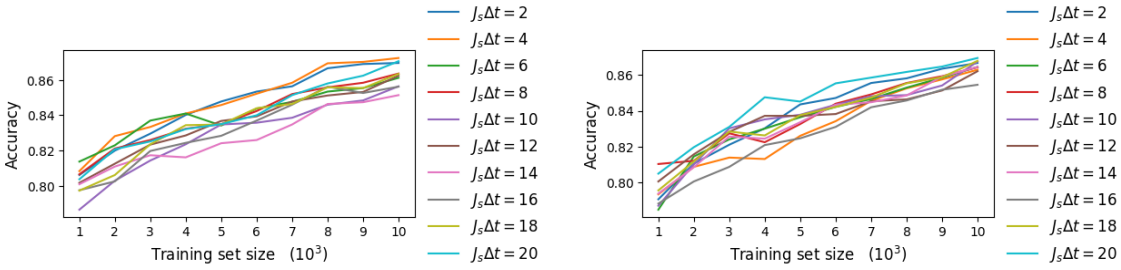
Figure 14: Accuracy of the different convergent models trained with time evolutions with $J_s\Delta t \in [2, 20]$, for the different points chosen from the phase diagram.

First, we can notice that all the models perform similarly, and that the accuracy is almost constant in time, only with minor variations due to statistical fluctuations. Therefore, we can conclude that the performance does not depend on time or on the point in the phase diagram. Looking at 14, we observe that all the models approximate the accuracy of the baseline model. This model was trained encoding the images into quantum states and measuring the mentioned observables without any time evolution. Thus, the data encoding plus measurement process is enough to find an appropriate vector space where the different classes are separable. Following [15] and [2], this behavior is expected: all supervised QML models can be understood as kernel methods, and their expressive power is largely determined by the choice of data encoding. In our case, this situation highlights that the main contribution to the model's performance comes from the encoding scheme itself. This phenomenon suggests potential applications beyond classification tasks. Future research could explore how temporal evolution affects data correlations and investigate the possibility of reconstructing input images after encoding and time evolution, in order to assess the practical impact of the process. Notably, under certain conditions, such as when the evolution scrambles information in a non-invertible or computationally hard-to-invert manner, this approach could function as a quantum data obfuscation mechanism. Related ideas are being discussed (for example, [33]).

Finally, there does not appear to be an influence of the dynamics of the quantum system over the performance of the corresponding model at this level. Also, the performance of the model is acceptable considering that only a sixth of the total data set was used, and accuracies near to 90 % are achieved.

5.3 Learning speed for the different models

In this study, only the convergent models have been considered, and an analysis of the accuracy as a function of the training dataset size has been performed. The results, shown in figure 15, reveal no significant performance differences across models, regardless of the system's evolution time. This observation suggests that the dynamics of the system do not play a crucial role in the learning capacity of the model. Rather, these findings support the hypothesis that the essential work of preparing the data for successful learning is primarily carried out by the data encoding and the measurement process. This interpretation aligns with the results discussed in the previous section, where the performance of the baseline model already displayed decent accuracy levels, indicating that expressivity and learning capacity are largely determined at the encoding stage.



(a) Accuracy versus training set size in point II.

(b) Accuracy versus training set size in point III.

(c) Accuracy versus training set size in point IV.

(d) Accuracy versus training set size in point I to II.

(e) Accuracy versus training set size in point II to III.

(f) Accuracy versus training set size in point III to IV.

Figure 15: Plots for accuracy versus training set size for the convergent models. All the evolution times are considered for the zones.

5.4 Tensor Networks and Quantum Reservoir Computing

Simulating a QRC framework with TN presents some additional challenges: the TN technique needed to represent the quantum state and the time evolution method. These two conditions considerably increase the difficulty of using TN methods for QRC compared with QELM. Throughout this section, we will explain some of the steps towards using TN for QRC. Most of what is shown here is still a work in progress, and thus, there will be no quantitative support, but only a qualitative discussion.

First, to fulfill the contractive map condition mentioned in 2.2, the quantum states of the reservoir need to be mixed. As a consequence, MPS structures are no longer an option to represent the state of the substrate. Two natural options arise. On the one hand, we have MPDO [23]. In MPDOs, the quantum state is purified using an ancillary Hilbert space to find a pure representation of it. That is, if our given mixed quantum state is $\rho \in \mathcal{H}_{\text{original}}$, we find a pure state $|\psi\rangle \in \mathcal{H}_{\text{original}} \otimes \mathcal{H}_{\text{ancilla}}$ such that $\text{Tr}_{\text{ancilla}}[|\psi\rangle \langle\psi|] = \rho$. This pure quantum state has physical indices, corresponding to the original Hilbert space coordinates, and auxiliary indices, corresponding to the ancillary Hilbert space coordinates. Taking that into account, we can build an MPS using this quantum state, with the same number of tensors as the number of qubits in the original state. These tensors will have physical legs and auxiliary legs. Operations will only be carried out over the physical legs.

