Constructing networks of quantum channels for state preparation

Der QUEST-Leibniz-Forschungsschule
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
Dr. rer. nat.

vorgelegte Dissertation von

M. Sc. Dmytro Bondarenko

Leibniz
Universität
Hannover

2021

Possible Machines Inc.

¬ Reinhard F. Werner

Abstract

Entangled possibly mixed states are an essential resource for quantum computation, communication, metrology, and the simulation of many-body systems. It is important to develop and improve preparation protocols for such states.

One possible way to prepare states of interest is to design an open system that evolves only towards the desired states. A Markovian evolution of a quantum system can be generally described by a Lindbladian. Tensor networks provide a framework to construct physically relevant entangled states. In particular, matrix product density operators (MPDOs) form an important variational class of states. MPDOs generalize matrix product states to mixed states, can represent thermal states of local one-dimensional Hamiltonians at sufficiently large temperatures, describe systems that satisfy the area law of entanglement, and form the basis of powerful numerical methods. In this work we develop an algorithm that determines for a given linear subspace of MPDOs whether this subspace can be the stable space of some frustration free k-local Lindbladian and, if so, outputs an appropriate Lindbladian.

We proceed by using machine learning with networks of quantum channels, also known as quantum neural networks (QNNs), to train denoising post-processing devices for quantum sources. First, we show that QNNs can be trained on imperfect devices even when part of the training data is corrupted. Second, we show that QNNs can be trained to extrapolate quantum states to, e.g., lower temperatures. Third, we show how to denoise quantum states in an unsupervised manner. We develop a novel quantum autoencoder that successfully denoises Greenberger-Horne-Zeilinger, W, Dicke, and cluster states subject to spin-flip, dephasing errors, and random unitary noise.

Finally, we develop recurrent QNNs (RQNNs) for denoising that requires memory, such as combating drifts. RQNNs can be thought of as matrix product quantum channels with a quantum algorithm for training and are closely related to MPDOs.

The proposed preparation and denoising protocols can be beneficial for various emergent quantum technologies and are within reach of present-day experiments.

Contents

List of Abbreviations

AKLT	Affleck-Lieb-Kennedy-Tasaki
AdS	anti-de Sitter
AE	autoencoder
BB84	Bennett-Brassard 1984
BEC	Bose-Einstein condensate
CFI	classical Fisher information
CFT	conformal field theory
CHSH	Clauser-Horne-Shimony-Holt
cMPS	continious matrix product state
CNOT	controlled-NOT
CP	completely positive
CPTP	completely positive trace-preserving
CSS	Calderbank-Shor-Steane
DI	device independent
DMRG	density matrix renormalization group
EPR	Einstein–Podolsky–Rosen
FF	feed forward
FF	feed forward neural network
FI	Fisher information
GHZ	Greenberger-Horne-Zeilinger
GKP	Gottesman-Kitaev-Preskill
GKSL	Gorini–Kossakowski–Sudarshan–Lindblad
LOCC	local operations and classical communication
LSTM	long short-term memory
ML	machine learning
MPDO	matrix product density operator
MPO	matrix product operator
MPS	matrix product state (states)
NISQ	noisy intermediate-scale quantum
NMR	nuclear magnetic resonance
NN	neural network
NP	non-deterministic polynomial-time
NV	nitrogen-vacancy
PEPS	projected entangled pair stares

POVM	positive operator-valued measurement
QEC	quantum error correction
QEC	quantum error correcting code
QFI	quantum Fisher information
QKD	quantum key distribution
QMA	quantum Merlin Arthur
QML	quantum machine learning
QNN	quantum neural network
ReLU	rectified linear unit
RNN	recurrent neural network
RQNN	recurrent quantum neural network
SQL	standard quantum limit
TP	trace-preserving
TN	tensor networks

1. Overview

Entangled states are an essential resource for quantum computation, communication, metrology, and the simulation of many-body systems. If the system of interest is open, it is usually described by a mixed state.

We are interested in the following question: how do we prepare a known, possibly highly entangled and mixed, quantum state ρ? The word "know" in this question can be understood in two different ways. We can say that we know a state if we have its full classical description. Alternatively, we can say that we know a state if we can build a source that produces states that contain information about ρ. This thesis deals with both of these scenarios.
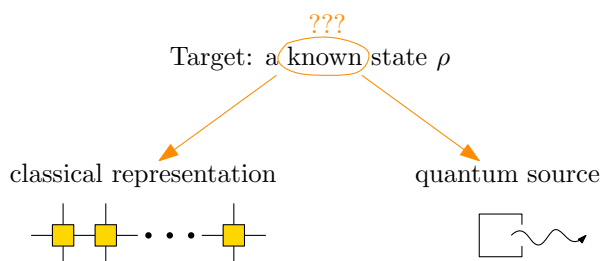
The number of parameters of a generic many-body quantum state grows exponentially with the system size. Thus, it is crucial to identify classes of states that admit an efficient classical description and are physically relevant. Tensor networks provide a framework for constructing physically relevant entangled states. In particular, matrix product states [196] (MPS, see Fig. 1.1) can efficiently describe ground states of one-dimensional gapped local Hamiltonians (originally proven in [257], later with exponentially better parameters in [33]).

It is also important to demand that the state can be prepared via a protocol such that the control effort does not grow too fast with the system size. One way to achieve such a state preparation is dissipative engineering—the design of few-body interactions and dissipation such that the system relaxes into a desired state. It is clear how to dissipatively prepare a given space of MPS [334, 598]. Indeed, for any space of MPS one can construct a so-called *parent* Hamiltonian [196]—a frustration-free local Hamiltonian that has the given MPS as its unique ground state. By adding energy dissipation to each local term one can ensure that the only stable state of the constructed evolution is the desired MPS.

Figure 1.1.: Matrix product states can approximate well ground states of 1d local Hamiltonians and have a parent Hamiltonian.

A direct generalization of MPS for mixed states are matrix product density operators (MP-DOs) [597, 657]. MPDOs can represent thermal states of local Hamiltonians [287, 122]. However, it has not been previously clear if a given space of MPDOs can be dissipatively prepared using k-local operators (see Fig. 1.2). In this work we develop an algorithm that constructs k-local parent Lindbladians. That is, the algorithm either outputs a k-local frustration-free Lindbladian that has stable space consisting only of a given space of MPDOs or proves that such a Lindbladian does not exist. This construction is based on the connection between fixed spaces of quantum channels and stable spaces of Lindbladians and is presented in Chapter 3.

$$\text{MPDO} \equiv \quad \boxed{} \, \boxed{} \cdots \boxed{} \quad \text{¿}\Longrightarrow\text{? frustration-free local } \mathcal{L}$$

Figure 1.2.: Does a given space of matrix product density operators have a parent Lindbladian?

We proceed to study how we can improve noisy quantum sources of states. Virtually every experimental preparation of a quantum state introduces noise. Usually, it is hard to design a denoising protocol. First, one has to identify and characterize all noise sources. Second, one has to invent a protocol that corrects the noise without affecting any relevant features of the quantum state. Machine learning can automate this task.

We simulate a source to produce a data set and use it to train a post-processing device. We train networks of quantum channels, also known as quantum neural networks (QNNs) [56], to perform denoising (see Fig. 1.3). QNNs can be implemented and trained on a quantum computer and deal with highly entangled states.
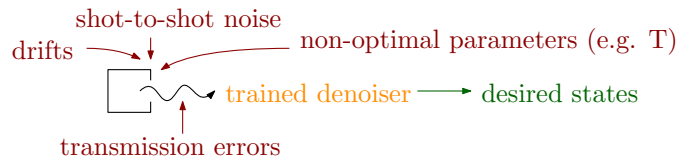
Figure 1.3.: Trained post-processor improves a noisy source.

In Chapter 4 we deal with noise that feed-forward QNNs can mitigate. We do **not** assume access to a noiseless supervising source.

First, we verify that this is a viable approach for not perfect devices by numerically studying if QNNs can be trained on noisy devices with noisy data [56, 516].

Second, we work with sources that produce parametrized states in an undesired parameter region. We use QNNs trained on the available parameter interval to extrapolate quantum states to more desired parameter values, see Fig. 1.4. We concentrate on the case of temperature for the transverse-field Ising model, even though the techniques discussed can be used to other parameters and systems of interest [179].
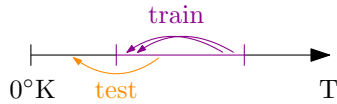
Figure 1.4.: We assume that the source can produce states in some temperature interval. We train a QNN to cool states inside this interval. The network learns to extrapolate, i.e. to take input states from the lower-temperature part of the available interval and output colder states outside the training interval.

Third, we combat shot-to-shot noise. As there is often no noiseless reference state to compare with, unsupervised learning is required. Classical data can be efficiently denoised by autoencoders—neural networks with a bottleneck that can be trained without supervision. We develop a novel quantum autoencoder [85] that successfully denoises Greenberger-Horne-Zeilinger, W, Dicke, and cluster states subject to spin-flips, dephasing errors, and random unitary noise.
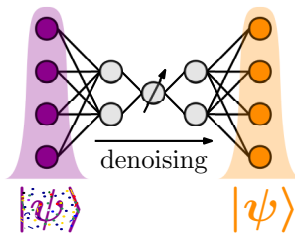
Figure 1.5.

We conclude Chapter 4 with the remark about mitigating noise while the quantum state is being transmitted or stored by learning quantum error correcting codes.

Finally, in Chapter 5 we combine intuition from MPDOs with QNNs to develop a technique to combat drifts. Drift mitigation requires memory, so feed-forward networks are not ideal. We develop an algorithm to train recurrent QNNs (RQNNs, see Fig. 1.6).
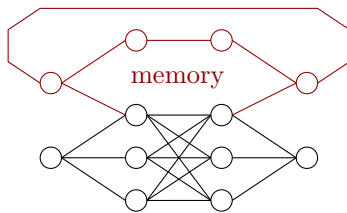
Figure 1.6.: Recurrent networks incorporate memory.

RQNNs can be thought of as matrix product quantum channels with a quantum algorithm for training (see Fig. 1.7) and are closely related to MPDOs studied in Chapter 3. We demonstrate that RQNNs can be used to design optimal low- and high-pass filters for quantum devices.
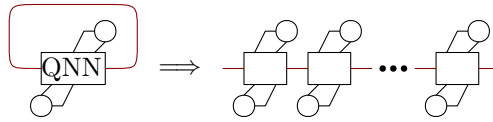
Figure 1.7.: If recurrent networks are trained on sequences of length L, they can be unrolled to a length-L matrix product quantum channel. Lower circles represent input qubits and upper circles—output qubits.

We conclude in Chapter 6 that quantum channels can be organised in networks or optimized for tensor network states. Such networks of quantum channels is a versatile tool for state preparation that is robust against noise. We follow by a discussion of possible applications and open problems.

The preliminary material needed to understand the results in this thesis is contained in Chapter 2.

2. Preliminary material

This chapter contains the ingredients necessary for understanding the results of this thesis.

Notation 2.0.1. *Given m 2-level systems, we denote*

$$\sigma_j^i = \mathbb{1}^{\otimes(j-1)} \otimes \sigma^i \otimes \mathbb{1}^{\otimes(m-j)}, \quad i \in \{x, y, z\}, \tag{2.1}$$

where $\mathbb{1} = |{\uparrow}\rangle\langle{\uparrow}| + |{\downarrow}\rangle\langle{\downarrow}|$ is the identity and $\sigma^x = |{\uparrow}\rangle\langle{\downarrow}| + |{\downarrow}\rangle\langle{\uparrow}|$, $\sigma^y = i|{\uparrow}\rangle\langle{\downarrow}| - i|{\downarrow}\rangle\langle{\uparrow}|$, $\sigma^z = |{\uparrow}\rangle\langle{\uparrow}| - |{\downarrow}\rangle\langle{\downarrow}|$ the Pauli operators for a single qubit.
We also associate $|0\rangle \leftrightarrow |{\downarrow}\rangle$ and $|1\rangle \leftrightarrow |{\uparrow}\rangle$.

2.1. Quantum channels

In this section we will define the most general maps between quantum states—*quantum channels*. These are the main protagonists of this thesis—we discuss network-based methods for the construction of an appropriate quantum channel.

For a more in-depth introduction to quantum channels, see e.g. [73, 639, 303, 496, 424].

The most general quantum state can be described as a density matrix. A density matrix is a generalisation of a probability distribution. As probabilities are real positive numbers that sum up to one, a density matrix is a self-adjoint positive trace one operator.

Notation 2.1.1 (Trace-class)**.** *We will denote the space of linear operators with trace acting from a Hilbert space $\mathcal{H}_1$ to $\mathcal{H}_2$ as $\mathcal{T}(\mathcal{H}_1, \mathcal{H}_2)$. We will denote the space $\mathcal{T}(\mathcal{H}, \mathcal{H})$ as $\mathcal{T}(\mathcal{H})$.*

A map $\varepsilon : \mathcal{T}(\mathcal{H}) \to \mathcal{T}(\mathcal{H})$ should preserve convex combination of states. Indeed, for a probability distribution $\{p_i\}_{i=1}^n$ and density matrices $\{\rho_i \in \mathcal{T}(\mathcal{H})\}_{i=1}^n$ the convex combination $\sum_i p_i \rho_i$ represents that the state is ρ_i with probability p_i

$$\varepsilon\left(\sum_i p_i \rho_i\right) = \sum_i p_i \varepsilon(\rho_i). \tag{2.2}$$

The map ε can be naturally extended to a linear map.

For the image of any density matrix to also be a density matrix, the map ε should be positive—that is, to map positive operators to positive operators

$$\varepsilon \geq 0 \text{ if } \rho \geq 0 \Rightarrow \varepsilon(\rho) \geq 0, \tag{2.3}$$

and ε should be trace preserving (TP)

$$\text{Tr}\left(\varepsilon(\rho)\right) = \text{Tr}\left(\rho\right). \tag{2.4}$$

However, even if ε maps any quantum state to a quantum state, it does not mean that ε can be realised physically. Indeed, it is not practically possible to build a machine that operates on any scale—any conceivable device can have a non-negligible effect only on a small part of the universe. Let us choose some n-dimensional subsystem of the universe that is not affected by the operation under consideration. It makes sense to demand that $\varepsilon \otimes \mathrm{id}_n$ also maps density matrices to density matrices. If $\varepsilon \otimes \mathrm{id}_n$ is positive, the ε is called n-positive. As $\mathcal{H}$ is usually only a tiny subspace of the universe, it is reasonable to demand that ε is n-positive for any n. Such maps are called completely positive (CP)

$$\varepsilon \text{ is completely positive} \Leftrightarrow \varepsilon \otimes \mathrm{id}_n \geq 0 \; \forall n. \tag{2.5}$$

Not every positive map is completely positive. The simplest example is the transpose map for qubits, $\varepsilon_{\mathrm{transpose}} : \mathcal{T}(\mathbb{C}^2) \to \mathcal{T}(\mathbb{C}^2)$, $\varepsilon_{\mathrm{transpose}}(\rho) = \rho^T$. This map is positive, but not completely positive.

Let us collect all the properties discussed so far in this section:

Definition 2.1.2 (Quantum channel). *We call a map ε a channel or, alternatively, a completely positive trace preserving (CPTP) if*

- *ε is linear.*

- *ε is completely poitive: $\varepsilon \otimes \mathrm{id}_n \geq 0 \; \forall n$.*

- *ε is trace preserving: $\mathrm{Tr}\left(\varepsilon(\rho)\right) = \mathrm{Tr}\left(\rho\right).$*

The most general physical operations that act between density matrices are CPTP maps. [1]

2.1.1. Observables

To describe experiments, one needs to describe measurements. Only discussing states is not sufficient, it is also crucial to define things that are being measured—*observables*. A state corresponds to a linear functional mapping an observable A onto finite real numbers—an expectation value $\langle A \rangle$.

Notation 2.1.3 (Bounded operators). *We will denote the algebra of bounded linear operators acting from a Hilbert space $\mathcal{H}_1$ to $\mathcal{H}_2$ as $\mathcal{B}(\mathcal{H}_1, \mathcal{H}_2)$. We will denote the space $\mathcal{B}(\mathcal{H}, \mathcal{H})$ as $\mathcal{B}(\mathcal{H})$.*

The condition that the expectation values should be finite leads to observable $A \in \mathcal{B}(\mathcal{H})$ and the condition that the numbers should be real—to the space of observables being closed under adjoints. It is possible to construct observables by combining other observables. This leads to a demand that the space of observables is closed under multiplication. Finally, it is natural to have a trivial observable.

---

[1] Description of a system may not take some possibilities into account. In classical systems, that corresponds to the sum of considered probabilities being less than 1; in the quantum case, instead of a density matrix our knowledge of a system is described by a positive operator σ such that $\mathrm{Tr}[\sigma] \leq 1$. The corresponding operations are described by CP, trace non-increasing maps.

Definition 2.1.4 (Observables). *An algebra of observables $\mathcal{A}$ is a closed linear subspace of $\mathcal{B}(\mathcal{H})$ that contains an identity $\mathbb{1} \in \mathcal{A}$ and is closed under multiplication $A, B \in \mathcal{A} \Rightarrow AB \in \mathcal{A}$ and adjoints $A \in \mathcal{A} \Rightarrow A^\dagger \in \mathcal{A}$.*

Definition 2.1.5 (Expectation value). *Given a state $\rho \in \mathcal{T}(\mathcal{H})$, the expectation value of an observable $A \in \mathcal{A} \subseteq \mathcal{B}(\mathcal{H})$ is*

$$\mathrm{Tr}(\rho A). \tag{2.6}$$

Instead of describing the evolution as a transformation of states $\rho \to \varepsilon(\rho)$ (Shrödinger picture), one may describe the evolution of observables $A \to \varepsilon^\dagger(A)$ (Heisenberg picture). The expectation values should not depend on the picture, so the consistency condition is

$$\mathrm{Tr}\left[\varepsilon(\rho)A\right] = \mathrm{Tr}\left[\rho\varepsilon^\dagger(A)\right], \tag{2.7}$$

and Equation (2.7) defines mutually *adjoint* or *dual* maps.

Notation 2.1.6. *We will denote by $\mathcal{M}_d$ the algebra of $d \times d$ matrices with complex entries.*

In this thesis we will be mainly interested in the case of finite-dimensional observables $A \in \mathcal{B}(\mathbb{C}^d) \cong \mathcal{M}_d$.

2.1.2. Representations

In this section we will discuss several ways to parametrize and represent quantum channels. More information on this topic can be found in Chapters 1 and 2 of [639] and Chapter 2 of [303].

Theorem 2.1.7 (Kraus representation). *A linear map $\varepsilon^\dagger \in \mathcal{B}(\mathcal{M}_d, \mathcal{M}_q)$ is CP if and only if it admits a representation*

$$\varepsilon^\dagger(A) = \sum_{j=1}^r K_j A K_j^\dagger$$

$$\varepsilon(\rho) = \sum_{j=1}^r K_j^\dagger \rho K_j. \tag{2.8}$$

This representation has the following properties:

1. **Normalization**: *ε is trace-preserving iff $\sum_j K_j^\dagger K_j = \mathbb{1}$.*

2. **Kraus rank**: *The sufficient number of Kraus operators $\{K_j \in \mathcal{B}(\mathbb{C}_d, \mathbb{C}_q)\}_{j=1}^r$ is $r \leq dq$.*

3. **Orthogonality**: *There is always a representation with r Hilbert-Schmidt orthogonal Kraus operators (i.e. $\mathrm{Tr}\left[K_i^\dagger K_j\right] \propto \delta_{ij}$).*

4. **Freedom**: *Two sets of Kraus operators $\{K_j\}_{j=1}^r$ and $\left\{\tilde{K}_l\right\}_{l=1}^{\tilde{r}}$ represent the same map ε iff there is a unitary U such that $K_i = \sum_l U_{jl}\tilde{K}_l$ and the smaller set is padded with zeros.*

It is often convenient to see a quantum channel as an evolution of a part of a closed subsystem. This view can be made rigorous. Let us start with a useful tool.

Proposition 2.1.1 (Schmidt decomposition)**.** *For every vector $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ there exist orthonormal bases $\{|e_i\rangle \in \mathcal{H}_1\}$ and $\{|f_i\rangle \in \mathcal{H}_2\}$ such that*

$$|\psi\rangle = \sum_i^d \sqrt{\lambda_i}|e_i\rangle|f_i\rangle \ \ \text{with} \ \ \lambda_i \geq 0, \ \sum_i \lambda_i = \|\psi\|^2, \tag{2.9}$$

where $d = \min(\dim(\mathcal{H}_1), \dim(\mathcal{H}_2))$. The set $\{\sqrt{\lambda_i}\}_{i=1}^d$ are called Schmidt coefficients.

Corollary 2.1.8 (Purification)**.** *For a density matrix $\rho_1 = \sum_{i=1}^d \lambda_i |e_i\rangle\langle e_i|$ Proposition (2.1.1) provides a purification such that $\rho_1 = \text{Tr}(|\psi\rangle\langle\psi|)$.*

Thus, any mixed state can be regarded as a subsystem of some larger pure system. However, it is important to establish that not only the states, but the dynamics can also be treated as being part of a larger closed system.

Theorem 2.1.9 (Stinespring dilation)**.** *Let $\varepsilon : \mathcal{M}_d \to \mathcal{M}_q$ be a CP map. For every Kraus rank r there is a $V : \mathbb{C}^d \to \mathbb{C}^q \otimes \mathbb{C}^r$ such that*

$$\varepsilon^\dagger(A) = V^\dagger(A \otimes \mathbb{1}_r)V, \quad \forall A \in \mathcal{M}_d. \tag{2.10}$$

ε is TP iff $V^\dagger V = \mathbb{1}_d$ (i.e. V is an isometry).

Proof. This follows from Proposition 2.1.7: let $V := \sum_j K_j \otimes |j\rangle$, where $\{|j\rangle\}$ is an orthonormal basis in $\mathbb{C}^r$ and $\varepsilon^\dagger(A) = \sum_{j=1}^r K_j A K_j^\dagger$. $\qquad\square$

The ancillary system $\mathbb{C}^r$ is usually called *dilation space*. If we move to Schrödinger picture, we get $\varepsilon(\rho) = \text{Tr}_{\mathbb{C}^r}[V\rho V^\dagger]$. As a consequence we get

Theorem 2.1.10 (Open-system representation)**.** *Let $\varepsilon : \mathcal{M}_d \to \mathcal{M}_q$ be a CPTP map. Then there is a unitary $U \in \mathcal{M}_{dq^2}$ and a normalized vector $|\theta\rangle \in \mathbb{C}^q \otimes \mathbb{C}^q$ such that*

$$\varepsilon(\rho) = \text{Tr}_E \left[U(\rho \otimes |\theta\rangle\langle\theta|)U^\dagger \right] \tag{2.11}$$

where E denotes the first two tensor factors in $\mathcal{H} = \mathbb{C}^d \otimes \mathbb{C}^q \otimes \mathbb{C}^q$.

Proof. For Stinespring dilation in Schrödinger picture $\varepsilon = \text{Tr}_{\mathbb{C}^r}[V\rho V^\dagger]$ we can embed V into unitary which acts on tensor product as $V = U(\mathbb{1}_d \otimes |\theta\rangle)$. $\qquad\square$

Let us compare open-system representation with closed-system evolution $\tilde{\rho} \to U\tilde{\rho}U^\dagger$. We can see that $\varepsilon(\rho)$ can be interpreted as evolving a closed enlarged system $\tilde{\rho} = \rho \otimes |\theta\rangle\langle\theta|$ and than restricting to a subsystem $\text{Tr}_E(\tilde{\rho})$.