This structure has several advantages: first, it allows for operating with it just like an MPS, and thus time evolution methods, like TDVP, can be used normally. Furthermore, this TN is positive by construction and will always represent physical states. However, the main disadvantage is that the updating protocol needed for QRC cannot be executed easily. That is, the data encoding in each time step would require computing the density matrix of the quantum state represented by the MPDO, tracing out the first qubit state, and inserting via tensor product the new first qubit state. However, this would leave a density matrix for the whole purified system in an MPO structure, not an MPDO, so it would be necessary to find the original quantum state by tracing out the ancillary system and purifying it again. This, obviously, is highly inefficient computationally.

On the other hand, an alternative is to use an MPO to represent the mixed state directly. Although MPOs might end up representing non-physical states, this can be somehow fixed during time evolution with a fine-tuning of the method's parameters. Without that inconvenience, MPOs arise as a natural structure for encoding mixed states as TN. The QRC protocol would be easy to implement, as the necessary tools (partial traces, tensor products) are natural in this formalism. Nevertheless, the main problem comes from time evolution. There are no clear methods for time evolution, other than direct approximations of the time evolution operator acting over the MPO [34]. These methods, however, require a big effort to ensure the unitarity of the evolution, and are computationally inefficient when applied over MPOs. Another option is described in [35]. It consists of using an MPO and reshaping it as an MPS, doubling the local Hilbert spaces for time evolution, and splitting back for the injection protocol. Therefore, we can consider an updated Hamiltonian acting over the MPS representation, with doubled local Hilbert space dimensions for time evolution. As shown in [35], however, TDVP does not naturally conserve energy with this scheme, and a correction over the MPS representation has to be made before and after the time evolution

to conserve energy. This correction introduces a hyperparameter in the evolution. For arbitrary operators, no fine-tuning seems to be necessary. In our situation, on the contrary, we have found the time evolution to be extremely sensitive concerning this correction hyperparameter, although heuristic rules can be found to ensure the unitarity of the time evolution and the conservation of energy.

The last problem is that TDVP algorithms find good representations for the quantum states in terms of observables: energy, spin, etcetera. However, the QRC framework, as detailed in this project, requires a local update on the TN. This local update requires a high fidelity between the TN coordinate representation and the quantum state coordinate representation. If this is not the case, the input data injection drives apart the TN from the quantum state, and thus, time evolution from that moment on is different for each of them. The mentioned TDVP algorithms, when applied over mixed states, fail indeed to find accurate coordinate representations for the quantum state. Therefore, they do not seem to be an option for this task.

Next to this, there are some next steps to consider. First, consider other time evolution methods. This is not trivial, since most of them require local interactions in the Hamiltonian (nearest neighbours), and this is not our situation. Others cannot be adapted to use the trick MPO $\longleftrightarrow$ MPS transformations. Second, consider also other QRC frameworks [36], and implement TN methods for open quantum systems [37]. As already mentioned, this is work in progress.

6 Conclusions

This work explores the potential of a quantum many-body system, a spin chain, as reservoir for machine learning tasks (MNIST image classification), focusing on the simulation of such systems through tensor network (TN) methods and their comparison with exact diagonalization (ED), and the evaluation of the performance of the obtained models. The conclusions drawn from the different analyses are the following:

1. Simulation accuracy with Tensor Networks: the comparison between TN time evolution (using two-site TDVP) and ED reveals that the TN method performs well when the Hamiltonian parameters remain below a certain threshold (specifically, $\lesssim 10^2$). In these regimes, the TN evolution is capable of quite accurately capturing both the values and trends of one- and many-body correlators. However, when the parameters increase, discrepancies emerge, likely due to numerical instabilities introduced by the Lie-Trotter decomposition under high Hamiltonian magnitudes. This highlights a fundamental limitation of the TDVP method in highly chaotic or strongly interacting regimes. Moreover, it was observed that the TN method struggles to accurately represent the dynamics of qubits located towards the end of the spin chain.

2. Performance of Quantum Extreme Learning Machines: across all explored regions in the phase diagram, models trained with different Hamiltonians for time evolutions achieve similar classification accuracies to the baseline model—where no time evolution is applied. This observation supports the theoretical claim [15] [2] that the expressivity of supervised quantum machine learning models largely depends on the data encoding scheme. In our setup, amplitude encoding followed by measurement already provides a sufficiently rich feature space, and the time

evolution contributes little to no additional discriminatory power for the later classification model The classification accuracy remains stable across different dynamical regimes (localized and ergodic) for the convergent models. This reinforces the conclusion that the data encoding plus the measurement are the sufficient non-linear process required to separate the classes.