2.1.3. Fixed spaces

Often it is important to look at states that are not changed under time evolution.

Definition 2.1.11 (Fixed space). *For a quantum channel ε the space of states such that*

$$\varepsilon(\rho) = \rho \tag{2.12}$$

is called a fixed space.

These states naturally arise in long-term dynamics, see e.g. [639, Chapter 6]. Indeed, if the quantum channel is applied many times, the state either changes after each application, or reaches the fixed space. Engineering interactions such that any initial state dissipates to a desired fixed space is a viable strategy for state preparation (see Subsection 2.5.7 and Chapter 3).

It is known, see e.g. [639, Section 6.4], that for a trace-preserving positive map $T : \mathcal{M}_d \to \mathcal{M}_d$ fixed space $\mathcal{F}_T : \{\rho | T(\rho) = \rho\}$ is of the form

$$\mathcal{F}_T = U \left(\mathbb{0}_{d_0} \oplus \bigoplus_{k=1}^{K} (\rho_k \otimes \mathcal{M}_{d_k}) \right) U^\dagger \tag{2.13}$$

where ρ_k are diagonal and positive, U is unitary, and $\mathbb{0}_{d_0}$ is a $d_0 \times d_0$ zero matrix.

2.1.4. Simultaneous block diagonalization and subalgebras of observables

From the Equation 2.13 follows, that any fixed space of a quantum channel has a block diagonal structure, $U \left(\mathbb{0}_{d_0} \oplus \bigoplus_{k=1}^{K} (\rho_k \otimes \mathcal{M}_{d_k}) \right) U^\dagger \in U \left(\mathbb{0}_{d_0} \oplus \bigoplus_{j=1}^{\dim(\rho_k)} \bigoplus_{k=1}^{K} \mathcal{M}_{d_k} \right) U^\dagger$. In Chapter 3 we will need a subroutine that, given a set of states $\{\rho_i\}_{i \in J}$, finds the smallest fixed space $\mathcal{F}$ such that $\forall i \in J \; \rho_i \in \mathcal{F}$. A reasonable starting point in this endeavor is to simultaneously block diagonalize the set $\{\rho_i\}$, for example via the algorithm presented in [413].

Algorithm 1 Pseudocode for the algorithm [413] that finds finest block-diagonalization for a set of self-adjoint matrices.

Input: A set of self-adjoint matrices $\{O_i\}$.
Output: A matrix Q such that $Q^\dagger O_i Q \in \mathbf{Block}\,(\{d_i\})$, where $\mathbf{Block}\,(\{d_i\}) = \mathbb{0}_{d_0} \oplus \bigoplus_i \mathcal{M}_{d_j}$.
With probability 1, dim(**Block**) is minimal.
Optional output (convenient for Section 3.2)**:** The dimensions of blocks $\{d_i\}$, a diagonalized random linear combination Σ of inputs.

1: Take a random linear combination of the inputs, $A = \sum_i x_i O_i$, where $\{x_i\}$ are uniformly distributed on a real interval, e.g. $[0, 1]$.
2: Find a unitary matrix R that diagonalizes A, that is $R^\dagger A R = \Sigma$ and $\Sigma = \mathrm{diag}\,(\alpha_1 \mathbb{1}_{m_1}, \ldots \alpha_k \mathbb{1}_{m_k})$. Let R_i be a matrix consisting of orthonormal vectors corresponding to eigenvalues $\alpha_i \in \mathbb{R}$.
3: Put $K = \{1, \ldots, k\}$, and let $\sim$ be an equivalence relation on K such that

$$i \sim j \Leftrightarrow \exists p :\ R_i^\dagger O_p R_j \neq \mathbb{0}. \tag{2.14}$$

Let $K = \bigcup_i^q K_i$ be the partition of K into equivalence classes with respect to $\sim$. Define matrices $R[K_j] = (R_i | i \in K_j)$
4: Output $Q = (R[K_1], \ldots, R[K_q])$.

There are several available simultaneous block diagonalization algorithms, see e.g. [4, 413, 379, 378, 380, 150, 283, 62, 204, 61, 108]. An advantage of the Algorithm 1 is that it, with probability 1 and in contrast to Jacobi-like algorithms, is exact; moreover, it can be generalised to find the smallest algebras of observables.

As a quantum channel can be treated in different pictures (see Subsection 2.1.1), it should not be surprising that a statement that is similar to Equation 2.13 exists for observables.

Theorem 2.1.12. *Any observable subalgebra $\mathcal{I}$ of $\mathcal{B}(\mathbb{C}^d)$ can be represented as*

$$\mathcal{I} = U \bigoplus_{j=1}^{J} \left(\mathbb{1}_{k_j} \otimes \mathcal{M}_{d_j} \right) U^\dagger \tag{2.15}$$

where $\sum_j k_j d_j = d$.

See e.g. [73, Section and Theorem 2.7] and [639, Section 1.6]) for the proof and discussions.

Experiments are often composed of a discrete set of machines each measuring a corresponding observable O_i. It is useful to understand the power of this measurement apparatus, that is, the smallest algebra that contains the whole set $\{O_i\}$. This algebra, specified e.g. by a set $\left\{ U, \{d_j\}_{j=0}^{J}, \{k_j\}_{j=1}^{J} \right\}$ like in Theorem 2.1.12, can be found using the algorithms developed in [413, 379].

Algorithm 2 Pseudocode for the algorithm [413, 379] that finds the smallest observable algebra that contains a given set of observables.

Input: A set of self-adjoint matrices $\{O_i\}$.
Output: With probability 1, the smallest observable algebra $\mathcal{I}$ such that $\forall i\, O_i \in \mathcal{I}$. With probability 0, an observable algebra containing $\mathcal{I}$.

1: Take a random linear combination of the inputs, $A = \sum_i x_i O_i$, where $\{x_i\}$ are uniformly distributed on a real interval, e.g. $[0, 1]$.
2: Find a unitary matrix R that diagonalizes A, that is $R^\dagger A R = \Sigma$ and $\Sigma = \mathrm{diag}\left(\alpha_1^1 \mathbb{1}_{k_1}, \ldots \alpha_{d_1}^1 \mathbb{1}_{k_1}, \ldots, \alpha_1^J \mathbb{1}_{k_J}, \ldots \alpha_{d_J}^J \mathbb{1}_{k_J}\right)$. Let us denote by R_i^j a unitary matrix such that $\left(R_i^j\right)^\dagger A R_i^j = \alpha_i^j \mathbb{1}_{k_j}$.
3: Let $G_i = (V_i, E_i)$ be a set of directed graphs with vertices $V_i = \{1, \ldots, d_i\}$ and edges $E_i = \left\{(l, m; p) : \left(R_l^i\right)^\dagger O_p R_m^i \neq \mathbb{0}\right\}$. Fix a spanning tree T_i for each G_i.
4: For a tree T_i, let $P_1^i, \ldots P_{d_i}^i$ be the $k_i \times k_i$ matrices that satisfy

$$P_1^i = \mathbb{1}_{k_i}, \tag{2.16}$$

$$P_m^i = \left(\left(R_l^i\right)^\dagger O_p R_m^i\right)^\dagger P_l / c_{plm}, \quad (l, m; p) \in T^i, \tag{2.17}$$

where c_{plm} is a positive number such that $\left(R_l^i\right)^\dagger O_p R_m^i \left(\left(R_l^i\right)^\dagger O_p R_m^i\right)^\dagger = c_{plm}^2 \mathbb{1}_{k_i}$.
5: Output the set $\left\{\tilde{U}^\dagger, \{d_j\}_{j=0}^J, \{k_j\}_{j=1}^J\right\}$, where $\tilde{U}^\dagger = R \cdot \mathrm{diag}\left(P_1^1, P_2^1, \ldots, P_{d_j}^J\right)$ and $\mathcal{I} = \tilde{U}\left[\mathbb{0}_{d_0} \oplus \bigoplus_{j=1}^J \left(\mathcal{M}_{d_j} \otimes \mathbb{1}_{k_j}\right)\right] \tilde{U}^\dagger$.
6: (Optional) Perform an appropriate SWAP to get $\left\{U, \{d_j\}_{j=0}^J, \{k_j\}_{j=1}^J\right\}$ such that $\mathcal{I} = U\left[\mathbb{0}_{d_0} \oplus \bigoplus_{j=1}^J \left(\mathbb{1}_{k_j} \otimes \mathcal{M}_{d_j}\right)\right] U^\dagger$.

Our Wolfram Mathematica code for this algorithm is available at [1]. An analogous algorithm for states and fixed spaces rather than observables can be found in Chapter 3.

2.2. Lindbladians

An important set of quantum channels are the ones that describe continuous time evolution. In this section we concentrate on the case of an evolution without memory—that is, given the present, the future does not depend on the past. This is called the Markov property. This property naturally arises in physical systems that are weakly coupled to an environment, as we discuss in Subsection 2.2.2. We construct Markovian local CPTP maps that prepare desirable quantum states in Chapter 3.

For a more in-depth introduction to Markovian quantum channels, see e.g. [496].

Notation 2.2.1. *We will denote by $\varepsilon_{(t_1, t_0)}(\cdot)$ a channel that corresponds to the evolution from time t_0 to time t_1, that is $\rho(t_1) = \varepsilon_{(t_1, t_0)}(\rho(t_0))$.*

Definition 2.2.2 (Markovian evolution). *If an evolution that corresponds to a family of channels* $\varepsilon_{(t,\tau)}(\cdot)$ *has the property*

$$\varepsilon_{(t_2,t_0)} = \varepsilon_{(t_2,t_1)} \circ \varepsilon_{(t_1,t_0)}, \quad t_0 \le t_1 \le t_2, \tag{2.18}$$

it is called Markovian.

2.2.1. Differential equation for quantum Markov evolution

For positive ϵ, consider the difference

$$\rho(t + \epsilon) - \rho(t) = \left(\varepsilon_{(t+\epsilon,t)} - \mathbb{1}\right)\rho(t). \tag{2.19}$$

Provided that the limit $\epsilon \to 0$ is well defined, we can obtain a linear differential equation for $\rho(t)$ (called *master equation*).

$$\frac{d\rho(t)}{dt} = \lim_{\epsilon \to 0} \frac{\varepsilon_{(t+\epsilon,t)} - \mathbb{1}}{\epsilon}\rho(t) \equiv \mathcal{L}_t\rho(t), \tag{2.20}$$

where by definition the *generator* of evolution is

$$\mathcal{L}_t \equiv \lim_{\epsilon \to 0} \frac{\varepsilon_{(t+\epsilon,t)} - \mathbb{1}}{\epsilon}. \tag{2.21}$$

For Markovian evloution, one can derive the most general form for the generators (2.21).

Theorem 2.2.3 (GKSL equation). *A differential equation is a Markovian master equation iff it can be written in the form*

$$\frac{d\rho(t)}{dt} = -i[H(t), \rho(t)] + \sum_k \left[V_k(t)\rho(t)V_k^\dagger(t) - \frac{1}{2}\{V_k^\dagger(t)V_k(t), \rho(t)\}\right], \tag{2.22}$$

where $H(t)$ and $V_k(t)$ are time-dependent operators, $H(t)$ is self-adjoint, $[\cdot, \cdot]$ is a commutator and $\{\cdot, \cdot\}$— an anticommutator.

The proof for this theorem can be found in Chapter 4 (5 in the arXiv version) of [496].

The equation (2.22) is called the Gorini–Kossakowski–Sudarshan–Lindblad (GKSL) equation [228, 328, 364]. The generator (2.21) of Markovian evlolution is called quantum Liouvillian, or Lindbladian.

If one is also interested in expectation values of unbounded operators, there are Markovian generators that are not of this from [550].

For discussion about non-Markovian dynamics, see e.g. [158].

From the Schrödinger equation it is straightforward to get that for closed system evolution with the Hamiltonian $H(t)$

$$\frac{d\rho(t)}{dt} = -i[H(t), \rho(t)], \tag{2.23}$$

thus it is tempting to interpret equation (2.22) as an open system that evolves with the Hamiltonian $H(t)$ and is coupled to an environment that manifests itself through operators $V_k(t)$. This interpretation can be made rigorous.

2.2.2. The weak coupling derivation

Let us derive a master equation for a density matrix of some system ρ_S. Assumed that the system is weakly coupled to an environment. The Hamiltonian H is a sum of Hamiltonians of the system H_S, the environment H_E and an interaction between the two H_{SE},

$$H = H_S + H_E + H_{SE}. \tag{2.24}$$

The dynamics of the entire system is governed by the equation (2.23). Let us go to the interaction picture

$$\tilde{M} = e^{i(H_S+H_E)t} M e^{-i(H_S+H_E)t} \quad \text{for arbitrary operator } M. \tag{2.25}$$

Equation (2.23) becomes

$$\dot{\tilde{\rho}} = -i\left[\tilde{H}_{SE}, \tilde{\rho}\right]. \tag{2.26}$$

The equation (2.26) can be integrated directly to give

$$\tilde{\rho}(t) = \tilde{\rho}(0) - i \int_0^t d\tau \left[\tilde{H}_{SE}(\tau), \tilde{\rho}(\tau)\right]. \tag{2.27}$$

We can substitute (2.27) back into (2.26) to obtain

$$\dot{\tilde{\rho}} = -i\left[\tilde{H}_{SE}(t), \tilde{\rho}(0)\right] - \int_0^t d\tau \left[\tilde{H}_{SE}(t), \left[\tilde{H}_{SE}(\tau), \tilde{\rho}(\tau)\right]\right]. \tag{2.28}$$

We now assume that there is initially no correlation between the system and environment $\tilde{\rho}(0) = \tilde{\rho}_S(0) \otimes \rho_E(0)$. By tracing out the environment degree of freedom we get

$$\dot{\tilde{\rho}}_S = -\text{Tr}_E \left(\int_0^t d\tau \left[\tilde{H}_{SE}(t) \left[\tilde{H}_{SE}(\tau), \tilde{\rho}(\tau)\right]\right] \right). \tag{2.29}$$

If the environment is much larger than the system and the coupling between the system and the environment is week, the dynamics of the system cannot significantly alter the environment. This yields yet another approximation, $\tilde{\rho}(t) = \tilde{\rho}_S(t) \otimes \rho_E(0)$. The master equation becomes

$$\dot{\tilde{\rho}}_S = -\text{Tr}_E \left(\int_0^t d\tau \left[\tilde{H}_{SE}(t), \left[\tilde{H}_{SE}(\tau), \tilde{\rho}_S(\tau) \otimes \rho_E(0)\right]\right] \right). \tag{2.30}$$

Finally, we employ the Markov approximation—that the time derivative of the density matrix depends only on its current state, and not on its past. This assumption is valid under fast bath dynamics, wherein correlations within the bath are lost extremely quickly, and amounts to replacing $\tilde{\rho}_S(\tau) \to \tilde{\rho}_S(t)$ on the right hand side of the equation.

$$\dot{\tilde{\rho}}_S = -\text{Tr}_E \left(\int_0^t d\tau \left[\tilde{H}_{SE}(t), \left[\tilde{H}_{SE}(\tau), \tilde{\rho}_S(t) \otimes \rho_E(0)\right]\right] \right). \tag{2.31}$$

If the interaction Hamiltonian H_{SE} has the form

$$H_{SE} = \sum_k S_k \otimes E_k \tag{2.32}$$

for system operators S_k and environment operators E_k, the master equation yields

$$\dot{\tilde{\rho}}_S = -\sum \int_0^t d\tau \left([S_i(t)S_j(\tau)\tilde{\rho}_S(t) - S_j(\tau)\tilde{\rho}_S(t)S_i(t)] \operatorname{Tr}[E_i(t)E_j(\tau)\rho_E(0)] + \right.$$
$$\left. [\tilde{\rho}_S(t)S_j(\tau)S_i(t) - S_i(t)\tilde{\rho}_S(t)S_j(\tau)] \operatorname{Tr}[E_j(\tau)E_i(t)\rho_E(0)] \right). \quad (2.33)$$

If system and environment operators can be chosen in a way so that $S_{2i+1} = S_{2i}^\dagger$, $E_{2i+1} = E_{2i}^\dagger$ and, due to rapid decay of correlations in the environment, $\operatorname{Tr}\left[E_i(t)E_j^\dagger(\tau)\rho_E(0)\right] = \frac{1}{2}\delta_{i,j}\delta(t-\tau)$, we obtain

$$\dot{\tilde{\rho}}_S = -\frac{1}{2}\sum_i \left[S_i(t)S_i^\dagger(t)\tilde{\rho}_S(t) - S_i^\dagger(t)\tilde{\rho}_S(t)S_i(t) \right]$$
$$-\frac{1}{2}\sum_j \left[\tilde{\rho}_S(t)S_j(t)S_j^\dagger(t) - S_j^\dagger(t)\tilde{\rho}_S(t)S_j(t) \right]$$
$$= \sum \left(S_i^\dagger(t)\tilde{\rho}_S(t)S_i(t) - \frac{1}{2}\left\{ S_i(t)S_i^\dagger(t), \tilde{\rho}_S(t) \right\} \right). \quad (2.34)$$

By setting $S_k(t) = \tilde{V}_k(t)$, we arrive to equation (2.22). A slightly more general derivation is contained in [55].

Let us summarize. The differential equation for Markovian evolution is GKSL equation (2.22). This equation can also be derived by assuming weak enough interaction with an environment much larger than the system that does not become correlated with it. This makes GKSL equation an extremely useful tool for the study of open quantum systems. For more discussion, see e.g. [496, 468], [639, Chapter 7], [648, Section 1.3] and [279, 22].

2.3. Entanglement

Quantum states can have correlations that are not present in any classical theory. We deal with states exhibiting such correlations throughout this thesis. In this section we discuss how to define these non-classical correlations and how to quantify them. For more information on the topic, see e.g. [276, 303].

Definition 2.3.1 (Entanglement). *A state $\rho \in \mathcal{T}\left(\bigotimes_i \mathcal{H}_i\right)$ is called classically correlated [619] if it can be written as*

$$\rho = \sum_j \lambda_j \bigotimes_i \rho_i^j, \quad \lambda_j \geq 0, \ \rho_i^j \in \mathcal{T}(\mathcal{H}_i). \quad (2.35)$$

Otherwise, the state is called entangled.

We can make the definition more precise by including the amount of parties that are entangled.

Definition 2.3.2 (Entanglement depth). *A state $\rho \in \mathcal{T}\left(\bigotimes_i \mathcal{H}_i\right)$ is called m-particle entangled if it can **not** be written as*

$$\rho = \sum_j \lambda_j \bigotimes_i \rho_{i,m_{ij}}^j, \quad \lambda_j \geq 0, \ \rho_{i,m_{ij}}^j \in \mathcal{T}\left(\bigotimes_{k=1}^{m_{ij}} \mathcal{H}_{\pi_i(k)}\right), \quad \max(m_{ij}) \leq m-1, \quad (2.36)$$

where $\{\pi_i\}$ is a set of permutations. The maximum m such that the state ρ is m-particle entangled is called the entanglement depth.

It is often important to know not just that there is some entanglement, but to quantify it. It can be easily done for pure bipartite systems.

Indeed, the only classically correlated pure states are of the form $|\psi\rangle = \bigotimes_i |\psi_i\rangle$. For bipartite state one can use Schmidt decomposition $|\psi\rangle = \sum_i^d \lambda_i |e_i\rangle |f_i\rangle$ and quantify entanglement using Schmidt coefficients.

Claim 2.3.3 (von Neumann entropy). *For a bipartite pure state $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ and corresponding reduced density matrix $\rho_2 = \mathrm{Tr}_{\mathcal{H}_1}(|\psi\rangle\langle\psi|)$ the von Neumann entropy*

$$S_{vN}(\rho_2) = -\mathrm{Tr}\left[\rho_2 \ln(\rho_2)\right] \tag{2.37}$$

quantifies the amount of entanglement. We will use the notation $S_{vN}(|\psi\rangle)$ interchangeably with $S_{vN}(\rho_2)$.

The claim 2.3.3 can be axiomatised by treating entanglement as a resource. A state $|\psi\rangle$ can be transformed into state $|\phi\rangle$ via only local operations and classical communication (LOCC) iff $S_{vN}(|\psi\rangle) \geq S_{vN}(|\phi\rangle)$ [422]. However, there are other *entanglement monotones*—entanglement measures that do not increase under LOCC [600]. A prominent family of entanglement monotones are Rényi entropies [494, 600]

$$S_\alpha(\rho) = \frac{1}{1-\alpha} \ln \mathrm{Tr}(\rho^\alpha). \tag{2.38}$$

While one can prove that $S_{vN} = \lim_{\alpha \to 1} S_\alpha$ [410], von Neumann entropy has some unique algebraic properties (see [170, 463]) that make it especially useful. See e.g. [121] for a review.

For three parties or more, it is much harder to formalise the amount of entanglement in the system. One of the reasons is that there is no Schmidt decomposition for non-bipartite systems. Moreover, there are different classes of entangled states that cannot be transformed via LOCC into each other. There are three classes for tripartite states [177] and infinitely many classes for more than three parties. In fact, for more than three parties almost any state is a maximally entangled state [514, 105]! Thus, quantifying entanglement becomes somewhat meaningless in the LOCC framework for many parties. Understanding multipartite entanglement and suitable resource theories is an active research topic (see e.g. [135, 420, 555]).

It can be NP-hard to check if even a bipartite mixed state is entangled [244, 222]. Thus, any way to quantify mixed state entanglement is either hard to compute or cannot discriminate between certain classes of mixed and entangled states. Despite these drawbacks, there are many entanglement measures for mixed states, e.g. logarithmic negativity [602], quantum Fisher information [265, 224, 452], relative entropy [591, 593], convex roof of entanglement or entanglement of formation [580, 258], Wigner function negativity [593, 159], entanglement of distillation [65], and others [424, 592], that can be useful for appropriate tasks.

Luckily for us, in this thesis we mainly deal with the situations when intuition about pure bipartite states can be generalised to mixed multipartite states.

In practice one usually works with a relatively small subset of states. If there is an interest in using non-classical properties of this subset, it is important to have a certificate that witnesses that these states are indeed entangled.

Definition 2.3.4 (Entanglement witness). *$A \in \mathcal{B}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is called an entanglement witness for $\rho \in \mathcal{T}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ an entangled state if $\mathrm{Tr}[\rho A] > 0$ and $\mathrm{Tr}[\sigma A] < 0$ for all classically correlated $\sigma \in \mathcal{T}(\mathcal{H}_1 \otimes \mathcal{H}_2)$.*

Proposition 2.3.1. *For any $\rho \in \mathcal{T}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ there exists an entanglement witness.*

For proof see e.g. [303, Section 2.4].

Entanglement can find numerous applications, from modeling of ground states (see Section 2.4), to quantum computing (see Section 2.5); from secure cryptography to precise measurements (see Section 2.7). Entanglement is at the core of the quantum theory, and understanding its properties is a key to using it in practice.

2.4. Tensor networks

Tensor networks (TNs) are a family of variational classes well suited to study highly entangled quantum many-body systems. In this section we introduce this family and proceed to discuss a physically motivated subfamily of states—matrix product density operators (MPDOs). MPDOs describe mixed one-dimensional states that obey area law for entanglement. In Chapter 3 we discuss how an MPDO space of choice can be prepared via long-term local dynamics.

TNs were initially used to construct [196] and numerically find [625] ground states of one-dimensional spin chains. However, the range of applicability of this formalism has quickly extended to higher dimensions, excited and thermal states. TNs are currently also investigated as a natural framework to classify exotic phases of quantum matter [526, 484], as the basis for new non-perturbative formulations of the renormalization group [625, 601, 190, 189, 324] and interacting quantum field theories, as a lattice realization of the AdS/CFT correspondence in quantum gravity [442], to maximise achievable estimation precision in quantum metrology [118], quantum error correction [52, 201, 200], category theory [57] and as a variational class for machine learning [560, 607, 497]. See e.g. [431, 649, 182, 565] for a review.

We will use the term *tensor* to refer to a multidimensional array of complex numbers.

To understand why TNs are natural and needed, let us consider a chain of n spin-j particles. If we are interested, for example, in the ground-state wave function of an arbitrary Hamiltonian, it is a superposition of computational basis vectors

$$|\psi\rangle_{GS} = \sum_{i_1,\ldots,i_n \in \{0,\ldots,2j\}} C_{i_1,\ldots i_n} |i_1 \ldots i_n\rangle. \tag{2.39}$$

In general, it leaves us with $(2j+1)^n - 2$ degrees of freedom - that is, the number of components of $C_{i_1,\ldots i_n}$ minus phase and normalization. This immediately yields a problem, as even relatively small systems of, say, a hundred particles do not fit into the memory of not only any existing,

but also any currently conceivable classical computer. This dictates the need to work in some specific subspace of wave functions, and this subspace should contain wave functions that are arbitrarily close to some set of entangled ground-states.

One way to tackle this problem is to use a variational class that limits the amount of entanglement between different local parts of the system. Let us do a Schmidt decomposition (2.1.1) of $|\psi\rangle_{GS}$ with respect to spins $\{1, \ldots, k\}$ and $\{k+1, \ldots, n\}$ and absorb the Schmidt coefficients into the tensor A

$$C_{i_1, \ldots i_n} = \sum_{a=1}^{\chi} A^a_{i_1, \ldots i_k} B^a_{i_{k+1}, \ldots i_n}. \tag{2.40}$$

The quantity χ is called the *bond dimension*. Now there are $\chi((2j+1)^k + (2j+1)^{n-k}) - 2$ degrees of freedom—significantly fewer than $(2j+1)^n - 2$, if χ is not too large.

The bond dimension limits the maximal amount of entanglement. Indeed, for a bipartite wave function $|\psi\rangle_{GS}$ with bond dimension χ there are exactly χ Schmidt coeficients $\{\lambda_i\}_{i=1}^{\chi}$. The entropy of entanglement (see claim 2.3.3) for such state is

$$S_{vN}(|\psi\rangle_{GS}) = -\sum_{i=1}^{\chi} \lambda_i \ln(\lambda_i) \le \ln(\chi). \tag{2.41}$$

Nevertheless, this ansatz can give highly entangled states.

Example 2.4.1. Let us consider the GHZ state

$$|GHZ^n\rangle = \frac{|0\rangle^{\otimes n} + |1\rangle^{\otimes n}}{\sqrt{2}}, \tag{2.42}$$

which is known to be entangled with maximal depth (see definition 2.3.2 and e.g. [237]), and demonstrate that it can be obtained via the decomposition discussed above. Take $a \in \{0, 1\}$, so that the bond dimension is 2. We can set

$$A^a_{i_1, \ldots i_k} = \begin{cases} \frac{1}{\sqrt[4]{2}} & \text{if } a = i_1 = \cdots = i_k \\ 0 & \text{otherwise} \end{cases} \tag{2.43}$$

and use an analogous definition for $B^{\alpha}_{i_{k+1}, \ldots i_n}$ - it is non-zero and equals $\frac{1}{\sqrt[4]{2}}$ only if all of the indices are the same. It is trivial to check that with such definitions we get $|GHZ^n\rangle$.

As $\chi((2j+1)^k + (2j+1)^{n-k}) - 2$ can still be too large, we would like to reduce A and B further— that is, to introduce smaller tensors that are being contracted with each other.

Formulas for contractions of n tensors often start to look very complicated. For simplicity, a diagrammatic notation was invented. It has also proven to help introduce patterns of tensor contractions that correspond to physical intuition.

In *tensor network diagrams* tensors are represented by shapes and their indices by lines emerging from the shapes. When two shapes share a line, it corresponds to a contraction of the corresponding index. It is also possible to join several lines into one multi-index, if the same two

Figure 2.1.: Diagrams for (≤ 3) tensors of rank ≤ 3 and (n) an n-rank tensor $C_{i_1 i_2 \ldots i_n}$.

tensors share them. Likewise, it is possible to join shapes in such a way that outer lines do not change.

$$C_{i_1 i_2 \ldots i_n j_1 j_2 \ldots j_m} = \sum_{a_1 \ldots a_K} A^{a_1, \ldots a_K}_{i_1 i_2 \ldots i_n} B^{a_1, \ldots a_K}_{j_1 j_2 \ldots j_m} =$$

Figure 2.2.: Joining shapes and introducing multi-indices.

Consider a physical system that is defined on some arbitrary lattice. Let A be an operator which depends only on the lattice sites $k, \ldots, m$. Then we can also represent such an operator by some shape with $m - k$ ingoing and outgoing lines – $2(m-k)$ lines in total. If we have such a diagram for $|\psi\rangle$, we will denote $\langle\psi|$ by the same diagram but flipped upside down. Then we will have the following figure 2.3 representing the expectation value $\langle\psi, A\psi\rangle$ of A with respect to ψ

Figure 2.3.: Diagrammatic expression for $\langle\psi, A\psi\rangle$.

Suppose $A : V^n \to V^m$ is an operator that has n ingoing and m outgoing lines. Given positive definite Hermitian forms $\langle\cdot, \cdot\rangle_{V^n} : V^n \times V^n \to \mathbb{C}$ and $\langle\cdot, \cdot\rangle_{V^m} : V^m \times V^m \to \mathbb{C}$, one can define the adjoint of operator A, namely $A^\dagger : V^m \to V^n$, via

$$\langle A\psi, \phi\rangle_{V^m} = \langle\psi, A^\dagger\phi\rangle_{V^n} \quad \forall\psi \in V^n \ \text{ and } \ \forall\phi \in V^m \tag{2.44}$$

and such object is represented, just like $\langle\psi|$, by a flipped diagram (see figure 2.4).

Figure 2.4.: The adjoint of A is represented by the flipped diagram.

We would like to represent objects like $C_{i_1,\ldots i_N}$ which have N outer lines. If we have a tensor with k outer lines, we can contract it with another tensor so that the total number of outer lines increases. This requires at least tensors of rank 3 (see figure 2.5).
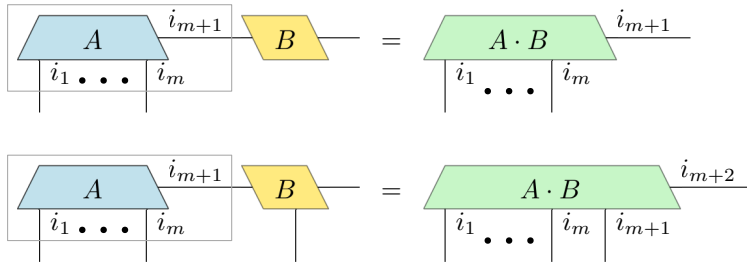

Figure 2.5.: In order to construct a tensor with many legs from a low-rank connected tensors, one needs tensors of at least rank 3.

If a TN for a state has disconnected components, it is a product state where each party corresponds to a connected component.

We can obtain a rank n tensor with no disconnected parts by contracting $O(n)$ rank-3 tensors. This gives a state which is potentially entangled and has only $O(n)$ parameters! This is a great simplification of the task, however, it is essential to see that such states are physically relevant.

2.4.1. Area law for entanglement

Physical interactions in many-body systems are typically local: individual constituents interact mainly with their few nearest neighbors. This locality of interactions is reflected by the ground state entanglement entropy scaling laws. The entropy of the reduced state of a subregion often grows with the subregion's boundary area, not its volume. See e.g. [638, 183] for the discussion.

This "area law" for entanglement can be used as an inspiration for TN classes. Indeed, consider a TN diagram built from tensors that are situated in some lattice. If each constituent tensor is connected only to neighbouring sites by legs of the same dimension and has one open physical leg, the TN diagram exhibits an area law.

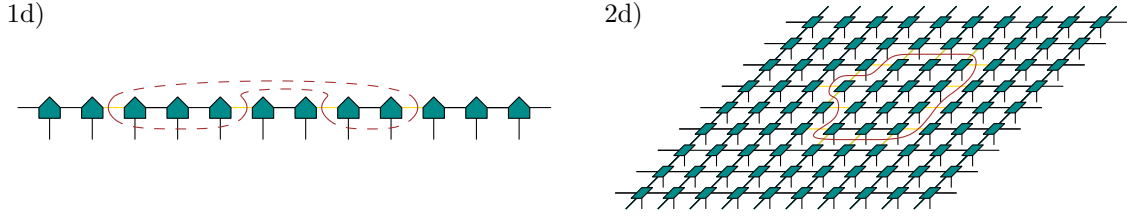1d)   2d)

Figure 2.6.: Tensor networks in 1 and 2 dimensions that obey the area law for entanglement. The number of bonds (gold) that are connecting a subregion (encircled by the brown line) is proportional to the area of the subregion (solid part of the brown line).

Such TN diagrams form the basis of powerful numerical methods. We will concentrate on these TNs in one dimension for the rest of this section.

2.4.2. Matrix product states (MPS)

The space of matrix product states (MPS) is spanned by the states of the form

$$|\psi\rangle_{MPS}^{a_0,a_n} = \sum_{i_1,...,i_n \in \{0,2j\}} \sum_{a_1,...,a_{n-1}=1}^{\chi} A_{i_1}^{a_0 a_1,[1]} \cdot A_{i_2}^{a_1 a_2,[2]} \cdot \ldots \cdot A_{i_n}^{a_{n-1} a_n [n]} |i_1 i_2 \ldots i_n\rangle \qquad (2.45)$$

There are $(2j+1)n\chi^2 - 2 = O(n)$ parameters in such an ansatz, where χ is the bond dimension.

For simplicity, we will concentrate on the translational invariant case

$$A_i^{ab,[j]} = A_i^{ab} \quad \forall j. \qquad (2.46)$$

Most results discussed in this subsection can be trivially generalised to the non-translational invariant case.

Fig.2.7shows the TN diagram for an MPS.

$$A_i^{\alpha\beta} \equiv \quad {}^\alpha\!\!\!\vartriangle^{\!\beta}_{|i} \qquad |\psi\rangle_{MPS}^{a_0,a_n} = \quad {}^{a_0}\!\!\vartriangle\!\!\vartriangle\!\!-\!\bullet\ \bullet\ \bullet\!-\!\!\vartriangle^{a_n}$$

Figure 2.7.: Diagrammatic representation of MPS.

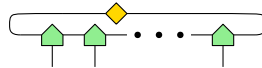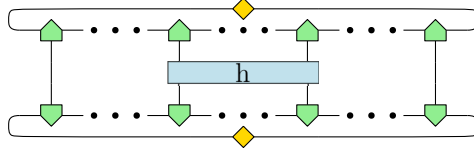Boundary conditions can be imposed by contracting free horizontal legs with some additional tensor.

Figure 2.8.: MPS (green) with boundary conditions (gold).

Expectation values of local operators in MPS can be computed efficiently. Let us consider a k-local operator h (see Fig. 2.9) for a system of length $n \gg k$.

Figure 2.9.: Expectation value of k-local operator h.

Let us introduce the transfer operator (see Fig.2.10)

$$T^{ab;\alpha,\beta} = \sum_i A_i^{ab} \left(A_i^{\alpha\beta} \right)^\dagger.$$ (2.47)

Figure 2.10.: Transfer operator for MPS.

The transfer operator can be treated as a $\chi^2 \times \chi^2$ matrix. A contraction of m transfer operators (see Fig.2.11) is a multiplication of m matrices of dimensions $\chi^2 \times \chi^2$ and this operation requires just $O(\chi^4)$ memory and at most $O(m\chi^6)$ time. For the translational invariant case we can take a power $(T)^m$, which is even faster.
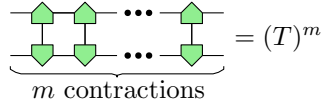
Figure 2.11.: Contraction of m transfer operators.

Let us assume that h acts on spins $m, m+1, \ldots, m+k-1$. Let us denote the tensor that specifies the boundary conditions by $B^{a_0 a_n}$. The expectation value

$$\langle h \rangle = \sum_{a_0, a_n, b_0, b_n} \langle \psi |_{MPS}^{b_0, b_n} \left(B^{b_0 b_n} \right)^\dagger h B^{a_0 a_n} | \psi \rangle_{MPS}^{a_0, a_n}.$$ (2.48)

is, once $(T)^m$ and $(T)^{n-m-k}$ have been contracted, a contraction of $2k+5$ tensors with at most $\max(4, 2k)$ legs with bond dimensions that are either $2j+1$ or χ (see Fig.2.9 and Fig.2.12). Under assumption that both k and χ are small, this contraction is easy to perform.
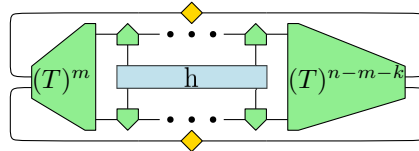
Figure 2.12.: Expectation value of k-local operator h after contracting transfer operators to the left and to the right of it.

The procedure also works for operators that act on k sites not adjacent to each other. Let us demonstrate it for $k = 2$. Suppose the operator of interest acts on sites m and $m + s$. We can efficiently contract all adjacent operators (see Fig.2.13). Once again, if χ is small, the rest of the contractions is easy to perform. The procedure is straightforward to generalize for any $k \ll n$.
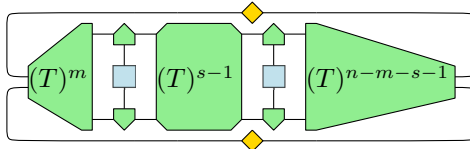

Figure 2.13.: Partially contracted TN diagram for expectation value of operator acting on on sites m and $m + s$.

Let us normalise the wave function so that the eigenvalues of transfer operators are between 0 and 1. This can always be done as the transfer operator is manifestly positive.

The appearance of $(T)^{s-1}$ in Fig.2.13 leads to the fact that MPS can represent only exponentially decaying correlations. Indeed, entries of normalized $(T)^{s-1}$ fall exponentially with s.

Despite being limited to exponentially decaying correlation functions, MPS can be used to describe critical systems. On the numerical side, it is possible to extract certain properties of the system by looking at the scaling with growing bond dimension, see e.g. [568, 349, 654]. On the theoretical side, König and Scholz have developed a rigorous truncation procedure for MPS approximations to (1+1)d CFTs [326, 325]. König and Scholz have proved that the error, namely, the difference between any correlation function and the MPS representation of this correlation function, decreases exponentially with the *truncation parameter* N, while bond dimension increases as the number of vectors with weight less or equal to nN, where n is the number of (non-vacuum) fields.

Translational invariant MPS allows calculations in the thermodynamic limit. Normalized $(T)^\infty$ is a projector on the eigenspace of T with the eigenvalue 1.

We can cut the system in two parts by removing one of the connections between tensors that constitute the MPS. Then, we can use Schmidt decomposition

$$|\psi\rangle_{MPS} = \sum_{\alpha=0}^{\infty} \lambda_\alpha \psi_{L,MPS} \otimes \psi_{R,MPS} \tag{2.49}$$

with Schmidt coefficients λ_α. We can quantify an error of the approximation as

$$\text{Error}_{MPS} = \sum_{\alpha=\chi}^{\infty} |\lambda_\alpha|^2. \tag{2.50}$$

Hastings has proved [257] (see [33] for the proof with exponentially better parameters) that ground states of gapped Hamiltonians in one spatial dimension can be approximated arbitrarily well by MPS in an efficient manner (see Fig. 2.14).

Theorem 2.4.2 (MPS approximates ground states)**.** *For ground states of one-dimensional gapped local Hamiltonians $Error_{MPS}$ scales as χ^{-c}, where c is a constant that depends on the energy gap and the dimension of the Hilbert space on each site.*
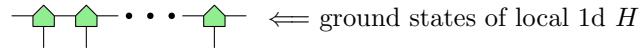
Figure 2.14.: Matrix product states can approximate ground states of 1d local Hamiltonians well (see theorem 2.4.2)

The MPS ansatz is used in various numerical algorithms including the *density matrix renormalization group* (DMRG) ([625], for a more up-to-date review see e.g. [524]). Today it remains a method of choice for the analysis of a large number of one-dimensional systems.

An important generalization of the MPS to two-dimensional (or, in a similar way, higher-dimensional) systems are *projected entangled pair stares* (PEPS) [595, 599].
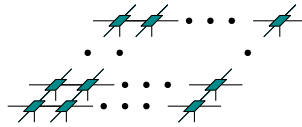
Figure 2.15.: PEPS in two dimensions.

This approach enables not only to describe the bulk of the material but also the edge modes [646]. However, PEPS is much harder to work with than MPS, both numerically and theoretically. For open mathematical questions regarding PEPS, see e.g. [131].

There have been studies that generalize the MPS to a continuous number of variables [596]. These continuous MPS, or cMPS, provide a new approach to quantum field theory and a fresh view on the real-space renormalization group methods (see e.g. [432, 292]).

MPS algorithms can also be implemented on quantum computers, see e.g. [49].

2.4.3. Parent Hamiltonian

While it is established that MPS can be used to approximate ground states of local Hamiltonians (see theorem 2.4.2 and e.g. [524]), it is worthwhile to draw a connection in the other direction. Given a state, for example an MPS, can we construct a local Hamiltonian H such that this MPS is the ground state of H (see Fig.2.16)?
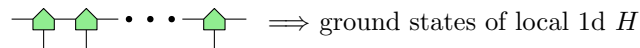
Figure 2.16.: Matrix product states are the ground states of a parent Hamiltonian (see definition 2.4.5 and theorem 2.4.7)

One application of such construction is to obtain Hamiltonians for novel integrable models. This was one of the original motivations to introduce tensor networks. The construction of a Hamiltonian from what is now considered a prototypical MPS occurred in [15] for the AKLT state [14]. This result, together with [5], has inspired a series of generalisations [194, 195] culminating in a general way to produce an integrable model from an MPS [196].

The other application is to prepare states of interest. If a certain state $|\psi\rangle$ is the ground state of a **gapped** Hamiltonian H that can be realised in an experiment, and if there is either a way to ensure that the temperature is low enough or to have energy dissipation (see Subsection 2.5.7), the system will evolve towards $|\psi\rangle$.

It is important to concentrate on local Hamiltonians. Indeed, the unique ground state of $H = \mathbb{1} - |\psi\rangle\langle\psi|$ is $|\psi\rangle$. However, $\mathbb{1} - |\psi\rangle\langle\psi|$ can have interacting terms that have arbitrary ranges and do not fall with distance. These kind of interactions are very hard to realise experimentally with sufficient precision (see Section 2.5), often rendering such global Hamiltonians rather useless.

It is also vital to be able to construct the Hamiltonian using limited computational resources. One way to do this is to demand that a model is frustration free—a ground state minimizes each local part in the Hamiltonian. This is especially useful if the density matrix of a subregion is efficiently computable, as is the case for MPS.

A prototypical frustrated system—the antiferromagnetic Ising chain on a triangular lattice was studied in [611], yet a general concept of frustration was defined in [581, 586].

Definition 2.4.3 (Frustration free Hamiltonian). *A k-local Hamiltonian $H = \sum_i h_i$, where each h_i acts non-trivially only on k sites is called frustration free, if*

$$\inf_{|\psi\rangle} \langle\psi|H|\psi\rangle = \sum_i \inf_{|\psi\rangle} \langle\psi|h_i|\psi\rangle \tag{2.51}$$

Example 2.4.4. Consider a spin-$\frac{1}{2}$ antiferromagnetic chain described by the Hamiltonian $H = J \sum_{i=1}^{N} \sigma_i^z \sigma_{i+1}^z$ with $J > 0$ and periodic boundary conditions—the site $N + 1$ identified with site 1.

If N is even, the energy is minimised by the states in which the spins are anti-aligned, so there are two ground states. In this case the Hamiltonian is frustration free. For odd N, there is a kink produced by two contiguous aligned spins in a ground state. There are $2N$ ground states and the system is frustrated. See Fig. 2.17 for $N = 4, 3$.
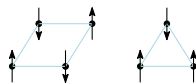
Figure 2.17.: Ground states of $N = 4$ (left, one of 2) and $N = 3$ (right, one of 6) spin-$\frac{1}{2}$ antiferromagnetic chain with periodic boundary conditions. The chain is not frustrated for even N and frustrated for odd N.

Any short-range Hamiltonian with a gap between the ground and excited states can be approximated by a frustration free Hamiltonian with interaction length that is logarithmic in system size [256].

Combining the demand for local and frustration free interactions we can define a class of Hamiltonians we are looking for—parent Hamiltonians.

Definition 2.4.5 (Parent Hamiltonian)**.** *A Hamiltonian H is called parent for a linear space of states V if it is local, frustration free and*

$$\inf_{|\psi\rangle} \langle \psi | H | \psi \rangle = \langle \phi | H | \phi \rangle \quad \forall |\phi\rangle \in V \tag{2.52}$$

Let us construct a candidate for MPS parent Hamiltonian.

Construction 2.4.6 (Parent Hamiltonian for MPS)**.** Consider an MPS $|\psi\rangle$ and its reduced density matrix for k particles, starting from particle i: $\rho^{[i,i+k-1]}$. Let h_i be a projector on $\ker\left(\rho^{[i,i+k-1]}\right)$. Construct $H_P = \sum_i h_i$.

While it is easy to see that $|\psi\rangle$ is indeed a ground state of H_P, there might be other ground states of H_P if the locality k is chosen inappropriately. Nevertheless, H_P is indeed a parent Hamiltonian for the MPS $|\psi\rangle$ if k is chosen large enough.

Let us argue that the number of ground states of H_P can indeed depend on locality of interactions. Suppose that l-particle $|\psi\rangle \in \mathcal{H}^l$ has a bond dimension χ and the physical dimension $\dim(\mathcal{H}) = d$. By construction 2.4.6, $\max\left(\mathrm{rank}(\rho^{[i,i+k-1]})\right) = \chi^2$. Let us first consider a situation when $d^k \leq \chi^2$. This can lead to $\rho^{[i,i+k-1]}$ being full-rank, which in turn produces $H_P = 0$, and thus any state is a ground state of H_P. Let us show that this problem can be solved by increasing k. Let us increase k so that $d^k > \chi^2$. Then each local term h_i has at least $d^k - \chi^2$ excited states, leading to non-trivial H_P. Note that the higher the locality k is, the smaller is the proportion of ground states in the spectrum of each h_i.