3. Learning speed and sample efficiency: analysis of accuracy versus training set size shows no clear variation in learning speed across different models or evolution times. This further supports the idea that the critical component lies in the data encoding and not in the dynamics of the reservoir.

4. Preliminary work on Quantum Reservoir Computing (QRC): a qualitative discussion on using TN methods to simulate QRC architectures reveals several technical challenges, particularly related to representing mixed states and achieving faithful time evolution. Though promising strategies (e.g., using MPDOs or reshaped MPOs) are discussed, more robust evolution schemes and fidelity-preserving protocols are required before a functional TN-based QRC implementation can be achieved.

References

[1] M. Cerezo et al. "Variational quantum algorithms". In: *Nature Rev. Phys.* 3.9 (2021), pp. 625–644. DOI: 10.1038/s42254-021-00348-9. arXiv: 2012.09265 [quant-ph].

[2] Maria Schuld. "Supervised quantum machine learning models are kernel methods". In: (Jan. 2021). arXiv: 2101.11020 [quant-ph].

[3] Michael Ragone et al. "A Lie algebraic theory of barren plateaus for deep parameterized quantum circuits". In: *Nature Communications* 15.1 (Aug. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-49909-3. URL: http://dx.doi.org/10.1038/s41467-024-49909-3.

[4] Pere Mujal et al. "Opportunities in quantum reservoir computing and extreme learning machines". In: *Advanced Quantum Technologies* 4.8 (2021), p. 2100027.

[5] Gouhei Tanaka et al. "Recent advances in physical reservoir computing: A review". In: *Neural Networks* 115 (July 2019), pp. 100–123. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2019.03.005. URL: http://dx.doi.org/10.1016/j.neunet.2019.03.005.

[6] Keisuke Fujii et al. "Harnessing Disordered-Ensemble Quantum Dynamics for Machine Learning". In: *Physical Review Applied* 8.2 (Aug. 2017). ISSN: 2331-7019. DOI: 10.1103/physrevapplied.8.024030. URL: http://dx.doi.org/10.1103/PhysRevApplied.8.024030.

[7] Daniel Brunner et al. *Photonic Reservoir Computing: Optical Recurrent Neural Networks*. Walter de Gruyter GmbH & Co KG, 2019. ISBN: 9783110556478.

[8] Johannes Nokkala et al. "Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing". In: *Communications Physics* 4.1 (Mar. 2021). ISSN: 2399-3650. DOI: 10.1038/s42005-021-00556-w. URL: http://dx.doi.org/10.1038/s42005-021-00556-w.

[9] Román Orús. "Tensor networks for complex quantum systems". In: *APS Physics* 1 (2019), pp. 538–550. DOI: 10.1038/s42254-019-0086-7. arXiv: 1812.04011 [cond-mat.str-el].

[10] Payal D. Solanki et al. "Exploring Tensor Network Algorithms as a Quantum-Inspired Method for Quantum Extreme Learning Machine". In: (Mar. 2025). arXiv: 2503.05535 [quant-ph].

[11] Marvin Minsky et al. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press, 1969.

[12] Guang-Bin Huang et al. "Extreme learning machine: Theory and applications". In: *Neurocomputing* 70.1–3 (2006), pp. 489–501. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2005.12.126. URL: https://www.sciencedirect.com/science/article/pii/S0925231206000385.

[13] Luca Innocenti et al. "Potential and limitations of quantum extreme learning machines". In: *Communications Physics* 6.1 (2023), p. 118. DOI: 10.1038/s42005-023-01233-6. URL: https://www.nature.com/articles/s42005-023-01233-6.

[14] Weijie Xiong et al. "On fundamental aspects of quantum extreme learning machines". In: *Quantum Machine Intelligence* 7.1 (2025), p. 20. DOI: 10.1007/s42484-025-00239-7. arXiv: 2312.15124 [quant-ph].

[15] Maria Schuld et al. “Effect of data encoding on the expressive power of variational quantum-machine-learning models”. In: *Physical Review A* 103.3 (2021), p. 032430. DOI: 10.1103/PhysRevA.103.032430. URL: https://journals.aps.org/pra/abstract/10.1103/PhysRevA.103.032430.

[16] Minati Rath et al. “Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy”. In: *EPJ Quant. Technol.* 11.1 (2024), p. 72. DOI: 10.1140/epjqt/s40507-024-00285-3. arXiv: 2311.10375 [quant-ph].

[17] Mantas Lukoševičius et al. “Reservoir computing approaches to recurrent neural network training”. In: *Computer Science Review* 3.3 (2009), pp. 127–149. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2009.03.005. URL: https://www.sciencedirect.com/science/article/pii/S1574013709000173.

[18] Zoran Konkoli. “Unconventional Approaches to Computation: An Overview”. In: *Advances in Unconventional Computing*. Ed. by Andrew Adamatzky. Springer, 2017, pp. 573–607.