But is there a sufficient interaction range such that the construction 2.4.6 produces a parent Hamiltonian? And, if so, what happens to the gap of this Hamiltonian in the thermodynamic limit? While each h_i is gapped, can we be sure that $\lim_{N\to\infty} \sum_{i=1}^{N} h_i/N$ is also gapped? Fortunately, in the translational invariant case, the answer to both of these questions is yes.

Theorem 2.4.7. *Consider a transnational-invariant MPS $|\psi\rangle$ defined by a tensor A. Let an H_P be a parent Hamiltonian candidate constructed via 2.4.6 for $|\psi\rangle$. There exists a function $i(A)$ such that if the interaction range k of H_P satisfies $k > i(A)$, then $|\psi\rangle$ is the unique ground state of H_P, and H_P has a spectral gap above the ground state energy. Moreover, $i(A) \leq \chi^2(\chi^2 - d + 1)$, where χ is the bond dimension and d—the physical dimension of $|\psi\rangle$.*

Theorem 2.4.7 was proven in [512], building on earlier results obtained in [450, 415, 196]. A detailed discussion about parent Hamiltonians can be found in e.g. [506].

2.4.4. Matrix product density operators (MPDOs)

A direct generalization of MPS for mixed states are matrix product density operators (MPDOs) [657, 597]. The length-l MPDO with bond dimension χ can be written as

$$\rho_{MPDO}^{\alpha_0\alpha_l} = \sum_{i_1,j_1...,i_l,j_l\in\{0,2j\}} \sum_{\alpha_1,...,\alpha_{l-1}=1}^{\chi} R_{\alpha_0\alpha_1}^{i_1j_1,[1]} R_{\alpha_1\alpha_2}^{i_2j_2,[2]} \ldots R_{\alpha_{l-1}\alpha_l}^{i_lj_l,[l]} |i_1,i_2,\ldots i_l\rangle\langle j_1,j_2,\ldots j_l| \tag{2.53}$$

for some family of rank-4 tensors $R^{[k]}$ (see Fig. 2.18).

$$\rho_{MPDO}^{\alpha_0 \alpha_l} \equiv {}^{\alpha_0}\boxed{R^{[1]}} - \boxed{R^{[2]}} \cdots \boxed{R^{[l]}}{}^{\alpha_l}$$

Figure 2.18.: Diagrammatic representation of MPDOs.

Just like MPS, MPDOs obey the area law for entanglement (see Section 2.4.1). That is, MPDOs efficiently approximate mixed states for which entanglement of purification grows with the area of a subregion [287].

MPDOs are well suited to represent thermal states of local Hamiltonians [122, 398, 256, 624, 77], see Fig. 2.19. In particular, it is sufficient for the bond dimension to scale as $\chi = (N/\epsilon)^{O(\beta)}$, where N is the system size, β is the inverse temperature and ϵ is the error in approximation the thermal state [398]. The error is defined as $\left\| \rho - e^{-\beta H}/Z \right\|_1 = \epsilon$, where ρ is the approximation and $e^{-\beta H}/Z$ is the thermal state. Conveniently, this result holds in any dimension [398].

$$\text{MPDO} \equiv \text{—}\boxed{} \boxed{} \text{—} \cdots \text{—} \boxed{} \text{—} \Longleftarrow \text{ thermal states of local } H$$

Figure 2.19.: MPDOs can approximate thermal states of local Hamiltonians for high enough temperatures. This result can be generalized to arbitrary dimensions [398].

Algorithms that use MPDOs as a variational class are more elaborate than the algorithms for MPS. Any non-degenerate MPS represents a wave function, but a matrix product operator (MPO) must be positive to be an MPDO. It is NP-hard in the system size to check whether a given MPO is positive [318].

One way around the positivity problem is to work with a purification (see corollary 2.1.8) of an MPDO to an MPS [290, 617] (see Fig.2.20).
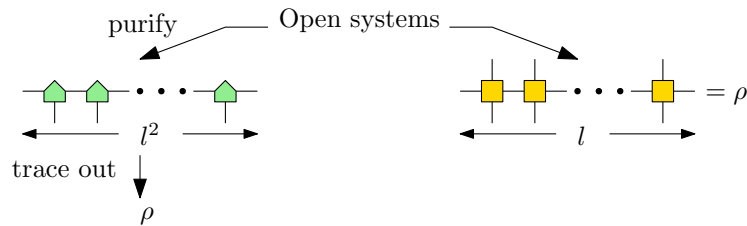
Figure 2.20.: An MPDO obtained by tracing out additional degrees of freedom in an MPS is guaranteed to be positive.

However, the bond dimension of the purification may grow significantly [154, 155] and translational invariance may be lost [151]. However, the growth of the bond dimension can be mitigated if a purification is performed up to a small error [153]. Another way around the positivity problem is to unravel the master equation and then employ pure-state techniques, at the expense of having to sample over many realizations [146, 397]. Yet another option is to obtain an MPDO of interest by evolving a simple initial MPDO state and ensuring that the errors during time evolution stay small [597], or by searching only over the stable space of a Lindbladian (see Section

2.2)[143]. See e.g. [86] for comparison between methods based on time evolution and sampling realizations of pure states and e.g. [570] for comparison with methods based on purification.

MPDOs have been successfully used for numerical investigation of various physical systems (see e.g. [40, 41, 42, 564, 371, 219, 288, 117, 455]). See e.g. [615] for a review of methods based on MPDOs and a discussion about their alternatives. There are open-source software packages that can be used for such simulations, e.g. [19, 289].

This makes MPDOs a very compelling variational class to study stable spaces of local Lindbladians (Fig.2.21).

$$\text{MPDO} \equiv \quad\boxed{}\boxed{}\cdots\boxed{} \quad \leftarrow \quad \text{stable space of local } \mathcal{L}$$

Figure 2.21.: While there are more open qustions compared to MPS, MPDOs are known to be a very powerful variational class for stable spaces of local Lindbladians.

However, unlike for MPS, the question about parent Lindbladians for MPDOs is not yet fully settled (see Fig.2.22).

$$\text{MPDO} \equiv \quad\boxed{}\boxed{}\cdots\boxed{} \quad \Longrightarrow? \quad \text{stable space of local } \mathcal{L}$$

Figure 2.22.: Is there a parent Lindbladian for a given space of MPDOs?

We show how to construct a k-local parent Lindbladian or to verify that it does not exist for a given space of MPDOs and locality k in Chapter 3.

2.5. Quantum computation

In this section we discuss how highly tunable quantum systems can perform computations inaccessible to classical computers. We start by introducing *quantum simulators*—adjustable systems that can reproduce the behaviours of important, yet complex and often poorly understood quantum systems. We proceed to highlight the benefits that *quantum computers*—digital quantum simulators, have. We then describe what conditions quantum computers have to satisfy to be maximally useful. Among these conditions is the long relevant coherence time. We outline how with error correction the coherence time can be prolonged arbitrarily, yet current quantum hardware is too noisy for error correction. Nevertheless, even noisy intermediate-sized quantum computers are believed to be an enabling technology; we assume access to such a machine in Chapters 4 and 5. We conclude this section with a description of a simulation approach based on designed interactions and explain how it can **gain** robustness from designed dissipation. We assume access to a dissipation-based device in Chapter 3.

2.5.1. Quantum simulators and engineering

Insights about some systems are easier to obtain by employing a simulator—a system that shares key features with the original, but is simpler and can be precisely controlled. While classical simulation is available for a restricted set of problems, such as studying few-particle systems or calculation of ground- and thermal-state properties in low dimensions (see section 2.4), in general such simulation is expected to be inefficient. However, current progress in building quantum systems that are highly controllable and scalable enables quantum simulators. Originally proposed in [382] and [205], various physical platforms are currently used as quantum simulators [8].

Highly controllable systems can be interesting not just as simulators, but in their own right. They can make scenarios that have previously been merely thought experiments and toys for theoreticians a reality. Examples include systems with synthetic dimensions (e.g. [470, 82, 376, 433]) and artificial gauge fields (e.g. [147, 627, 218, 241, 45]), quantum computing (e.g. [300]) and cellular automata (e.g. [199]), many-body localisation [525, 88, 129, 375, 25] and systems with exotic long-range order (e.g. [604, 603, 539, 306]). The art of choosing useful and interesting systems that are within experimental reach can be called quantum engineering.

Of particular interest for this thesis is quantum state engineering—the preparation of useful multipartite entangled states (e.g. [81]).

2.5.2. Digital quantum computers

Several challenges are present in any highly tunable system. First of all, it is important to build systems that show a rich variety of behaviours using only few operations that are experimentally available. Second, it is not trivial to design control protocols and tuning parameters in such a system to obtain something useful. Third, any real-world control is not perfect and is prone to errors. Thus, it is crucial to mitigate such errors.

All of these problems can be accessed via using *digital* devices, see e.g. [300, 424]. A sequence of operations performed on a digital device is called *computation.* Computation can be defined as via the circuit model (see Fig. 2.23, [163] and e.g. [300, Section 1.3]). First, an elementary subsystem is defined. In classical computation, the most common elementary subsystem is a bit. In quantum computation, an elementary subsystem is associated with a Hilbert space $\mathcal{H} = \mathbb{C}^d$; d is usually chosen as 2, and such subsystem is called a *qubit.* Each elementary subsystem corresponds to a *wire.* A circuit consists of wires and elementary operations on wires, also called *gates.* Each gate is an element of a finite set of operations $\mathcal{G}$ that, without the loss of generality, can be assumed to be reversible. Thus, for quantum computation, $\mathcal{G}$ can be assumed to be a finite subset of unitary operations. The wires never feedback to a prior location in the circuit. At each time t each wire enters at most one gate. The outputs are read off the wires leaving the circuit at the right side of the diagram. The outputs can either be classical values or quantum states. It is sufficient to assume that classical outputs are measured in some pre-set *computational* basis.
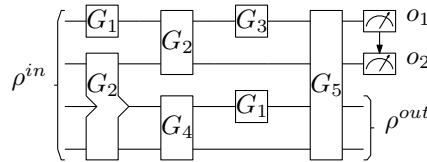
Figure 2.23.: A circuit diagram. The horizontal lines represent wires carrying the elementary subsystems, and the blocks represent gates. The same gate can be applied to different and not necessarily adjacent wires, e.g. gate G_2 on this diagram. Gates can be local, such as 1-local G_1, G_2 and 2-local G_2, G_4, or global, such as G_5. Information propagates through the circuit from left to right. The input is ρ^{in}. Some of the wires can be measured at the end of the diagram to produce real values o_2 and o_1. It is enough to demand that the measurements are performed in some pre-set computational basis. However, it might be reasonable to assume that the measurement basis for each wire can be different. Moreover, the basis may depend on the outcomes of previous measurements. For example, the measurement basis for the second wire might depend on the outcome of the measurement of the first wire. The output of the circuit is a combination of the measured values o_1, o_2 and a possibly entangled state ρ^{out}.

Sometimes it is easier to perform different measurements than to perform reversible operations. In this case, some gates can be performed by measurements that are conditioned on earlier measurement outcomes. If a part of the input is a so-called cluster state, any quantum computation can be performed via measurements only [103, 481, 482].

It turns out that a very small number of local gates is sufficient to represent any operation, see Subsection 2.5.3. As few building blocks describe simple operations, it is often easier to program digital rather than analog devices. Finally, in digital devices the information contained in one elementary subsystem can be redundantly encoded into several subsystems, making error correction possible, see Subsection 2.5.5.

Unfortunately, digital quantum devices also have drawbacks. For one, it is much harder to build a digital, rather than analog, quantum system that is complex enough to shed light on previously unknown phenomena. Current analog quantum simulators can prepare macroscopically entangled states of 10^4 particles [26], where digital devices coherent enough have not yet reached a 100 qubits [35, 401] (with the record for a state that has a reasonable fidelity with GHZ being 27 [400] qubits and 24 [459] qubits on a machine that can fit in existing data center racks).

2.5.3. Universality

It turns out that a small number of digital operations is sufficient to perform any computation. Thus, building many-qubit devices that are tuned to perform this small but *universal* [162] instruction set is a viable way to explore physics of arbitrary complication and richness.

Definition 2.5.1 (Universal instruction set). *We call an instruction set $\mathcal{G}$ universal, if any unitary U can be approximated arbitrary well using only operations from $\mathcal{G}$.*

It is of high interest to understand what are the simplest universal instruction sets. A detailed derivation of the universality of sets of gates discussed in this subsection can be found in [424].

**Theorem 2.5.2.** *A universal instruction set can be composed of operations that act non-trivially on at most two levels.*

Proof. Any element $U \in SU(d)$ can be written as

$$U = e^{it_1 G_1} \cdot \ldots e^{it_{n^2-1} G_{n^2-1}}, \tag{2.54}$$

where $t_j \in \mathbb{R}$ and G_j are generators of the $\mathfrak{su}(d)$ Lie algebra, $j = 1, \ldots n^2 - 1$. The generators G_j can be chosen as a set of diagonal matrices and matrices of the form $\delta_{mn} + \delta_{nm}$ or $i\delta_{mn} - i\delta_{nm}$, $d \geq m > n \geq 1$. If G_j is diagonal, $e^{it_j G_j}$ can be written a composition of one-level operations, otherwise $e^{it_j G_j}$ is a two-level operation. $\square$

A general two-level operation can be global. However, a universal instruction set can be composed only of one- and two-qubit operations.

Definition 2.5.3 (Controlled gates). *A controlled-U is a linear operation on two subsystems—a control qubit and a target subsystem. If the control qubit is in the state $|1\rangle \equiv |\uparrow\rangle$, U is applied to the target subsystem, otherwise target is left alone; that is, for basis states $|c\rangle|t\rangle \to |c\rangle U^c|t\rangle$. The controlled-U is denoted by a circuit* $\begin{smallmatrix} \bullet \\ \boxed{U} \end{smallmatrix}$.

In particular, the action controlled-NOT or CNOT on basis states is $|c\rangle|t\rangle \to |c\rangle|c \oplus t\rangle$, where $\oplus$ is addition modulo 2, and is denoted by a circuit $\begin{smallmatrix} \bullet \\ \oplus \end{smallmatrix}$.

Theorem 2.5.4. *One-qubit operations and CNOT gate is a universal instruction set.*

This theorem can be proved by observing that any two levels can be exchanged via one-qubit and CNOT gates (see [424, Subsection 4.5.2]). Thus, any operation on levels l_1 and l_2 can be performed by exchanging l_1 with the level that forms a qubit together with l_2, doing the desired two-level operation on the levels of one qubit and undoing the exchange; Theorem 2.5.3 concludes the proof. It should be noted that the choice of CNOT is somewhat arbitrary, as almost any other two-qubit gate would still yield universality [368, 164].

Any one-qubit gate can be generated by very few operations. In fact, universal instruction set can be discrete.

Definition 2.5.5 (Standard one-qubit gates). *The Hadamard gate H, phase gate S and $\pi/8$ gate T are*

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \tag{2.55}$$

Theorem 2.5.6 (Standard set of univesal gates). *An instruction set consisting of the Hadamard, phase, $\pi/8$ and CNOT gates is universal.*

A universal instruction set has to contain a two-qubit gate, e.g. CNOT, as one-qubit operations map product states to product states. However, it is sufficient to work with two-qubit operations that act only on adjacent qubits according to some ordering. Indeed, a two-qubit operation on non-adjacent qubits can be performed by SWAP-ing one of the qubits in adjacency to the other one and then SWAP-ing it back.

Definition 2.5.7 (SWAP). *A linear operation $SWAP : \mathcal{H}_1 \otimes \mathcal{H}_2 \to \mathcal{H}_2 \otimes \mathcal{H}_1$ is defined on basis states as $\forall\ |\phi\rangle \in \mathcal{H}_1, |\psi\rangle \in \mathcal{H}_2\quad SWAP|\phi\rangle|\psi\rangle = |\psi\rangle|\phi\rangle$. The SWAP is represented by a circuit* $\times\!\!\!\times$ *.*

A SWAP gate can be constructed from 3 CNOT gates.

Now that we have a universal instruction set, how do we approximate an n-qubit unitary of interest U with accuracy ϵ? It can be a hard task, as simple counting of number of states versus number of operations shows that there are unitary transformations that take $\Omega\left(\frac{2^n \log^c(1/\epsilon)}{\log(n)}\right)$ operations to approximate, see e.g. [424, Subsection 4.5.4]. Nevertheless, there is an efficient way to generate such an approximation [149].

Theorem 2.5.8 (Solovay-Kitaev, [313, 149]). *Let $\mathcal{G}$ be an instruction set for $SU(d)$, and let a desired accuracy $\epsilon > 0$ be given. There is a constant $c < 4$ such that for any $U \in SU(d)$ there exists a finite sequence S of gates from $\mathcal{G}$ of length $O\left(\log^c(1/\epsilon)\right)$ such that $\sup_{\|\psi\|=1} \|(U - S)\psi\| < \epsilon$.*

There is no algorithm such that $c < 1$. [255]

Approximating sequence S can be found on a classical computer (using an algorithm in [149]) with a running time $O\left(\log^e(1/\epsilon)\right)$.

Results discussed in this subsection tell us that any unitary transformation can be approximated using, possibly many, instructions chosen from a universal set. A universal instruction set can consist of just 4 elements acting on at most 2 qubits.

A transformation can be experimentally relevant if it requires a small, e.g. polynomial or logarithmic, number of instructions. Unfortunately, the results discussed in this section do not tell us what transformations are experimentally relevant.

2.5.4. DiVincenzo criteria

We are now in a position to state a minimal set of conditions necessary for an experimental realization of a scalable quantum computer. [169]

1. A scalable physical system with well-characterized qubits.

2. An ability to initialize the state of the qubits to a simple fiducial state, such as $|0\ldots0\rangle$.

3. Long relevant decoherence times, much longer than the gate operation time.

4. A universal instruction set.

5. A qubit-specific measurement capability.

Meeting these criteria suffice for quantum computation. However, it is highly beneficial to be able to exchange information between different devices. In order to perform *quantum communication* (see Subsection 2.7.5), two more criteria should be met.

1. An ability to interconvert stationary and flying qubits.

2. An ability to faithfully transmit flying qubits between specified locations.

2.5.5. Quantum error correction (QEC)

Long coherence times necessary to build a quantum computer (see Subsection 2.5.4) and reliable transmission of quantum information (see Section 2.7) can be achieved by redundantly encoding information into bigger Hilbert space, using e.g. more qubits, see Fig. 2.24. The encoding ensures that if an error on a small fraction of the qubits occurs, the decoder can correct, or at least detect, the error. The information to be transmitted over a noisy channel is encoded by a sender and decoded by a receiver; the information manipulated by a quantum device can be shielded against degradation by performing gates on encoded data [545] and running an error correction routine with an appropriate frequency.
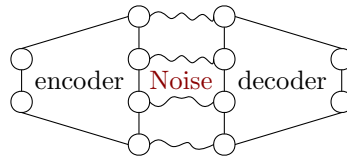
Figure 2.24.: Errors can be detected and corrected by redundant encoding into bigger Hilbert space. This example shows an encoding of two qubits into four.

Depending on a situation, either detection or correction of errors is desirable. Error correcting codes conveniently allow transmission or computation to go uninterrupted by noise. However, error detecting codes tend to require fewer resources. If the error probability is small, it is often more economical to redo the transmission or computation if the error is detected rather than pay the overhead for error correction.

Error detection codes are exemplified by bytes consisting of information bits and one extra parity bit. If the party of information bits does not equal the parity bit, an error has occurred. However, for a typical 8 information bits in a byte, this encoding is insufficient to correct the error.

Classically, the simplest kind of encoding is repeating one bit of information n times. Even if $\left\lfloor \frac{n-1}{2} \right\rfloor$ bits are corrupted, the information can still be decoded by the majority rule. See e.g. [377] for a deeper discussion on classical error detection and correction. However, such repetition error correcting schemes cannot work for quantum data, which cannot be broadcasted and, thus, cloned.

Theorem 2.5.9 (No broadcasting, [48])**.** *Let* $\rho_s \in \mathcal{T}(\mathcal{H})$, $s \in \{0, 1\}$ *and* $\tilde{\rho}_s \in \mathcal{T}(\mathcal{H} \otimes \mathcal{H})$ *be quantum states such that* $\mathrm{Tr}_1(\tilde{\rho}_s) = \mathrm{Tr}_2(\tilde{\rho}_s) = \rho_s$. *A quantum channel* $\rho_s \to \tilde{\rho}_s$ *exists if and only*

if ρ_0 and ρ_1 commute.

Corollary 2.5.10 (No cloning, [440, 641]). *There is no unitary operator U acting on $\mathcal{H} \otimes \mathcal{H}$ and such that for any $|\phi\rangle$ and some $|e\rangle \in \mathcal{H}$ and real $\alpha(\phi, e)$*

$$U|\phi\rangle \otimes |e\rangle = e^{i\alpha(\phi,e)}|\phi\rangle \otimes |\phi\rangle. \tag{2.56}$$

There is a bound on how well an approximate cloning can be performed; notably, the error vanishes in the classical limit.

Theorem 2.5.11 (Approximate N to M cloning, [304]). *For any cloning map of N to M copies $T : \mathcal{B}\left(\mathcal{H}^{\otimes N}\right) \to \mathcal{B}\left(\mathcal{H}_+^{\otimes M}\right)$ of approximate cloning, the error in one-particle expectation values*

$$\Delta(T) = \sup_{\psi, a \leq \mathbb{1} \in \mathcal{B}(\mathcal{H}), k} \left| \langle\psi|T\left(\mathbb{1}^{\otimes(k-1)} \otimes a \otimes \mathbb{1}^{\otimes(M-k)}\right)|\psi\rangle - \langle\psi|a|\psi\rangle \right| \tag{2.57}$$

is bounded from below as

$$\Delta \geq \frac{d-1}{d}\left|1 - \frac{N}{N+d}\frac{M+d}{M}\right|, \tag{2.58}$$

where $d = \dim(\mathcal{H})$ and $\mathcal{H}_+^{\otimes M}$ is a space spanned by $|\phi\rangle^{\otimes M}$, $|\phi\rangle \in \mathcal{H}$. The bound is tight.

Despite no-cloning theorem, quantum error correction is possible. Consider a set of messages $\{i\}$ and a set $\mathcal{C} = \{c_i\}$ of sufficiently distinguishable states in a large Hilbert space $\mathcal{H}_c$. If noise $\mathcal{E}$ maps different c_i to disjoint regions $\mathcal{C}_i$ of $\mathcal{H}_c$, then the encoding from *physical* to *logical* states $i \to c_i$ renders any error induced by $\mathcal{E}$ correctable, see Fig. 2.25.

The noise present in an experiment typically cannot be canceled exactly. However, it usually can be approximated by a correctable noise $\mathcal{E}$. It is important to characterise how stable are the logical states. Note that the noise acting on the logical states may be significantly different from the noise acting on the physical states [302, 435].
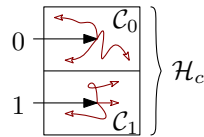
Figure 2.25.: Errors can be corrected if information (0 and 1) is encoded (black arrows with filled heads) into states that are mapped by noise (dark red arrows with empty heads) to disjoint regions ($\mathcal{C}_0$ and $\mathcal{C}_1$) of a large Hilbert space $\mathcal{H}_c$.

Definition 2.5.12 (Quantum error correcting code (QECC)). *Let $\mathcal{H}_c$ be a Hilbert space and $\mathcal{C}$ a subspace of $\mathcal{H}_c$. $\mathcal{C}$ is a QECC correcting the set of errors $\{E_i\}$ iff $\forall |\psi\rangle \in \mathcal{C}$ there is a quantum channel R such that $R \circ E_i|\psi\rangle = |\psi\rangle$.*

The historically first and simplest example of QECC is a three-qubit bit flip code [449], see Fig. 2.26. It encodes one qubit into three and corrects a set of errors $\{E_i = \pi_i \mathbb{1}^{\otimes 2} \otimes \sigma^x \pi_i^\dagger\}$, where π_i is a set of permutations and σ^x is a spin-flip operator.
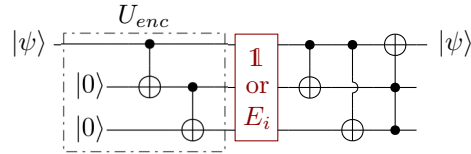
Figure 2.26.: The tree-qubit bit flip code encodes $|\psi\rangle = \alpha|0\rangle + |1\rangle$ into $U_{enc}|\psi\rangle = \alpha|000\rangle + |111\rangle$ and can correct any error $E_i = \pi_i \mathbb{1}^{\otimes 2} \otimes \sigma^x \pi_i^\dagger$.

The first QECC that corrects any one-qubit error was discovered in [544]. It encodes one qubit into 9 via

$$|0\rangle \to |c_0\rangle = (|000\rangle + |111\rangle)^{\otimes 3}, \quad |1\rangle \to |c_1\rangle = (|000\rangle - |111\rangle)^{\otimes 3}. \tag{2.59}$$

Later, encoding into 5 qubits—the minimal number required to correct any on qubit error, was constructed [348, 71].

The number of qubits needed to correct a given set of errors is bounded from below by the Knill-Laflamme bound [319].

Note that a large Hilbert space needed for QEC can be in principle realized using as little as one particle. Information can be encoded in a big subspace of the infinite-dimensional Hilbert space of a single oscillator. First considered in [130], such QECC can use, for example, symmetric and anti-symmetric superposition of coherent states [134], or equal-weight superposition of squeezed states [233].

Since the inception of QEC, a number of diverse and powerful families of codes were discovered. the most widely-used class of QECC are stabilizer codes [229], with prominent subclasses of CSS [110, 558], $GF(4)$ [109] and topological codes [314, 315, 161, 84].

Automated search of QEC codes using classical computers was pioneered by [489]. Today this is a well-established technique; as a characterization of a noise channel is usually a hard task, machine learning (see Section 2.6) is a popular method for this task [211], especially widely used for the search of decoders [576, 332, 589, 43, 367, 385, 44, 99, 590, 119, 30, 417].

So far we have not discussed that the gates used to encode and decode information are can be faulty. In a poorly designed error correction protocol, a problem with a single gate can spread the error throughout the circuit. Luckily, it is possible to keep the error propagation in check and build a device that works efficiently when its elementary components are imperfect. Such *fault-tolerant* quantum devices were first proposed in [545].

Definition 2.5.13 (Fault-tolerance)**.** *Suppose individual quantum gates produce errors in their output independently with probability p. For a given error-correcting code that corrects one error, an implementation of a gate acting directly on encoded states is considered to be fault-tolerant if the probability that the implementation introduces an unrecoverable error is bounded above by cp^2 for some constant c.*

Example 2.5.14. Let us consider a CNOT operation that acts on 3 encoded qubits operation $\text{CNOT}_{enc} : |i\rangle^{\otimes 3}|j\rangle^{\otimes 3} \to |i\rangle^{\otimes 3}|i \oplus j\rangle^{\otimes 3}$. It can be realized in several non-equivalent ways:
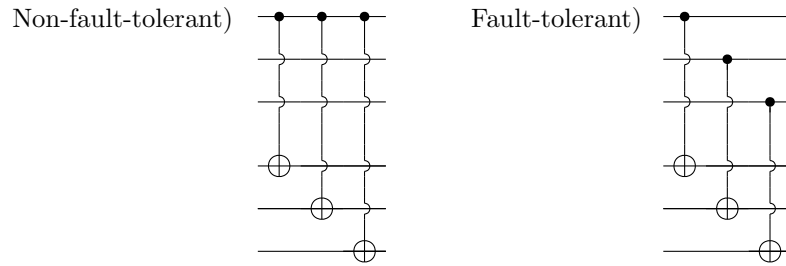
If an error occurs in the first qubit of a non-fault-tolerant implementation, it propagates to every controlled qubit. For the fault-tolerant implementation, however, this error only affects one controlled qubit and information can be successfully decoded.

Every stabilizer code can be implemented fault-tolerantly [230].

QEC may bring a hefty overhead in terms of extra qubits and run time over unencoded protocol. Thus, fully fault-tolerant computation is viable only if the information protection overcompensates the extra opportunities for errors due to a larger number of operations, decreasing the net logical error rate. This is possible if elementary steps in computation can be performed with the error rate smaller than the *threshold* $1/c$. Once the noise strength under the threshold was achieved, it can further be brought down using *concatenated codes* [321, 314, 16, 178]. Indeed, let an encoding $\mathcal{F}$ of one qubit into n decrease error rate r-fold. Then k concatenations $\mathcal{F} \circ \mathcal{F} \cdots \circ \mathcal{F}$ (see Fig. 2.27) is an encoding of 1 qubit to d^k that decreases error rate r^k fold.
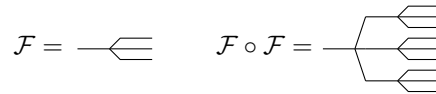
Figure 2.27.: 2 concatenations of $\mathcal{F}$ encoding 1 qubit into 3.

See e.g. [231, 232, 466] and [300, Chapter 10] for further discussion. Currently experimentally investigated codes are estimated to have a threshold of around $1/100$ [212, 46]

Experimental proof-of-principle demonstration of QEC protocol was achieved using various physical platforms, e.g. NMR [137, 320], ions [127, 517], NV centers [605, 569, 141], photons [458, 32], superconductors [487, 301, 136, 495]. Impressively, 7 topological [425, 27] and GKP [209, 157, 111] codes were implemented on ion-trap and superconducting devices. Lifetimes of qubits were prolonged via QEC [301, 428, 27]. Moreover, quantum devices themselves were used to optimize encodings [409]. Nevertheless, there are still enough hurdles on the way to fully fault-tolerant quantum computer [76]. Available number of qubits and performance of experimental building blocks, such as gates and read-out, is not yet adequate to achieve the breakeven point of beneficial QEC. Luckily, a lot of effort is directed to both enhancing existing platforms and developing new architectures to achieve this goal, see e.g [92, 251, 352].

2.5.6. Noisy intermediate-scale quantum (NISQ) devices

Current quantum computers have less than 100 noisy qubits. Fault-tolerance is not yet achievable with currently available experimental building blocks. Even worse, quantum computers are

expensive and often each qubit requires extensive calibration and certification by highly-skilled workers (see e.g. [35]).

On the other hand, classical computers are abundant, fault-tolerant, well-connected over the internet and cost per bit and operation is incredibly low. Why use quantum?

Definition 2.5.15 (Quantum advantage (or supremacy [467])). *A circuit with run time T_q exhibits a quantum advantage if the time T_c needed to simulate it on any existing classical computer of comparable cost satisfies $T_c \gg T_q$.*

Small-scale devices—computers with up to several dozen qubits, can not demonstrate quantum advantage, as any computation they perform can be accurately modelled on mass-produced classical computers. There are well-known tasks where quantum computers show exponential speed-up over classical ones (see e.g. [300]). Thus, with a lot of qubits and long coherence times quantum advantage is easily achievable. **But can modern, noisy intermediate-scale (from around 50 to several hundred) quantum (NISQ, [469]) be useful?**

The answer is believed to be yes.

First of all, a full description of an arbitrary intermediate-scale wave function can not fit into a memory of any currently conceivable classical computer. This makes even the simplest tasks for a universal quantum computer, such as producing a sufficiently generic and pure random state, classically hard (see e.g [91] for relevant bounds). This is a basis for current claims of quantum advantage, such as [35] (while not as dramatic as initially stated, the claim still holds if more efficient known classical algorithms are taken into account [443, 444, 436]) and [651]. States generated in these claims are not of the highest practical importance, yet they prove the potential utility of current quantum devices.

Second, memory is not the only bottleneck for classical devices. Even shallow circuits can reduce run time for computation.

Theorem 2.5.16 (Quantum advantage with shallow circuits [96]). *There are computational problems that can be solved by a constant-depth quantum circuit consisting of one- and two-qubit gates but require at least a logarithmic-depth classical probabilistic circuit with bounded fan-in gates.*

The classical circuit of depth asymptotically growing at least as fast $\frac{\log(n)}{\log(\log(n))}$ is requires if qubits are subject to local stochastic noise with strength below a threshold value independent of circuit width n [97].

Once again, the main use of problems studied in [96, 97] is to show the separation in run time between quantum and classical gates; hopefully, the class of problems where the separation exist also includes practically relevant tasks.

Third, there are a lot of sources of quantum data (see Section 2.7) that could benefit from some post-processing (see e.g. Chapters 3 and 5). Turning quantum information into classical is a resource-intensive procedure. For example, full tomography of a state requires exponential in the dimension of the Hilbert space number of measurements. Using quantum devices for post-processing can be a very natural and economical approach.

Finally, a quantum advantage is an overly strong requirement. Maybe there are efficient classical algorithms that can simulate this particular quantum circuit, but discovering these algorithms may be harder than building a sufficiently powerful quantum device. This is often the case for the simulation of quantum many-body systems with a straightforward implementation on a particular quantum computing architecture [310, 311].

We can see that there are many reasons why NISQ devices can be very useful, and it is important to design algorithms that exploit their potential to the fullest. See e.g. [78] for a review of available NISQ algorithms.

Some of the leading architectures for NISQ devices are superconducting qubits [316], trapped ions [104] and photons [556, 208].

2.5.7. Designing interactions and dissipation.

Instead of focusing on representing local gates, one can focus on designing interaction. While the two approaches are polynomially equivalent [18], there are several benefits in concentrating on interactions.

Evolution with local interaction allows targeting states inaccessible by shallow local circuits. If a system is evolving with a local Hamiltonian $H = \sum_i h_i$, the evolution operator e^{-iHt} is, in general, a non-local gate. Shallow circuits of local gates can approximate short time evolution via e.g. Suzuki-Trotter expansion $e^{-itH} = \prod_i e^{-ith} + O(t^2)$, yet are not conducive for exploration of long-term dynamics. In contrast, even noisy devices with tunable interaction can explore e.g. steady states of evolution operators.

Ground states of a large class of Hamiltonians $\{H_i\}$ can be obtained in the framework of adiabatic quantum computation [197]. The system can be initialized as a ground state of some easy to realize Hamiltonian, say spin-1/2 particles coupled with strength b to the external magnetic field $H_0 = b \sum_i \sigma_i^z$, by keeping the temperature much below the spectral gap $2b$. The system is then evolved for time T with time-dependant $H(t) = [1 - f(t)] H_0 + f(t)H_i$, such that $f(0) = 0$ and $f(T) = 1$. Due to the adiabatic theorem, the system will remain in the ground state if $T = O\left(\frac{1}{g_{\min}^2}\right)$, where $g_{\min}$ is the minimal spectral gap of $H(t)$. Fortunately, answers to many interesting problems can be encoded into ground states of relatively simple Hamiltonians. For example,

$$H_{ZZXX} = \sum_i (b_i \sigma_i^z + \Delta_i \sigma_i^x) + \sum_{i<j} (J_{ij} \sigma_i^z \sigma_j^z + K_{ij} \sigma_i^x \sigma_j^x) \tag{2.60}$$

is QMA-complete [80]. QMA (e.g. [17]) is the quantum analog of NP—if an answer to a decision problem is YES, the proof is provided in a form of a state that with high probability convinces a polynomial-time quantum verifier. The proof can always be provided as a ground state of H_{ZZXX} with appropriately chosen $\{h_i, \Delta_i, J_{ij}, K_{ij}\}$ and verified by measuring $\langle H_{ZZXX} \rangle$.

A device with tunable interactions may be easier to implement than a set of high-fidelity universal gates. There are many systems that sport interactions alike present in H_{ZZXX}, from flux qubits [430] and ions [75, 490] to polar molecules [395]. It is not a coincidence that the first commercially available digital quantum device was a quantum adiabatic computer [631].

Not every ground state can be efficiently prepared adiabatically—if the minimal gap vanishes, one has to compromise between extremely long run time and significant overlaps of the prepared state with excited states.

Quantum systems are usually coupled to their surrounding environment. Such couplings are needed to control the system and to measure observables. This may lead to decoherence, dissipation and consequential washing out of interesting quantum effects. Thus, significant efforts have been made to isolate quantum systems and dissipation is typically treated as an adversary.

However, many exciting systems are open and deserving of simulation. Moreover, whereas it is usually worthwhile to shield the system from noise, dissipation can be a fully-fledged resource. While closed systems can efficiently simulate open ones [317], a controlled environment can improve the robustness of the experiment by deriving the system to the desired stable space and suppressing unwanted overlaps.

There are several ways to introduce controlled dissipation into a quantum simulation. In closed-system simulators, part of the system can be treated as the environment, see e.g. [518]. A different adjustable physical platform coupled to the system may play a role of the reservoir, see e.g. [522, 100, 253, 312, 477, 331]. Alternatively, not the different platform itself, but rather the coupling between it and the system undergoing state preparation can be tuned, see e.g. [490, 167]. Measurements may be used to steer the system in a desired direction, see e.g. [344, 345, 54, 610, 416, 192].

Different physical platforms are best suited for each of the different controlled dissipation schemes. Some of them are more conducive to gate-based quantum computation, while continuous evolution with adjustable interactions is more appropriate for others, see e.g. [8]. It is important to understand what useful channels and states are realizable via a given *dissipative engineering* [167] scheme.

Universal gate-based open system simulators were experimentally demonstrated using ions, see [518, 407]. This is a promising platform for quantum machine learning, see e.g. [56, 502] and Subsection 2.8.2; we assume access to a universal quantum channel simulator in Chapters 4 and 5. We pay special attention to the realizability of our protocols on NISQ devices.

Highly-entangled states have been prepared using dissipation via adjustable interactions; some of the platforms used are superconducting devices [541, 246, 207], ions [490, 477] and atoms [335], including Rydberg ones [355]. There is also a number of theoretical proposals for experiments, e.g. [167, 491, 454, 475], including for schemes that yield universal quantum simulation, e.g. [637]. See e.g. [408] for a review.

The simplest preparation scheme involves designing local interactions and dissipation to produce steady states. It is a very robust preparation scheme—even if the evolution operator have substantial errors, the produced state can be closed to the desired one; see [106, Theorem 4] for the ergodic case, i.e. stable space is one-dimensional, and [142, Theorem 7] for the case of polynomially decaying errors in rapidly-mixing Lindbladians (compare with orthogonality catastrophe for non-dissipative Fermi gases [29, 220]). Local dissipation and driving can also rectify currents [645], activate approximate conservation laws [351] and improve sensors [476, 478]. Sta-

bility and simplicity make this protocol a conducive candidate for quantum memories [441]. There are large classes of states that can be prepared this way, such as stabilizer spaces [334] and, importantly, any state with parent Hamiltonian [574] (see Subsection 2.4.3). In particular, an MPS [598] (see Subsection 2.4.2) can be prepared via local interactions and dissipation.

Nevertheless, there are still large gaps in understanding what stable spaces local dissipative dynamics can have. Discovery of novel systems that can benefit from dissipative engineering is an active research field [20, 156, 508, 271] with a number of open questions. One such question is related to the generalization of MPS to mixed states—MPDOs, see Subsection 2.4.4. MPDOs are the basis of various variational methods for 1d open systems. While we know that MPS have parent Hamiltonians, see Subsection 2.4.3, we had not yet know if parent Lindbladians exist for MPDOs, what is the sufficient locality and how to construct them. We construct a parent k-local quantum channel and Lindbladian for a given space of MPDOs or prove that such evolution does not exist in Chapter 3.

2.6. Machine learning

In this section we discuss how algorithms can improve automatically through experience. Based on this discussion, we introduce quantum neural networks in Section 2.8—a workhorse for the denoising techniques described in Chapters 4 and 5.

With the ever increasing complexity of systems that our society deals with, the ab-initio understanding of important processes remains a distant dream. However, useful behaviours can be learned by experience and without in-detail understanding. As our ability to gather, store, generate and process data has rapidly progressed, the algorithms that are able to improve automatically by the use of data—*machine learning* (ML)—have become much more powerful. One of the most popular ML techniques are neural networks (NNs), which have found numerous applications, from self-driving cars to drug discovery (see e. g. [640, 423, 227]).

ML algorithm typically contains the *training data set*—a number of correct input-output pairs of the desired map $\{x_i, y_i\}_{i=1}^L \in (X \times Y)^L$, a *variational class* of maps $f_v : X \to Y$ parameterized by a vector v and a *cost function* C. The variational parameters in v are optimized such that $C(\{x_i, y_i\}_{i=1}^L) = \frac{1}{L} \sum_{i=1}^L d(f_v(x_i), y_i)$ reaches a (local) minimum. Here, d is an appropriate distance measure. Typically the optimization employs some variant of the gradient descent algorithm (see e.g. [505]).

An optimized f_v should generalize well to *validation set*—previously unseen data coming from the same source as the training set.

The richness of the variational class has to be sufficient to capture significant features of the data; otherwise we say that the ML approach is suffering from *underfitting*. Underfitting can often be diagnosed by observing the cost function after training—if it remains significantly above levels that can be attributed to noise.

While the cost function can reach the absolute minimum on the training set, it can be arbitrarily large on the validation set. There are several reasons why this might happen; one of them is a

poor quality of the training set, another one is too expressive variational class. Unlike many optimization algorithms, ML can also suffer from *overfitting*—a learning outcome that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably, see Fig. 2.28. The problem of overfitting is especially acute for noisy training data, as overparametrized variational optimization can learn to reproduce the features of noise rather than the signal.
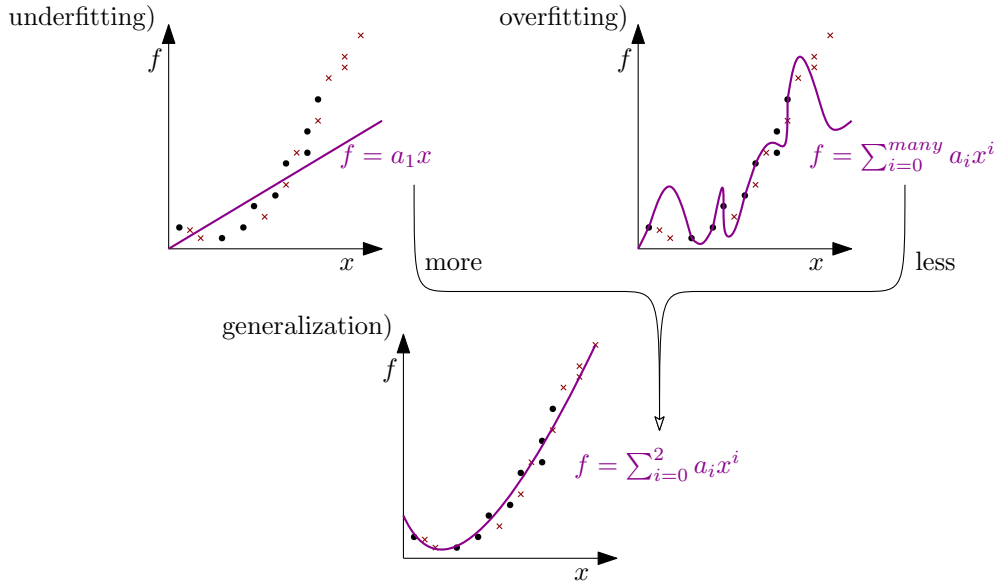
Figure 2.28.: A result of training $f \in \{f_{\vec{a}}\}$ (violet) for variational classes of various expressive powers and the same data. Training set is represented by black disks, validation set—by dark red crosses. A class of linear functions $f_{\vec{a}} = a_1 x$ parametrized by a single parameter underfitts and fails to reproduce even the structure of the training data. More parameters are needed for generalization. Polynomials of arbitrary degree $f_{\vec{a}} = \sum_i a_i x^i$ overfits. While achieving arguably perfect performance on the training set, the errors for validation set can be arbitrarily large. Less parameters are needed for generalization. Quadratic polynomials generalize well despite noise in the data.

Prior information is necessary for successful learning. Indeed, without any assumptions about underlying structure of the available data, any output on unseen data is no better than a random guess.

Theorem 2.6.1 (No free lunch, see e.g. Theorem 1.6 in [640])**.** *For the training data $S = \{x_i, g(x_i)\}_{i=1}^{L} \in (X \times Y)^L$, let $L < |X| < \infty$, $|Y| < \infty$. For any function to be learned $g : X \to Y$ and a variational function $f_S : X \to Y$ define $C_g(f_S) = \mathbb{P}_x [f_S(x) \neq g(x)]$, where $\mathbb{P}_x$ is the probability with respect to a uniform distribution of x over X. Then expected cost averaged*

uniformly over all $g \in Y^X$ and training data sets fulfills

$$\mathbb{E}_g\left[\mathbb{E}_S\left[C_g(f_S)\right]\right] \geq \left(1 - \frac{1}{|Y|}\right)\left(1 - \frac{L}{|X|}\right). \tag{2.61}$$

A quantum generalization of this theorem is available in [460].

The no free lunch theorem implies that there is no order among learning algorithms. If one algorithm beats another on some learning tasks, the converse has to hold on other tasks. This has to be put into perspective, however, since not all learning is equally relevant.

One approach to constructing learning algorithms that ace real-life tasks is to gain inspiration from networks of neurons that complex living organisms use: just like (artificial) neural networks.

2.6.1. Learning scenarios

Depending on the data, one can distinguish different learning scenarios for ML algorithms.

If the training data contains the desired outputs of the algorithm, the learning is called *supervised*. For example, for the task of image recognition the training data can be composed of images and corresponding labels [243, 633].

Sometimes, only partial knowledge about the desired output of the algorithm is available. For example, the full strategy of a game might be unknown while it is possible to assign a score to every set played. This is an example of semi-supervised or reinforced learning [551, 101].

If nothing is known precisely about the algorithm's desired outputs for the training inputs, the learning is called *unsupervised* or *self-supervised*. Understanding of how systems can adapt without supervision started to emerge as early as 1949 [261]. Applications of such learning include very relevant for this thesis data denoising without access to a noiseless source, see Section 4.3.

2.6.2. Neural networks (NN)

Artificial neural networks, usually simply called neural networks (NNs), are a family of parametrized maps vaguely inspired by the nervous system of animals. Just like the biological prototype, NNs are composed of relatively simple units—*neurons*, that are able to interact with each other. Neurons are usually denoted as vertices of a network and interactions between them—as connections in it.

A neuron in an artificial NN is a parameterized map. The outputs of a set of neurons—a *layer*—are fed into the next layer. If a layer n gets all its inputs from layers $k < n$, the network is called *feed forward* (FF). The input $x \in X$ of the NN is the first layer, and the output $f_v(x)$—the last. The number of layers is the *depth* of a NN, and the maximal number of neurons per layer—its *width*. The geometry of the neuronal interconnections—the *topology* of a NN—and the choice of neurons determine the variational class given by the NN.