[19] Herbert Jaeger. *The Echo State Approach to Analysing and Training Recurrent Neural Networks*. Tech. rep. GMD Report 148. German National Research Center for Information Technology, 2001.

[20] Wolfgang Maass et al. “Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations”. In: *Neural Computation* 14.11 (2002), pp. 2531–2560.

[21] Lyudmila Grigoryeva et al. “Echo state networks are universal”. In: *CoRR* abs/1806.00797 (2018). arXiv: 1806.00797. URL: http://arxiv.org/abs/1806.00797.

[22] M. Zwolak et al. “Mixed-State Dynamics in One-Dimensional Quantum Lattice Systems: A Time-Dependent Superoperator Renormalization Algorithm”. In: *Physical Review Letters* 93.20 (Nov. 2004), p. 207205. DOI: 10.1103/PhysRevLett.93.207205.

[23] F. Verstraete et al. “Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems”. In: *Physical Review Letters* 93.20 (Nov. 2004), p. 207204. DOI: 10.1103/PhysRevLett.93.207204.

[24] Jutho Haegeman et al. “Time-Dependent Variational Principle for Quantum Lattices”. In: *Phys. Rev. Lett.* 107 (2011), p. 070601. DOI: 10.1103/PhysRevLett.107.070601. arXiv: 1103.0936 [cond-mat.str-el].

[25] Jutho Haegeman et al. “Unifying time evolution and optimization with matrix product states”. In: *Phys. Rev. B* 94 (16 Oct. 2016), p. 165116. DOI: 10.1103/PhysRevB.94.165116. URL: https://link.aps.org/doi/10.1103/PhysRevB.94.165116.

[26] Helge Holden et al. *Splitting Methods for Partial Differential Equations with Rough Solutions: Analysis and MATLAB Programs*. European Mathematical Society, Apr. 2010. ISBN: 978-3-03719-078-4. DOI: 10.4171/078.

[27] Mario Collura et al. *Tensor Network Techniques for Quantum Computation*. SISSA Medialab S.r.l., 2024. ISBN: 978-88-98587-04-9. DOI: 10.22323/9788898587049. arXiv: 2503.04423 [quant-ph].

[28] Yann LeCun et al. *MNIST Handwritten Digit Database*. http://yann.lecun.com/exdb/mnist/. AT&T Labs [Online]. 2010.

[29] Rodrigo Martínez-Peña et al. "Dynamical Phase Transitions in Quantum Reservoir Computing". In: *Phys. Rev. Lett.* 127 (10 Aug. 2021), p. 100502. DOI: 10.1103/PhysRevLett.127.100502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.100502.

[30] J. R. Johansson et al. "QuTiP: An open-source Python framework for the dynamics of open quantum systems". In: *Computer Physics Communications* 183.8 (2012), pp. 1760–1772. DOI: 10.1016/j.cpc.2012.02.021. URL: http://qutip.org.

[31] Johannes Hauschild et al. "Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy)". In: *SciPost Physics* 5.5 (2018), p. 47. DOI: 10.21468/SciPostPhys.5.5.047. URL: https://scipost.org/10.21468/SciPostPhys.5.5.047.

[32] A. De Lorenzis et al. "Harnessing quantum extreme learning machines for image classification". In: *Physical Review Applied* 23.4 (Apr. 2025). ISSN: 2331-7019. DOI: 10.1103/physrevapplied.23.044024. URL: http://dx.doi.org/10.1103/PhysRevApplied.23.044024.

[33] James Bartusek et al. *Quantum State Obfuscation from Classical Oracles*. 2024. arXiv: 2401.10200 [quant-ph]. URL: https://arxiv.org/abs/2401.10200.

[34] Michael P. Zaletel et al. "Time-evolving a matrix product state with long-ranged interactions". In: *Phys. Rev. B* 91 (16 Apr. 2015), p. 165112. DOI: 10.1103/PhysRevB.91.165112. URL: https://link.aps.org/doi/10.1103/PhysRevB.91.165112.

[35] Christian B. Mendl. *Time evolution of matrix product operators with energy conservation*. 2018. arXiv: 1812.11876 [quant-ph]. URL: https://arxiv.org/abs/1812.11876.

[36] Antonio Sannia et al. "Dissipation as a resource for Quantum Reservoir Computing". In: *Quantum* 8 (2024), p. 1291. DOI: 10.22331/q-2024-03-20-1291. arXiv: 2212.12078 [quant-ph].

[37] Gerald E. Fux et al. "Tensor network simulation of chains of non-Markovian open quantum systems". In: *Phys. Rev. Res.* 5 (3 Aug. 2023), p. 033078. DOI: 10.1103/PhysRevResearch.5.033078. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.5.033078.