Output of a neuron, or *activation function*, p is usually chosen to be an affine function followed by a non-linearity s, $p(\vec{x}; \vec{w}, b) = s(\vec{w} \cdot \vec{x} + b)$. Parameters to be learned are *weights* $\vec{w}$ and bias b. Some popular choices for the non-linearity are (for more examples, see e.g. TensorFlow documentation [571] or [630]):

- Rectifier $s(z) = \max(z, 0)$. Is often used as a default in modern NN, the activation function is called rectified linear unit, or ReLU; see [247, 248] for initial biology-inspired motivation and e.g. [225] for comparison in with other activation functions in deep artificial NNs.

- Softplus $s(z) = \log(1 + e^z)$

- Sigmoid functions, such as:

  - Step $s(z) = \text{sgn}(z)$. Arguably historically the earliest [499], yet hard to optimize with gradient descent (see Subsection 2.6.3).

  - Logistic $s(z) = \frac{1}{1+e^{-z}}$.

  - Tanh $s(z) = \tanh(z)$.

There are activation functions that do not follow this pattern, for example

- Polynomial $p\left(\vec{x}; \{a_i^{j_1,\ldots,j_i}\}\right) = a_0 + \sum_j a_1^{j_1} x_{j_1} + \sum_{j_1,j_2} a_1^{j_1,j_2} x_{j_1,j_2} + \ldots$. Used in pioneering deep learning algorithms from late 1960s–early 1970s [285, 286].

- Radial activation functions, such as

  - Gaussian $p\left(\vec{x}; \vec{w}, b\right) = e^{\frac{-\|\vec{x} - \vec{w}\|^2}{2b^2}}$

  - Multiquadratics $p\left(\vec{x}; \vec{w}, b\right) = \sqrt{\|\vec{x} - \vec{w}\|^2 + b^2}$

With suitable neurons FFNNs are *universal*—any (sufficintly "nice") map can be represented as an FFNN. For this, it is sufficient to use one, although potentially very wide, layer. Original proof for sigmiod neurons [144] was extended to show that the only non-linearity for which wide one-layer FFNNs are not univeral is, despite its early practical success, the polynomial one [275, 456]. Width can often be to some extent traded for depth. In particular, any function $f : \mathbb{R}^n \to \mathbb{R}^m$ can be approximated up to arbitrary low error with a deep ReLU network of width $\min n + 1, m$ [252]. This bound is tight and for width below $\min(n+1, m)$ approximation power deteriorates drastically—only zero-measure sets can be approximated [372]. For more general non-affine neurons, width $n+m+2$ is sufficient for universality [308]. This can be extended to functions on more general spaces [333]. Deep networks of bounded width are also capable to represent distributions, i.e. transform uniformly distributed one-dimensional noise into an arbitrarily close approximation of any two-dimensional Lipschitz-continuous target distribution [448].

Gains from utilizing deep instead of wide network typologies can be dramatic. There are examples when adding just one more layer decreases the necessary width exponentially [185]. Additionally, for ReLU neuron approximations of sufficiently smooth functions finite-width deep networks require strictly smaller connectivity than finite-depth wide networks [447].

For more information about the modern state of the art, as well as the history of artificial neural networks, see e.g. [227, 521]

2.6.3. Gradient descent

Gradient descent is one of the simplest and most popular algorithms to perform optimization. It is far the most common way to optimize neural networks. We will review different variants of the algorithm following [505].

The objective is to minimize a cost function $C\left(\{x_i, y_i\}_{i=1}^{L}; \theta\right)$ parametrized by a model's parameters θ. Gradient descent seeks to achieve it by updating the parameters in the opposite direction of the gradient of the cost with respect to θ, $\nabla_\theta C\left(\{x_i, y_i\}_{i=1}^{L}; \theta\right)$. A related algorithm of gradient ascent seeks to maximize the cost function, it is achieved via updating in the direction of $\nabla_\theta C\left(\{x_i, y_i\}_{i=1}^{L}; \theta\right)$. For simplicity, we concentrate on the descent in this subsection.

There are, however, many ways to update the parameters once the gradient was computed.

Batch gradient descent

An update with a step size, also called *learning rate* in the opposite direction of the gradient of a cost function reads

$$\theta \rightarrow \theta - \eta \nabla_\theta C\left(\{x_i, y_i\}_{i=1}^{L}; \theta\right). \tag{2.62}$$

This update takes every data point into account. While it is crucial for a small data set, if the data is big, batch gradient descent recomputes gradients for similar examples before each parameter update; moreover, memory-intensive data sets simply can not fit into random-access memory in one batch.

Stochastic gradient descent

The redundant evaluations of the gradient can be resolved by using a single data point for each update

$$\theta \rightarrow \theta - \eta \nabla_\theta C\left(\{x_i, y_i\}; \theta\right). \tag{2.63}$$

Stochastic gradient descent can be much faster than batch, but the drawback is that frequent updates with large variance cause the cost function to fluctuate wildly.

Mini-batch gradient descent

The middle ground between batch and stochastic descent is to use some data for each update

$$\theta \rightarrow \theta - \eta \nabla_\theta C\left(\{x_i, y_i\}_{i=j}^{j+n}; \theta\right), \tag{2.64}$$

where n is the *batch size*.

As stated, gradient descent does not have memory, wastefully ignoring already computed gradients in the previous update steps. It can easily be fixed.

Momentum

Without memory, gradient descent has troubles navigating ravines, i.e. areas where cost surface curves much more steeply in one direction than in the other. This problem can be mitigated via momentum m—by accumulating updates that point in same direction and let the oscillations cancel each other out [473] (see Fig. 2.29)

$$
\begin{aligned}
m_t &= \gamma m_{t-1} + \eta \nabla_\theta C\left(\theta_{t-1}\right), \\
\theta_t &= \theta_{t-1} - m_t.
\end{aligned}
\tag{2.65}
$$
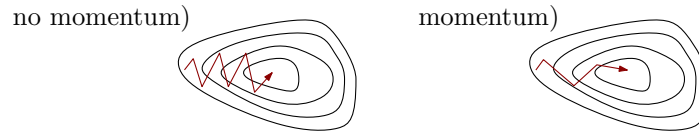
Figure 2.29.: Contour plots in the parameter space of a cost function (black) and optimization steps of a gradient descent (dark red). The extra momentum term accelerates the updates in the relevant direction and dampens oscillations.

Nesterov accelerated gradient [419]

In the system with momentum, we can use knowledge of the average direction of movement. The gradient can be calculated not at the current parameter value, but at the point where the system is expected to occur. That is, where the system would move only due to momentum

$$
\begin{aligned}
m_t &= \gamma m_{t-1} + \eta \nabla_\theta C\left(\theta_{t-1} - \gamma m_{t-1}\right), \\
\theta_t &= \theta_{t-1} - m_t.
\end{aligned}
\tag{2.66}
$$

So far we have not discussed how to choose the learning rate, η. Sadly, the optimal value for η does not just depend on data and specific model used in a machine learning algorithm, but also tend to change as optimization progresses.

RMSprop

If the gradients are large, even a small step can change the cost function substantially. If the gradients are small, however, the optimization can get stuck in barren plateaus where no significant changes happen. This can be addressed by dividing the learning rate by a weighted sum of the past cost function gradients. The currently computed gradients should have higher weights than the ones encountered long ago. This weighted sum can be obtained via an exponentially decaying average of squared gradients

$$
\begin{aligned}
g_t &= \nabla_\theta C\left(\theta_{t-1}\right), \\
E\left[g^2\right]_t &= (1-\alpha)E\left[g^2\right]_{t-1} + \alpha g_t^2, \quad \text{where } 0 < \alpha < 1, \\
\theta_t &= \theta_{t-1} - \frac{\eta}{\sqrt{E\left[g^2\right]_t + \epsilon}} g_t,
\end{aligned}
\tag{2.67}
$$

where ϵ is a small constant needed to avoid divisions by zero.

Adam [166]

Adaptable learning rate in RMSprop has a drawback—if $E\left[g^2\right]_0$ is initialized as 0, $E\left[g^2\right]_t$ is biased towards zero, especially during the initial time steps. Adam combines RMSprop with momentum and bias-corrects first and second moment estimates

$$
\begin{aligned}
g_t &= \nabla_\theta C\left(\theta_{t-1}\right), \\
m_t &= (\alpha_1 - 1)m_{t-1} + \alpha_1 g_t, \quad \text{where } 0 < \alpha_1 < 1, \\
r_t &= (1 - \alpha_2)r_{t-1} + \alpha_2 g_t^2, \quad \text{where } 0 < \alpha_2 < 1, \\
\tilde{m}_t &= \frac{m_t}{1 - \alpha_1^t}, \\
\tilde{r}_t &= \frac{r_t}{1 - \alpha_2^t}, \\
\theta_t &= \theta_{t-1} - \frac{\eta}{\sqrt{\tilde{r}_t} + \epsilon}\tilde{m}_t.
\end{aligned}
\tag{2.68}
$$

Nadam [172]

Putting all the pieces in this subsection together, Adam can be combined with Nestorov accelerated gradient

$$
\begin{aligned}
g_t &= \nabla_\theta C\left(\theta_{t-1}\right), \\
m_t &= (1 - \alpha_1)m_{t-1} + \alpha_1 g_t, \quad \text{where } 0 < \alpha_1 < 1, \\
r_t &= (1 - \alpha_2)r_{t-1} + \alpha_2 g_t^2, \quad \text{where } 0 < \alpha_2 < 1, \\
\tilde{m}_t &= \frac{m_t}{1 - \alpha_1^t}, \\
\tilde{r}_t &= \frac{r_t}{1 - \alpha_2^t}, \\
\theta_t &= \theta_{t-1} - \frac{\eta}{\sqrt{\tilde{r}_t} + \epsilon}\left(\alpha_1 \tilde{m}_t + \frac{1 - \alpha_1}{1 - \alpha_1^t}g_t\right).
\end{aligned}
\tag{2.69}
$$

Even if the optimum was reached, Adam and Nadam are prone to decaying oscillations if used with a small data set. It is more of a nuisance than a problem.

Note, that the found optimum can depend on the optimizer [426] and initial conditions [462, 427], especially for overparametrized variational classes.

One can ask if an increase in performance balances the additional complexity of Nadam. It is often so (see Fig. 2.30). Moreover, Nadam is less sensitive to the initial choice of the hyperparameters of the optimization algorithm, such as the initial learning rate. In this thesis we usually use Nadam. Our implementation of every optimizer presented in this subsection for quantum neural networks (see Section 2.8) is avaliable at [2].
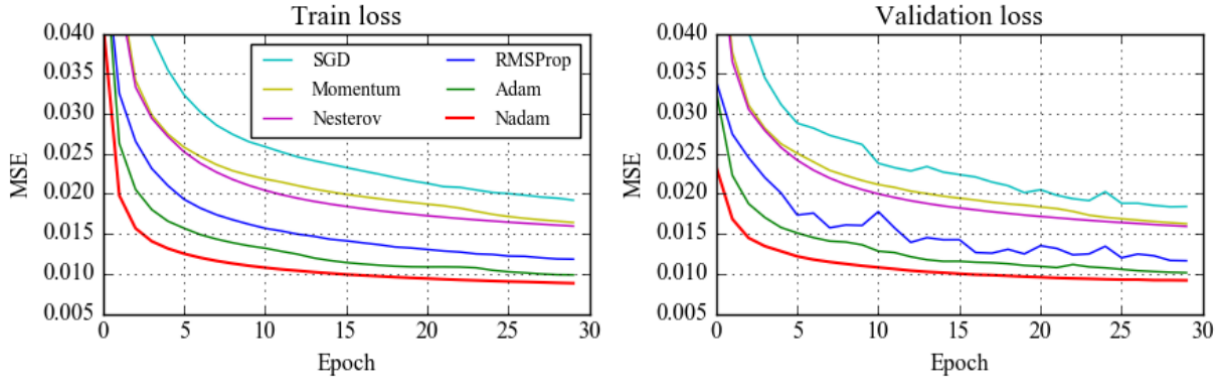
Figure 2.30.: Comparison between different optimizers: mean squared error (MSE) as cost for training and validation on the MNIST data set [172].

Backpropagation

The gradient of deep (classical) networks can be efficiently computed via a chain rule for differentiation and recursion. Let us consider a cost function C. The cost F_n of an n-layer network parameterized by θ_0 and consisting of layers $\{f_i\}$ is

$$F_n(\theta_0) = (C \circ f_n \circ \cdots \circ f_2 \circ f_1)(\theta_0). \tag{2.70}$$

For convenience, let us denote $f_{n+1} \equiv C$.

First, we compute $F_n(\theta_0)$ via *forwardpropagation*: each of the $F_{i+1}(\theta_0) \equiv (f_i \circ F_i)(\theta_0)$ is recursively evaluated for $i = 1, \ldots, n$. It is not necessary to know the value of the cost function for the gradient descent. Nevertheless, it is often advantageous, and we will reuse the results of the forwardpropagation. For the classical information, $\{F_i(\theta_0)\}$ can be stored and copied; if $F_i(\theta_0)$ is a quantum state, however, it has to be produced anew every time it is used. This is a fundamental bottleneck for optimization of quantum layered structures, see e.g. [594].

Via a chain rule,

$$(\nabla F_n)(\theta_0) = (\nabla [f_{n+1}] \circ F_{n-1})(\theta_0) \cdot (\nabla [F_{n-1}])(\theta_0). \tag{2.71}$$

We can recall $F_{n-1}(\theta_0)$ from the memory and compute $(\nabla [f_{n+1}] \circ F_{n-1})(\theta_0)$. Once it is done, the only yet unevaluated quantity on the right-hand side is $(\nabla [F_{n-1}])(\theta_0)$. It, in turn, can be evaluated via a chain rule until

$$(\nabla [F_0])(\theta_0) \equiv \nabla f_1(\theta_0) \tag{2.72}$$

is reached. The complexity of this algorithm, called *backpropagation*, scales linearly with depth of the network. It can be further optimised by taking into account how neurons in two neighbouring layers are connected to each other. For more details, see e.g. [227, Section 6.5].

2.6.4. Not-so-simple networks

So far we have only discussed how FFNN can learn on supervised data. Luckily, more general tasks can be reduced to this scenario, including unsupervised extraction of relevant features and networks with memory.

Autoencoders (AE) for unsupervised learning

Autoencoders are a prominent example of NNs that learn without supervision, see e.g. [227].

An AE is an FFNN for extracting the most relevant features from the input data. They have equal input and output layers separated by a bottleneck—AEs are designed so that they are unable to copy.

The bottleneck of an *undercomplete* AE is a layer with a smaller width than the input and output layers. The training data is a set $\{x_i, x_i\}_{i=1}^{L}$ of equal training inputs and reference outputs. In general, the desired output for x is not x itself: the bottleneck (see Fig. 2.31) should force the AE to discard irrelevant information. Since no correct reference outputs are provided, the training of AEs is unsupervised.

While in this thesis we concentrate on the undercomplete case, this is not the only way to limit the expressive power of a network. *Regularized* AEs use a cost function that encourages the model to have other properties besides the ability to copy its input to its output. Such properties can include e.g. sparsity of the representation or robustness to missing links and inputs. Training

input   bottleneck   output
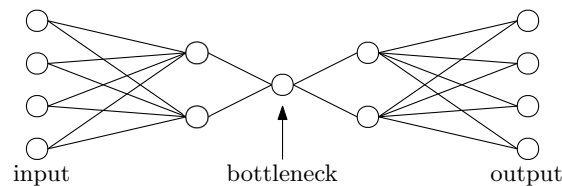
Figure 2.31.: Network architecture of an undercomplete AE. The bottleneck prevents the AE from just copying the input data to the output so that it has to extract relevant features.

AEs can be a stand-alone method. Some of their earliest applications was denoising [354] and compression [187] of data. There is often no source of noiseless labels to act as supervision, as exemplified recordings of bird songs in the wilderness [465]. Alternatively, AEs can be a data simplification subroutine in supervised learning. For example, AEs have been trained as a step in gesture generation [341, 342, 343] to reduce the dimensionality, and thus—complexity, of the learning task and to reduce redundancy in the training data.

Recurrent neural networks (RNN) for learning with memory

Networks with memory are not feed forward. Memory requires sequential structure; some information about previous inputs should be shared with the parts of the model responsible for

later outputs. This can be achieved via an information storage unit that different parts of the model have access to. However, this division is not necessary. We can always treat memory and computing units as a single entity that recalls information from itself. On the other hand, elements of the model can be responsible for both remembering and processing data, so there are systems that can remember, but without the dedicated information storage. In other words, some of the outputs of a general model with memory are its inputs, see Fig 2.32.
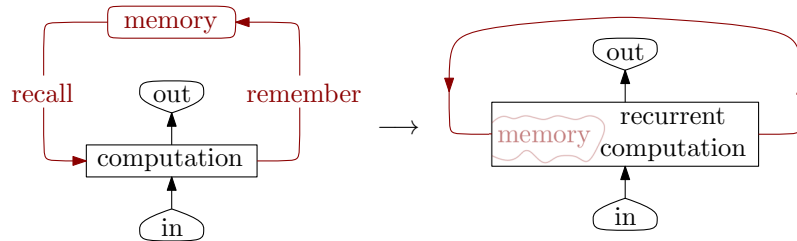
Figure 2.32.: Any computation with access to memory (dark red) can be represented by a system that treats some of its outputs as inputs at later times.

Recurrent neural networks (RNNs) are a family of NNs for processing sequential data. As an RNN shares parameters used for processing data at various positions in a sequence, the model can be applied to examples of different length and generalize across them. RNNs can be *unfolded*, or *unrolled*, transforming a recursive structure into repetitive one (noted already in [396]). Instead of passing information to itself, we can think that network passes information to its copy see and Fig. 2.33. Typically different copies of the network in the unrolled form are labeled by time
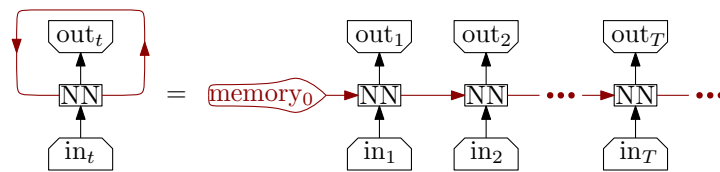
Figure 2.33.: Recurrence (dark red) is equivalent to a network passing some of the outputs to a copy of itself. The sequential data is labeled by a time index t. As the copies share parameters, the length T of training data sequences may vary. The memory can be initialized at the beginning of computation ($memory_0$).

index. As unrolled RNN has a layered structure without loops, computing the gradient can be done by a straightforward application of the backpropagation (see Subsection 2.6.3) through time [507].

Training of RNNs was demonstrated in the 1980s, e.g. [274, 507, 295, 616, 636], yet the performance was not yet sufficient for a widespread adoption. Simple recurrent cells, e.g. ReLU or sigmoid neurons, tend to produce either vanishing or exploding gradients while training [268, 64]. The problem was severe enough that random guessing of weights and biases tended to produce better results than backpropagation [270]. This was circumvented by long short-term memory

(LSTM [269]) via moving to a slightly more complicated cell. LSTM tames the gradients by having two types of recurrent connections—short-term and, only weakly coupled to the rest of the network, long-term memory. Since the invention of LSTM, RNNs have exploded in popularity, and other cells that do not suffer from vanishing gradients were invented [238]. Some, such as gated recurrent unit [128], have only one type of memory. Note that search for good NN architectures can be automated [188]. As sequential processes are omnipresent in human life, today RNNs enjoy diverse technological applications and can improve both our social (e.g. [342, 21, 236, 254]) and physical (e.g. [653, 429, 360, 340]) well-being.

Any digital computation can be represented by a specific RNN of finite size (RNNs are *universal*—in a different sense than the FFNN). The output can be read from the network after a number of time steps that is asymptotically linear in both the length of the input and the number of time steps used by the corresponding Turing machine [548, 549, 547, 281].

RNNs are close relatives of matrix product operators, see Subsections 2.4.2 and 2.4.4. Quantum recurrent neural networks discussed in Chapter 5 can be thought of as quantum training algorithms for matrix product channels.

For more information about classical RNNs, see e.g. [227, Chapter 10].

2.7. Quantum data

ML demands access to interesting data sets. As in this thesis we assume that such data is available, in this section we review sources of quantum data that would benefit from some post-processing.

Quantum many-body systems are a natural source of entangled states; a deeper understanding of them would greatly benefit material science and chemistry. Quantum simulators are a vast source of quantum data. Their outputs may need extra processing to extract or predict interesting features. This may require the transmission of an output of an analog quantum simulator to a digital quantum computer. Measurement devices and cryptography can benefit from sources of entangled states that are robust to noise. Information may be converted from one physical platform to another or exchanged via a *quantum internet* [175, 612] for e.g. distributed or cloud quantum computing. However, interfaces and transmission channels may lead to distortion, becoming sources in need of denoising themselves.

2.7.1. Data from many-body systems

Quantum many-body systems, such as molecules, nuclei and certain materials, are notoriously hard to understand. Even though first-principle methods allow us to predict the behaviour of and control small systems (see e.g. [623, 171, 272] for a review and e.g. [557, 546, 480, 272, 271] for applications), their capabilities are limited. Despite stellar progress in incorporating physically sound approximations into numerical methods such as tensor networks (see Section 2.4), density functional [543, 115] and dynamical mean-field [31, 329] theory, determining properties of real-life systems is often outside of their reach. The accuracy necessary to identify the path of relatively

simple chemical reactions that we do know to occur, such as nitrogen capture in plants [488], is beyond current technological capabilities; predicting unknown properties of complex compounds for now remains a distant dream.

It should not come as a surprise that understanding quantum many-body effects was the initial motivation behind the idea of quantum simulators [382, 205]. As of today, there are a variety of platforms and algorithms tailored to the exploration of many-body systems, see e.g. [386]. However, further progress in hard- and software is necessary to address numerous exciting applications of these methods.

Surprisingly, if there are experimental means to do enough observations and comprise a classical data set, ML can vastly improve the current state of art in predicting important properties [114, 336, 291] (sometimes employing quantum computers along the way [359, 403]), even for such huge systems as proteins [534, 535].

The success of classical ML and promising algorithms for quantum computers bring high hopes for quantum ML as a means to explore quantum many-body systems. Unfortunately, collecting quantum data from a system of choice is a non-trivial task. Transferring the entire quantum state of a real-life system to a quantum computer is usually unfeasible. However, quantum data sets can be harvested either from quantum simulations of a system of interest, or from quantum states used to probe the system. We will discuss those scenarios in Subsections 2.7.2 and 2.7.3, respectively.

2.7.2. Quantum simulators as data sources

Outcomes of a simulation can be used to understand, mimic, simplify and improve the processes and systems under simulation. For example, a digital device can learn to reproduce an experiment (see Example 2.8.1). Such reproduction may be easier to run and analyze than the original. Various unwanted features of the underlying analog implementation, such as noise (see Section 4.3), memory (see Chapter 5) or hard to achieve parameters (see Section 4.2), can be canceled or widely adjusted by quantum ML on a digital reproduction.

Furthermore, quantum computers can be used for other tasks such as the analysis and classification of experimental outcomes, see e.g. [531, 527, 536], or the processing of classical data [366]. Long computations on NISQ devices can benefit from shallow denoisers (see Section 4.3). The obtained data can be shared with a physically separated interested party or uploaded to a more powerful quantum cloud computer to perform resource-demanding tasks [245].

2.7.3. Quantum metrology

Precise and fast measurements were instrumental for technological development in every epoch of our history. Notable examples include the development of calendars, clocks, interchangeable parts—a keystone of industrial revolution [501], magnetic resonance tomography and satellite navigation. Entangled states can vastly increase the sensitivity of measurements [223]. This statement can be formalized using the Fisher information (FI). This section follows to a large extent [203]. For a review of quantum metrology, see e.g. [452, 224].

Definition 2.7.1 (POVM). *A positive operator-valued measure (POVM) is a set of positive semi-definite matrices $\{M_i\}$ on a Hilbert space $\mathcal{H}$ that sum to the identity matrix $\sum_i M_i = \mathbb{1}$.*

The POVM element M_i can be associated with a measurement outcome μ_i, such that the probability to find μ_i when making a measurement on the quantum state ρ is given by

$$P(\mu_i) = \text{Tr}(\rho M_i). \tag{2.73}$$

We consider a density operator $\rho(\theta)$ that is a differentiable function of a parameter $\theta \in \mathbb{R}$. We are interested in the precision with which θ can be determined from measurements on $\rho(\theta)$.

Suppose there is a measurement with the possible discrete outcomes μ_i corresponding to a POVM $\{M_i\}$. The probability to measure μ_i given $\rho(\theta)$ is

$$P_\theta(\mu_i) = \text{Tr}[\rho(\theta)M_i]. \tag{2.74}$$

The difference between the probability distributions P_θ and $P_{\theta+\epsilon}$ can be quantified by the Hellinger distance

$$[d(P_\theta, P_{\theta+\epsilon})]^2 = \frac{1}{2}\sum_j \left(\sqrt{P_\theta(\mu_j)} - \sqrt{P_{\theta+\epsilon}(\mu_j)}\right)^2. \tag{2.75}$$

The corresponding statistical speed [221], or **sensitivity**,

$$\lim_{\epsilon \to 0} \partial_\epsilon d(P_\theta, P_{\theta+\epsilon}) = \sqrt{\frac{F(\theta)}{8}} \tag{2.76}$$

can be expressed in terms of the classical Fisher information (CFI)

$$F(\theta) = \sum_j \frac{1}{P_\theta(\mu_j)} [\partial_\theta P_\theta(\mu_j)]^2 = \sum_j P_\theta(\mu_j) (\partial_\theta \ln[P_\theta(\mu_j)])^2. \tag{2.77}$$

Thus, the CFI quantifies how fast P_θ changes with θ and depends on the POVM. Maximizing the CFI over all POVMs defines the quantum Fisher information (QFI) [95].

When accessing the precision of measurements, the statistical speed is not the only possible choice for a figure of merit. A popular alternative is the variance of the observable. Conveniently, the FI also provides a bound for the achievable variance.

Theorem 2.7.2 (Cramér-Rao bound, [214, 140, 139, 479]). *Suppose θ is an unknown deterministic parameter to be estimated from N independent measurements. The variance of any unbiased estimator Θ of θ is then bounded by the reciprocal of the CFI $F^{[N]}(\theta)$:*

$$\Delta_\theta^2 \Theta \geq \frac{1}{F^{[N]}(\theta)}. \tag{2.78}$$

For large N the bound is asymptotically saturated by the maximum likelihood estimator [206, 451, 356].

To understand how sensitive and precise a measurement may be, we need to evaluate the maximal value that the FI can attain.

Theorem 2.7.3 (Convexity of FI)**.** *Consider a convex combination of density operators* $\{\rho^{(k)}(\theta)\}$,

$$\rho(\theta) = \sum_k p_k \rho^{(k)}(\theta), \quad with \quad \sum_k p_k = 1, \ 0 \le p_i. \tag{2.79}$$

Both the CFI F *and the QFI* F_Q *are convex,*

$$F(\theta) \le \sum_k p_k F^{(k)}(\theta) \quad and \quad F_Q(\theta) \le \sum_k p_k F_Q^{(k)}(\theta). \tag{2.80}$$

Theorem 2.7.4 (Additivity of FI)**.** *Consider a tensor product of density operators* $\{\rho^{(k)}(\theta)\}$,

$$\rho(\theta) = \bigotimes_k \rho^{(k)}(\theta). \tag{2.81}$$

Both the CFI F *and the QFI* F_Q *are additive,*

$$F(\theta) = \sum_k F^{(k)}(\theta) \quad and \quad F_Q(\theta) = \sum_k F_Q^{(k)}(\theta). \tag{2.82}$$

Additivity and convexity limit the FI of classically correlated states (see Definition 2.3.1).

Corollary 2.7.5 (Standard quantum limit (SQL))**.** *Suppose that the one-partite CFI* $F^{[1]}$ *is bounded from above,* $F^{[1]}(\theta) \le c$. *Then the CFI of a classically correlated state can grow at most linearly with the system size* N:

$$F^{[N]}(\theta) \le cN. \tag{2.83}$$

The same holds for the QFI.

   In particular, given the CFI F *and QFI* F_Q *of a single experiment, the corresponding CFI* $F^{[N]}$ *and QFI* $F_Q^{[N]}$ *of an experiment repeated* N *times are*

$$F^{[N]}(\theta) = NF(\theta) \quad and \quad F_Q^{[N]}(\theta) = NF_Q(\theta). \tag{2.84}$$

However, it turns out that the FI can exceed the SQL and have a more favourable scaling for entangled states.

Theorem 2.7.6 (FI for unitary evolution)**.** *Let* R *be a Hermitian operator and* $|\psi(\theta)\rangle = e^{-i\theta R}|\psi(0)\rangle$. *Then the QFI of* $\rho(\theta) = |\psi(\theta)\rangle\langle\psi(\theta)|$ *is*

$$F_Q = 4\left(\langle\psi(0)|R^2|\psi(0)\rangle - \langle\psi(0)|R|\psi(0)\rangle^2\right). \tag{2.85}$$

Theorem 2.7.7 (QFI is entanglement depth witness)**.** *Let* $R = \sum_{l=1}^N r^l \in \mathcal{B}(\mathcal{H}^{\otimes N})$, *where* $r^{(l)} = \mathbb{1}^{\otimes(l-1)} \otimes r \otimes \mathbb{1}^{\otimes(N-l)}$ *and* $|\psi(\theta)\rangle = e^{-i\theta R}|\psi(0)\rangle$. *Let the largest eigenvalue of* r *be* r_+ *and the lowest* r_-. *For a state with entanglement depth (see Definition 2.3.2)* k *the following bound for the QFI* F_Q *holds:*

$$F_Q \le (r_+ - r_-)^2(ak^2 + b), \quad where \quad a = \left\lfloor \frac{N}{k} \right\rfloor, \ b = N - ak. \tag{2.86}$$

The bound is tight: for an entanglement depth of k, *it is saturated by*

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}^{a+1}} \left(|r_+\rangle^{\otimes k} + |r_-\rangle^{\otimes k}\right)^{\otimes a} \otimes \left(|r_+\rangle^{\otimes b} + |r_-\rangle^{\otimes b}\right) \tag{2.87}$$

According to the expressions for the statistical speed (see Equation 2.76) and the Cramér-Rao bound (Theorem 2.7.2), states with a large entanglement depth allow to achieve the same measurement sensitivity with fewer repetitions. Quantum-enhanced metrology is an experimentally viable technique with increasing appeal.

Contemporary gravitational-wave detectors use squeezed light to exceed the SQL [583, 242, 6] (technique proposed in [116]). Proof-of-principle quantum-enhanced atomic clocks already exist [339, 445, 357, 278], and entanglement may soon improve cutting-edge clocks [260]. Atomic inertial sensors can benefit from entanglement in momentum space [510, 338, 567, 540]; central building blocks have recently been demonstrated [26, 642]. The read-outs of superconducting [609] and ion [388] systems used in quantum information processing have also been quantum-enhanced. Note that beyond-SQL sensitivity is not useful per se; balancing out constrains and goals of a particular device demands a detailed analysis [566, 532].

While quantum metrology is a very promising and already a practically useful tool, noise is a great challenge for its wider success. Macroscopically entangled state tend to be extremely fragile [217, 370, 174]. States that are especially sensitive to some signal are often also very sensitive to noise. A measurement apparatus needed to harness all the metrological advantage may be forbiddingly complicated [553]. Discovering states robust to decoherence [216, 173, 554], understanding why these states are stable [203] and protecting them from detrimental effects of the environment [176] is an important area of research. ML can greatly aid in this job [322, 421, 391].

A more in-detail introduction to quantum-enhanced metrology, as well as a discussion of the robust preparation of useful macroscopically entangled states using spin-1 BECs, can be found in [203].

2.7.4. Interfaces

Different physical systems are suitable for different tasks. For example, superconducting qubits [316] and trapped ions [104] are leading platforms for digital quantum computation (see Section 2.5), while cold atoms can be used for quantum-enhanced measurement of time and gravity (see Subsection 2.7.3). NV centers are suitable for nanoscale sensing [519] and memory [552]. Photons are the default medium for quantum communication (see Subsection 2.7.5), and squeezed light is used for gravitational wave detection.

There is a great interest in hybrid systems, which require interfaces between different quantum devices. Information can be converted from one device, say an analog simulator or a metrological apparatus, to another, such as a digital post-processor. Conversely, a quantum computer can be used as a state synthesizer for an analog piece of equipment. Quantum processor and memory can be realized on different physical platforms [509]. Multiple devices can be connected via photon links to form a quantum distributed system. Two variants of the same system can be encompassed in one hybrid machine. For example, in quantum logic spectroscopy [522], one type of ions is used for the state initialisation, control and sympathetic cooling of another species of ions used to observe physical quantities [100, 253].

Fortunately, quantum hybrid systems have been demonstrated, e. g., for superconducting qubits coupled to atomic and spin ensembles [346, 643], for trapped ions with cold atoms [575] and atomic systems coupled to light [250].

Unfortunately, every interface brings an extra possibility for errors. Quantum autoencoders discussed in Section 4.3 can remove deteriorating effects introduced at the interface between the coupled platforms.

2.7.5. Quantum communication

Quantum communication is the art of transferring a quantum state from one place to another. Not only that the state has to be transmitted faithfully—often it also has to be securely encrypted. While technically challenging, quantum communication is a vital part of distributed quantum systems and, due to increased security, a lucrative technology even for the transmission of classical data. Moreover, it vastly expands the applicability of quantum ML by providing access to remotely generated quantum data.

In the following, we sketch some of the basic building blocks of quantum communication between two parties—the sender A and the receiver B. Along the way, we point out how these building blocks can benefit from post-processing by quantum ML, including the methods described in Chapters 4 and 5.

Quantum teleportation and entanglement swapping

Channels that can transmit arbitrary entangled states may be hard to build. Fortunately, the information can be sent via a *quantum teleportation* protocol using a shared source of Einstein–Podolsky–Rosen (EPR [181, 83]) pairs $|\phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}(|\downarrow\downarrow\rangle + |\uparrow\uparrow\rangle)$ and the communication of two classical bits.

Let us demonstrate how to teleport the state $|s\rangle_A = \alpha|\downarrow\rangle + \beta|\uparrow\rangle$.

Definition 2.7.8 (Bell basis). *The following four states form an orthonormal 2-qubit basis*

$$|\phi^\pm\rangle = \frac{1}{\sqrt{2}}(|\downarrow\downarrow\rangle \pm |\uparrow\uparrow\rangle), \quad |\psi^\pm\rangle = \frac{1}{\sqrt{2}}(|\downarrow\uparrow\rangle \pm |\uparrow\downarrow\rangle). \tag{2.88}$$

Suppose that the parties A and B share a state

$$
\begin{aligned}
|s\rangle_A \otimes |\phi^+\rangle_{AB} = \frac{1}{2} \big[&|\phi^+\rangle_A \otimes (\alpha|\downarrow\rangle + \beta|\uparrow\rangle)_B + |\phi^-\rangle_A \otimes (\alpha|\downarrow\rangle - \beta|\uparrow\rangle)_B + \\
&|\psi^+\rangle_A \otimes (\alpha|\uparrow\rangle + \beta|\downarrow\rangle)_B + |\psi^-\rangle_A \otimes (\alpha|\uparrow\rangle - \beta|\downarrow\rangle)_B \big].
\end{aligned}
\tag{2.89}
$$

The sender A can perform a measurement in the Bell basis and send the outcome, encoded in two bits, to B. Depending on the outcome, the receiver B performs one of $\{\mathbb{1}|_{\phi^+}, \sigma^z|_{\phi^-}, \sigma^x|_{\psi^+}, \sigma^z\sigma^x|_{\psi^-}\}$ to obtain $|s\rangle_B$.

Quantum teleportation was originally proposed in [70] and experimentally demonstrated in [90, 93]. Today teleportation has been demonstrated up to distances of 1400 km using satellite communication [492].

A related protocol—*entanglement swapping* [282]—allows to entangle two parties A and B, which do not need to interact, via a third party C that shares an entangled bipartite state with A and with B. Let C perform a Bell measurement on $|\phi^+\rangle_{AC} \otimes |\phi^+\rangle_{BC}$ and communicate the outcome to B. Just like in the teleportation protocol, based on the outcome B performs one of $\{\mathbb{1}|_{\phi^+}, \sigma^z|_{\phi^-}, \sigma^x|_{\psi^+}, \sigma^z\sigma^x|_{\psi^-}\}$ to obtain $|\phi^+\rangle_{AB}$. The pioneering experimental demonstration is described in [438].

Teleportation and its relatives can be generalized to higher dimensions and sharing any predetermined maximally entangled state, see [620].

Quantum repeaters

Signals attenuate exponentially with the depth of the communication medium they are sent through. This fall is typically negligible at the scale of meters and can be mitigated for up to a thousand kilometers in satellite communication, albeit at the price of a reduced transmission rate and a larger equipment size. However, the attenuation is a serious problem for Earth-wide communication or low-power sender and receiver devices. The solution is to use a network of repeaters that amplify the signal and are positioned at an appropriate distance from one another (see Fig. 2.34).
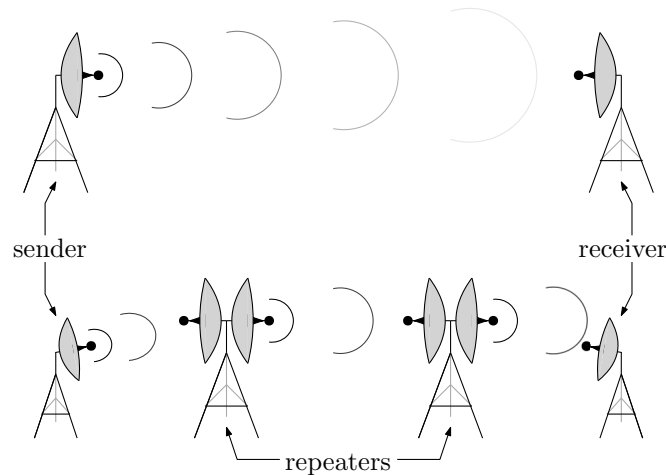
Figure 2.34.: Use of repeaters between sender and receiver to get a better signal while using smaller communication equipment.

No-broadcasting (Theorem 2.5.9) requires quantum amplifiers to differ significantly from conventional classical technology. The key to enhancing quantum signals is the quantum teleportation protocol. While we cannot produce multiple copies of an unknown state for amplification (without signal deterioration, see Theorem 2.5.11), we can produce multiple EPR pairs. Once distributed, the pairs will have a decreased fidelity F with the ideal EPR state. Fortunately, the entanglement can be *distilled* [71]—there is an LOCC (see Section 2.3) operation that produces one pair with high fidelity to a given target state from several source pairs with lower fidelity to this state.

Unfortunately, distillation works only if $F \geq F_{\min}$ for some cut-off fidelity $F_{\min}$ [305]. This makes the distribution of EPR pairs over large distances technologically unfeasible. Quantum repeaters can address this problem by combining distillation with entanglement swapping [102].

Let the distance L between repeater nodes $\{C_1, \cdots, C_{m+1}\}$ be such that $F > F_{\min}$ can be achieved. First, $n \times m$ EPR pairs are distributed and distilled to obtain m pairs shared by the neighbouring nodes. Next, the entanglement is swapped to obtain an EPR pair with fidelity of at least F between the nodes C_1 and C_{m+1} separated by the distance mL. See Fig. 2.35 for an example. These steps can be repeated in a nested fashion to obtain maximally entangled pairs at an arbitrary distance using technologically feasible communication devices.
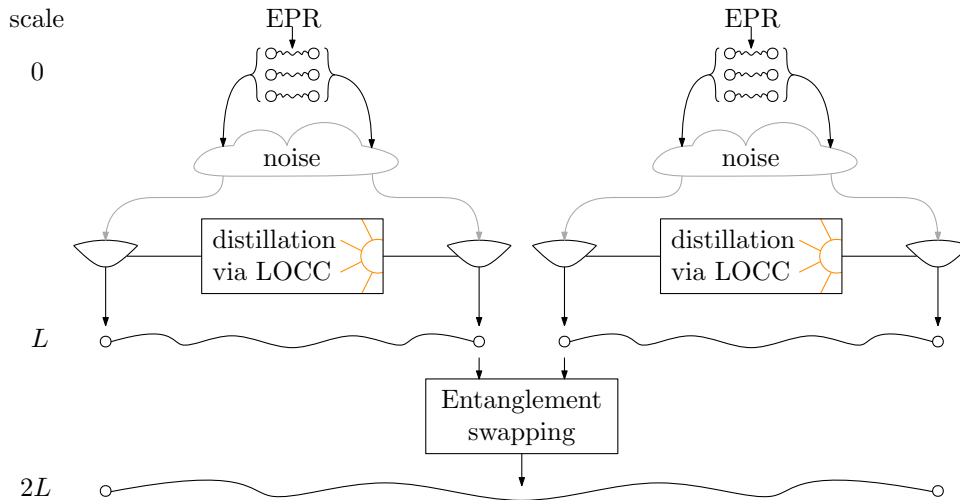
Figure 2.35.: A quantum repeater using $n = 3$ EPR pairs for distillation and swapping entanglement between $m = 2$ links.

The repeater nodes in the protocol discussed above should be equipped with memory (see Subsection 2.7.6), as they need to store the shared EPR pairs until all the links in the network are ready [542]. This requirement is a bottleneck in practical implementations.

Quantum repeaters have been experimentally demonstrated [647, 511]. Unfortunately, the technology is not yet developed enough to build a commercially viable quantum repeater. More complicated protocols were invented [411, 39] and demonstrated [361] in order to relax the memory constraint. The need for storage can be eliminated either completely or to an extent [89] taking into account the trade-offs [439, 107] imposed by other components in the set-up. These protocols are often based on the resilience of entanglement in graph (use in repeaters suggested in in [656, 655]) or code [411, 213, 191, 412] states and require the direct transition of quantum information. This makes the quantum autoencoders discussed in Section 4.3 a compelling tool for realizing quantum repeaters without memory. Moreover, quantum autoencoders can be trained to denoise code states (see Section 4.4) and, if the employed hardware platform offers some small amount of storage, endowed with recurrent connections (see Chapter 5) for a robust and economical quantum repeater.

Quantum key distribution

Quantum information cannot be broadcasted (see Theorem 2.5.9), and this can be advantageous for the security of communication. Quantum key distribution (QKD) protocols can detect whether an eavesdropper has accessed a significant portion of a generated key.

The first proposed [68] and implemented [69] QKD protocol is BB84. Inspired by [628] and building on [66], it encodes information into non-orthogonal sets of states.

Let us consider two bases in $\mathbb{C}^2$: $+ = \{|0\rangle, |1\rangle\}$ and $\times = \{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\}$. The BB84 protocol starts with the sender randomly choosing a basis from $\{+, \times\}$ and sending a random basis state to the receiver. The received state is measured in a basis chosen at random from $\{+, \times\}$. The transmission and measurement are repeated for L rounds. Afterwards, the parties publicly exchange via an authenticated classical channel their choice of measurement bases; if their choices differ, the round is ignored. A randomly chosen half is publicly discarded from the remaining $\sim L/2$ measurement outcomes to obtain a length $\sim L/4$ key. The parties proceed by running a check on a test message to verify the successful key generation, otherwise the process is started from scratch. An eavesdropper trying to peep the state of a transmitted qubit has to guess the measurement basis. An incorrect guess projects the qubit into another state, which leads, with 50% probability, to a one-bit error in the receiver's key. Such an error gets detected in decoding the test message. This protocol is provably secure if the devices that the parties use locally to execute the steps of the protocol, e.g., for preparation and measurement, do exactly what they are instructed to do. There is a non-zero threshold for how many bits can differ between the parties (due to noise or eavesdropping) in a usable key [67, 120].

QKD using BB84 is a mature (see e.g. [472, 362], for a review see [369] and Table 1 in [327]) and commercially available technology [513, 635, 632, 634]. It is, e.g., used in a communication network spanning over 4600 km [125]. This may seem to contradict the statements about the unsatisfactory performance of current quantum repeaters. However, this network is using trusted repeater nodes. That is, the network cannot distribute qubits over long distances. Instead, quantum key distribution is performed between neighboring nodes and the key is stored (classically) at each repeater station, which has to be trusted.

Entanglement can be used for *device independent (DI) QKD*, where the need for trust in devices used locally can be eliminated.

Definition 2.7.9 (Bell inequality [60])**.** *Consider p parties sharing a state ρ and a finite set of one-partite observables $\{O_{ij} \in \mathcal{B}(\mathcal{H}_i)\}$ such that $-\mathbb{1} \leq O_{ij} \leq \mathbb{1}$. We say that a function $f : \times_{ij} \mathcal{B}(\mathcal{H}_i) \to \mathcal{B}(\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_p)$ yields a Bell inequality if $\exists \beta_c \in \mathbb{R}$ such that*

$$\rho \text{ is classically correlated} \quad \Rightarrow \quad \mathrm{Tr}\left[\rho f(\{O_{ij}\})\right] \leq \beta_c \quad \text{and}$$

$$\exists \, \rho_q \in \mathcal{T}\left(\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_p\right), \, \mathrm{Tr}[\rho_q] = 1 : \quad \mathrm{Tr}\left[\rho_q f(\{O_{ij}\})\right] > \beta_c. \tag{2.90}$$

The state ρ_q is said to violate the Bell inequality.

States that violate a Bell inequality form a **strict** subset of entangled states, e.g. [276]. The simplest example of a Bell inequality is the Clauser-Horne-Shimony-Richard Holt (CHSH) inequality.

Theorem 2.7.10 (CHSH [133]). *Given two parties, $\exists \{O_{ij} \in \mathcal{B}(\mathcal{H}_i)\}_{i,j=1}^2$ with eigenvalues ± 1 such that the function*

$$f_{\mathrm{CHSH}}\left(\{O_{ij}\}_{i,j=1}^2\right) = O_{11} \otimes O_{21} + O_{11} \otimes O_{22} + O_{12} \otimes O_{21} - O_{12} \otimes O_{22} \qquad (2.91)$$

yields a Bell inequality. It satisfies the Tsirelson's (or Cirel'son's) bound [132]

$$\rho \text{ is classically correlated} \quad \Rightarrow \quad \mathrm{Tr}\left[\rho f_{\mathrm{CHSH}}(\{O_{ij}\})\right] \leq 2 \qquad and$$

$$\exists\, \rho_q \in \mathcal{T}\left(\mathcal{H}_1 \otimes \mathcal{H}_2\right),\ \mathrm{Tr}[\rho_q] = 1: \quad \mathrm{Tr}\left[\rho_q f_{\mathrm{CHSH}}(\{O_{ij}\})\right] = 2\sqrt{2}. \qquad (2.92)$$

Moreover,

$$\mathrm{Tr}\left[\rho f_{\mathrm{CHSH}}(\{O_{ij}\})\right] = 2\sqrt{2} \quad \Rightarrow \quad \rho = \rho_r \otimes \rho_{2\,\mathrm{max}}, \qquad (2.93)$$

where $\rho_{2\,\mathrm{max}}$ is a maximally entangled two-qubit state and ρ_r does not matter [618].

Bell inequality violations are observed with ever increasing efficiency since 1970s, e.g. [215, 37, 36, 614, 437, 500]. Such experiments have many applications, from answering fundamental questions about quantum mechanics, e.g. [493, 123], and quantum field theories [563, 562] to characterising correlations in BEC experiments, e.g. [523].

The earliest QKD protocol using the violation of a Bell inequality is E91 [184] (see e.g. [585, 365] for early experiments): Suppose that the sender and receiver share L copies of a maximally entangled state, e.g. $|\phi^+\rangle_{AB}$. In extension to BB84, for each copy of $|\phi^+\rangle_{AB}$ the parties randomly choose a basis from $\{+, \times, \pi/4\}$, where the $\pi/4$ basis is $\{\cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle, \sin(\pi/8)|0\rangle - \cos(\pi/8)|1\rangle\}$. Both parties measure in the chosen basis and, after observing the outcomes, publicly share their choice via an authenticated classical public channel. If both parties have chosen the $\pi/4$ basis, the measured outcome constitutes a key bit; otherwise, the outcomes are used to measure the CHSH inequality. If the Bell violation is sufficiently close to $2\sqrt{2}$, the sender and receiver can be sure that the states they share are approximately uncorrelated with a potential eavesdropper.

The E91 protocol can be somewhat simplified and modified to achieve DI. One of the parties can perform measurements in two bases instead of three [11, 9]. DI is ensured by abandoning the protocol if bases choices significantly deviate from the expected uniform distribution or if the CHSH measurements fail to yield the expected correlation [50, 9].

A large Bell violation is essential for the security of DI QKD (see e.g. [464] for a review of security in QKD). Curiously, while the GHZ state maximally violates a complete set of inequalities for n-partite systems with two dichotomic observables each [622], a high entanglement entropy is not necessary for a strong Bell violation [10, 12, 297]. This observation motivates the search for Bell-violating states, as well as inequalities and corresponding experimental set-ups, that are much more robust to noise; ease in incorporating various practical constraints makes ML a compelling tool for this task [392, 160, 112, 79].

Unlike for BB84, only proof-of-principle demonstrations of DI QKD are available to date. The rate of key generations is way too slow to even approach practicality [414]. Fortunately, recent theoretical proposals suggest that this technology may soon mature enough for applications [533].

For more information, see e.g. [457].

2.7.6. Quantum memory

Quantum memories are hard to realize experimentally. A significant fraction of the stored information can be lost or corrupted. Nevertheless, it is an enabling technology for a variety of applications.

Unless specifically stated, the implementations and technical details in this subsection can be found in [267, 552].

Platforms used for proof-of-principle demonstrations include such diverse sets of physical systems as rare-earth-ion doped solids, nitrogen-vacancy centers in diamond, alkali vapours (ranging from cryogenic to above room-temperature), semiconductor quantum dots, molecules and superconducting quantum circuits [485]. All of these systems are coupled to light, as is desirable for quantum communication (see Subsection 2.7.5).

There are several requirements that quantum memory should ideally fulfill. The fidelity between the original and recalled state should approach 1. Fidelities of the order of 0.9 can be achieved in several platforms, with few outliers able to reach the order of 0.95. The recall efficiency—the probability to re-emit photons that have been stored, should be high. The current state-of-art is well below the believed fundamental limitations of underlying physical platforms. Typical demonstrated efficiencies range from several to several tens of percent, yet recently a few exceptional implementations were able to reach efficiencies on the order of 80% [126, 126, 277]. Good memory should have a long coherence time. While there are experiments that probe the storage time scale of a minute [406, 263, 264], typical coherence times are on the order of $10^2 - 10^3$ μs (see [316] for the progress in superconducting devices). Finally, it is nice to have as much memory as possible, being preferably also capable of storing states with a significant entanglement depth (see Section 2.3). Although, for many quantum repeater protocols it is sufficient to store individual photons. Current experiments realize quantum memories of more than a hundred, yet not very entangled, qubits [293, 471]. Unfortunately, rarely does a single platform excel at different requirements.

There are large gaps between theoretically feasible and currently realized capacities to store quantum information. ML is a promising and often low-effort method to shrink these gaps. For example, it has been used to design a driving that significantly prolongs the coherence time [264]. We can also understand quantum memory as a source of data in need of denoising. In particular, quantum autoencoders (see Section 4.3) could be used to improve the read-out of the stored data.

2.8. Quantum neural networks

Machine learning (ML) could benefit from the rapid progress of quantum computing hard- and software (see e. g. [424, 621]). Moreover, there are important ML tasks where the data comes as a set of quantum, and possibly—highly entangled, states; see Section 2.7. We call ML quantum if it uses quantum algorithms and quantum data (see e. g. [63, 629]).

A big benefit of using quantum algorithms for ML tasks is that quantum speed-ups can be

obtained with general-purpose algorithms even for classical data sets [366]. Of course, the speed-up is not guaranteed, yet ML can help to vastly expand a rather small set of algorithms that show quantum advantage [424].

Neural networks (NNs) are an extremely capable and flexible class of models in classical ML (see Subsection 2.6.2). In this section we motivate and describe their quantum analog. We will use these quantum NNs (QNNs) in Chapters 4 and 5 to denoise quantum sources.

2.8.1. Criteria for a good quantum neuron

Various quantum neurons have been proposed in [56, 530, 358, 606, 145, 24, 226, 330, 577, 198, 529, 113, 309, 34, 559], see e.g. [381] for a review. It is not obvious which proposals can have higher merit for this thesis, thus it is worthwhile to make a list of criteria a suitable QNN must fulfill.

Criteria for a good QNN

1. A QNN should consist of simple building blocks: neurons and connections between them.

2. A QNN should be capable of universal quantum computation.

3. Deep QNNs could be implemented and trained efficiently, e.g. using few qubits.

4. There is a possibility of probabilistic output due to e.g. contradicting labels.

There are also extra conditions that, while may be not strictly necessary, make any proposal much more appealing.

1. The cost function should have a clear operational meaning.

2. There should be a convincing set of examples where QNNs work in practice.

3. Neurons should be capable of building networks of various topologies without additional elements.

4. An open-source implementation should be available.

2.8.2. Networks of quantum channels

We follow [56], since these QNNs satisfy all of the desired criteria (yet there are also other worthwhile QNNs, e.g. [606]). These networks are essentially networks of quantum channels (see Fig.2.36) and can perform general operations with mixed states—rather convenient for denoising applications studied in this thesis. The parameters of such a QNN are classical variables. In general, quantum parameters may be useful [594], but for ML tasks without memory they can give only a marginal improvement [399]. When the memory is needed, we use recurrent architectures, see Chapter 5.

We specify the quantum neuron from [56] by attributing a single qubit to every neuron. Let $\{|\uparrow\rangle, |\downarrow\rangle\}$ denote an orthonormal basis of a qubit. In each layer following the input, the jth neuron acts by a unitary U_j on its own qubit and the preceding layer. The non-input qubits are initialized in $|\downarrow\rangle$. The kth layer, $k > 1$, of m neurons maps the state ρ_{k-1} of layer $k - 1$ onto

$$\mathcal{N}^k(\rho_{k-1}) \equiv \mathrm{Tr}_{k-1}\left(U \left(\rho_{k-1} \otimes (|\downarrow\rangle_{\mathrm{out}}\langle\downarrow|)^{\otimes m} \right) U^\dagger \right), \tag{2.94}$$

where the unitary $U \equiv U_m \dots U_1$ is subject to optimization (see Fig. 4.14). Note that this definition is related to the general form of a quantum channel (see Theorem 2.1.10). The quantum channel describing the full network with K layers is $\mathcal{N}(\rho^{\mathrm{in}}) = \mathcal{N}^K(\cdots \mathcal{N}^2(\rho^{\mathrm{in}}) \cdots)$.

$$\rho^{\mathrm{out}} = \mathrm{tr}_{\mathrm{in,hid}}\left(U_{\mathrm{in,out,hid}} \left(\rho^{\mathrm{in}} \otimes | \downarrow \cdots \downarrow\rangle_{\mathrm{out,hid}}\langle\downarrow \cdots \downarrow | \right) U^\dagger_{\mathrm{in,out,hid}} \right)$$

$$U_{\mathrm{in,hid,out}} = U^K_{m_K} U^K_{m_K\ 1} \dots U^2_2 U^2_1$$

Figure 2.36.: A quantum neural network defined in [56] is a network of quantum channels (see Section 2.1). Each qubit in the open-system representation 2.1.10 of the full channel is associated to a neuron. Learning parameters are contained in a unitary $U_{\mathrm{in,out,hid}}$ that is a concatenation of many unitaries U_i^J. Each U_i^J is associated to the connections in the network, that is U_i^J acts non-trivially only on the ith qubit in the Jth layer and all the directly connected qubits in the previous layer.

These QNNs are capable of universal quantum computation even when the connectivity of a network is limited. A network of 4 neurons: two input connected to each of two output, can learn any 2-qubit unitary V. Indeed, such networks can apply V to the input qubits followed by the SWAP of the state from input to output qubits, see Fig. 2.37. Two-qubit gates are universal (see subsection 2.5.3), so networks composed of neurons with two inputs and two outputs are also universal.
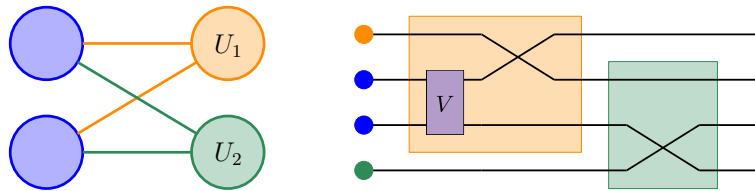
Figure 2.37.: Let us number input qubits as (in,1), (in,2) and output qubits as (out,1), (out,2). For every input two-qubit state ρ the depicted QNN outputs $V\rho V^\dagger = \mathrm{Tr}_{\mathrm{in}}\left[U_2 U_1 \left(\rho \otimes |\uparrow\uparrow\rangle\langle\downarrow\downarrow| \right) U_1^\dagger U_2^\dagger \right]$, where $U_1 = \left(\mathrm{id}_{(\mathrm{out},1)} \otimes V_{(\mathrm{in},1)(\mathrm{in},2)} \right) \cdot \left(\mathrm{SWAP}_{(\mathrm{out},1)(\mathrm{in},1)} \otimes \mathrm{id}_{(\mathrm{out},2)} \right)$ (orange) and $U_2 = \mathrm{id}_{(\mathrm{in},1)} \otimes \mathrm{SWAP}_{(\mathrm{in},2)(\mathrm{out},2)}$ (green).

The number of qubits needed to evaluate $\mathcal{N}^k(\rho_{k-1})$ is the sum of width of kth and $k - 1$st layers. As the whole network is just the concatenation of individual layers, at most $2 \times$ width qubits are needed to evaluate $\mathcal{N}(\rho^{\text{in}})$.

An advantage of variational classes that can simulate quantum channels, and are not restricted to unitary evolution, is the possibility of uncertain outputs. The QNNs we use are capable of learning and producing meaningful outputs even if there are contradicting labels in the data set, see Section 4.1 and Subsections 4.3.2 and 4.3.4.

Any distance measure for pure states is a function of fidelity F. For training data $\{\rho_i^{\text{in}}, |\psi_i^{\text{ref}}\rangle\}_{i=1}^L$ with pure desired outputs, $F(\rho, |\psi\rangle) = \langle\psi|\rho|\psi\rangle$. We choose the cost function as

$$C\left(\{\rho_i^{\text{in}}, |\psi_i^{\text{ref}}\rangle\}_{i=1}^L\right) = 1 - \bar{F}\left(\{\mathcal{N}(\rho_i^{\text{in}}), |\psi_i^{\text{ref}}\rangle\}_{i=1}^L\right), \tag{2.95}$$

where $\bar{F}(\{\rho_i, |\psi_i\rangle\}_{i=1}^L) = \frac{1}{L}\sum_{i=1}^L F(\rho_i, |\psi_i\rangle) \leq 1$. In the following, we abbreviate pure $\rho_i^{\text{in}} = |\psi_i^{\text{in}}\rangle\langle\psi_i^{\text{in}}|$ by $|\psi_i^{\text{in}}\rangle$.

For mixed states, however, there are many nonequivalent distance measures. The fidelity for mixed states ρ and σ is generalized as the largest fidelity between any two purifications of the given states [296]. By the theorem of Uhlmann [584]

$$F(\rho, \sigma) = \left(\text{Tr}\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2. \tag{2.96}$$

It is hard to design efficient quantum algorithms that estimate fidelity of two mixed states—the first algorithm with complexity scaling as a polynomial in the number of qubits appeared a couple of weeks before this thesis was submitted [608]. It is also hard to design efficient algorithms that obtain various other quantities with interesting operational interpretations [424], such as relative entropy [591] $S(\rho\|\sigma) = \text{Tr}\left[\rho(\log\rho - \log\sigma)\right]$ and a rather general recently introduced class of kringel divergences [74], or probability to distinguish two states via optimal measurement [273, 266, 434] $\frac{1}{2} + \frac{1}{4}\|\rho - \sigma\|_1$.

It is easy to calculate $\langle\psi|\rho|\psi\rangle$ and, more generally, the overlap $\text{Tr}[\rho\sigma]$, $\rho, \sigma \in \mathcal{T}(\mathcal{H})$ on a quantum computer via SWAP test [193], see Fig. 2.38. This leads us to use $\|\rho - \sigma\|_2^2 = \text{Tr}\left((\rho - \sigma)^\dagger(\rho - \sigma)\right)$ as a distance measure for mixed states. It can be calculated by evaluating $\text{Tr}(\rho^2)$, $\text{Tr}(\sigma^2)$ and $\text{Tr}(\rho\sigma)$. We choose the cost function for problems that involve mixed labels as

$$C_{\text{HS}}\left(\{\rho_i^{\text{in}}, \sigma_i^{\text{ref}}\}_{i=1}^L\right) = 1 - \frac{1}{L}\sum_{i=1}^L \|\mathcal{N}(\rho_i^{\text{in}}) - \sigma_i^{\text{ref}}\|_2^2. \tag{2.97}$$
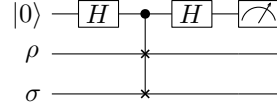
Figure 2.38.: A circuit for calculating the overlap via the SWAP test. The inputs are states ρ and σ, one extra qubit is initialized in the $|0\rangle$ state. The Hadamard gate H is applied to the first qubit, followed by control-SWAP gates controlled by ρ and σ. One more Hadamard gate is applied to the first qubit followed by a measurement in computational basis. The probability to measure 0 is $\frac{1}{2} + \frac{1}{2}\mathrm{Tr}(\rho\sigma)$.

The QNN can be optimised via the gradient descent (see Subsection 2.6.3). Let the current value of neuron unitaries be $U_I^J \in \mathcal{B}\left(\mathbb{C}^{2^{m_{J-1}+1}}\right)$. For a given training data, the cost to be minimized depends on the neuron unitaries

$$C\left(\{\rho_i^{\mathrm{in}}, \sigma_i^{\mathrm{ref}}\}_{i=1}^L\right) = \tilde{C}\left(\left\{\mathcal{N}|_{\{U_I^J\}}(\rho_i^{\mathrm{in}}), \sigma_i^{\mathrm{ref}}\right\}_{i=1}^L\right) \equiv \tilde{C}(\{U_I^J\}). \tag{2.98}$$

There exists a set of Hermitian basis $\{s_{IJ}^\alpha\}$ such that the exact gradient of the cost with respect to a variational parameterization of the neuron unitaries can be deduced from [528]

$$\left\{\tilde{C}(\{e^{is_{IJ}^\alpha}U_I^J e^{-is_{IJ}^\alpha}\})\right\}. \tag{2.99}$$

Thus, a round of the gradient descent can be performed by evaluating each of

$$\theta_{IJ}^\alpha = \tilde{C}(\{e^{is_{IJ}^\alpha}U_I^J e^{-is_{IJ}^\alpha}\}) - \tilde{C}(\{U_I^J\}) \tag{2.100}$$

and updating the neuron unitaries as

$$U_I^J \to e^{i\frac{2^{m_{J-1}}\lambda}{L}\sum_{I,J,\alpha}\theta_{IJ}^\alpha s_{IJ}^\alpha}U_I^J e^{-i\frac{2^{m_{J-1}}\lambda}{L}\sum_{I,J,\alpha}\theta_{IJ}^\alpha s_{IJ}^\alpha}, \tag{2.101}$$

where λ is the learning rate. While this algorithm is easy to implement on a quantum computer, its complexity scales exponentially with the amount of connections between the consecutive layers. While it might not be the problem for relatively small or sparse networks, yet an approach with better scaling is highly desirable. This issue is **not** architecture-specific.

A gradient of the network with respect to neuron unitary parameterization contains terms of the form

$$\nabla_{\theta^\beta}\mathrm{Tr}_{k-1}\left(e^{i\sum_\beta \theta^\beta s^\beta}U e^{-i\sum_\beta \theta^\beta s^\beta}\left(\rho_{k-1}\otimes(|\downarrow\rangle_{\mathrm{out}}\langle\downarrow|)^{\otimes m}\right)e^{-i\sum_\beta \theta^\beta s^\beta}U^\dagger e^{i\sum_\beta \theta^\beta s^\beta}\right) \tag{2.102}$$

that contain commutators. An algorithm that can compute commutators efficiently can be leveraged for QNN optimization with favourable scaling with width, see Supplementary note 5 in [56] for details.

The computational complexity per training round scales at most quadratically with the depth of the NN.

Example 2.8.1 (Unitary evolution)**.** A simple, yet an immensely useful task is learning to reproduce evolution. In the simplest case, the task reads: given the training data $\{|\psi_i\rangle, U|\psi_i\rangle\}_{i=1}^L$, $|\psi_i\rangle \in \mathcal{H}$, $UU^\dagger = U^\dagger U = \mathbb{1}_\mathcal{H}$ learn to maximize the fidelity of the output with $U|\phi\rangle$ for any input $|\phi\rangle$, see Fig. 2.39.

$$|\psi_i\rangle \longrightarrow \boxed{U} \longrightarrow U|\psi_i\rangle$$

Figure 2.39.: The learning objective is to reproduce whatever the black box box does—unitary evolution with U.

Of course, if $V = \mathrm{span}\left(|\psi_i\rangle_{i=1}^L\right) \neq \mathcal{H}$, any learning algorithm can only guess the correct output for any input $\in V^\perp$. If U as well as $\{|\psi_i\rangle\}$ are assumed to be random and uniformly distributed, the optimal fidelity for the best unitary possible which exploits all the available information can be estimated as $F \sim \frac{L}{N} + \frac{N-L}{ND(D+1)}\left(D + \min\left\{L^2 + 1, D^2\right\}\right)$, where $D = \dim \mathcal{H}$ and N is the number of test pairs. The authors of [56] have trained a ⬚ network, denoted by $[3, 3, 3]$. Impressively, the trained QNN is able to closely match this theoretical prediction, see Fig. 2.40.
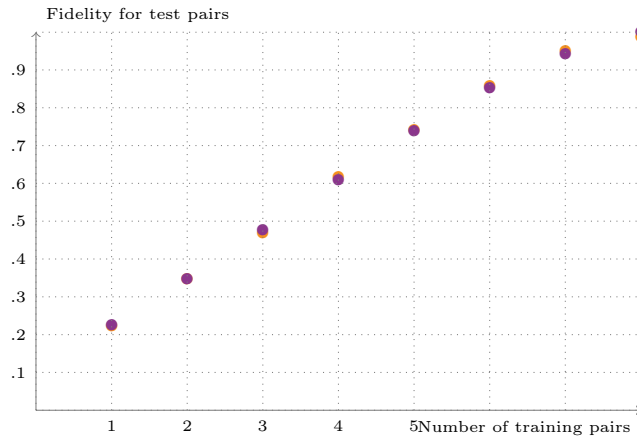
Figure 2.40.: A $[3, 3, 3]$ network trained for 1000 rounds with $n = 1, 2, \ldots, 8$ training pairs and evaluated the cost function for a set of 10 test pairs afterwards. The results are averaged over 20 rounds (orange points) and compared to the estimated value of the optimal achievable cost function (violet points).

For more examples, see e.g. [85, 7, 58].

In this thesis we investigate networks with layers of variable width, see Section 4. Conveniently, the neuron discussed in this section is the only building block required for this setting.

The authors of [56] provide an open-source implementation of QNNs written in several languages available at [2].

3. Parent Lindbladians for matrix product density operators

In this chapter we show how to dissipatively engineer sources for matrix product density operators (MPDOs).

We present an algorithm that determines whether there exists a parent Lindbladian for a given space of MPDOs and, if so, outputs such a Lindbladian (see Fig. 3.1). It turns out that the desired Lindbladian exists if and only if a local frustration-free quantum channel with given MPSOs as a fixed space exists. Sections 3.2 and 3.3 are devoted to different steps of the algorithm – to the construction of the local terms in the Lindbladian and to ensuring that the sum of the local terms does not have unwanted stable states. Examples are contained in Section 3.4.

Figure 3.1.: MPS are connected to ground states of local gapped 1d Hamiltonians. MPS can approximate such ground states well, and each MPS is the unique ground state of a parent Hamiltonian. MPDOs and Lindbladians are a direct generalization of MPS and Hamiltonians to mixed states. MPDOs can represent thermal states for high enough temperature and are a well-established variational method for stable states of open-system dynamics. But do MPDOs have parent Lindbladians?

As was discussed in Subsection 2.4.4, MPDOs are at the core of powerful numerical methods suitable for 1d open systems. MPDOs can represent thermal states of local Hamiltonians and are a direct generalization of MPS (see Subsection 2.4.2) to mixed states.

Ground states of local gapped 1d Hamiltonians are well approximated by an MPS (see Theorem 2.4.2). Moreover, every MPS is a unique ground state of a frustration free gaped local Hamiltonian—a parent Hamiltonian (see Subsection 2.4.3).

One way to prepare a state of interest is to design a dissipative evolution leading in the long

term only to the desired states, see Subsection 2.5.7. A Lindbladian describes the evolution in the Markovian case (see Section 2.2). Dynamics consisting of the parent Hamiltonian and energy dissipation provide a recipe for dissipative preparation for a given MPS [334, 598].

We seek to extend parent Hamiltonian construction to mixed states. We look for parent— local and frustration free, Lindbladians with stable space consisting only of given MPDOs. As achievable locality is often limited in experiments, we concentrate on k-local Lindbladians. This construction can be used to design preparation protocols for MPDOs, provide a way to invent new integrable systems and give insight into what kind of systems are well described by this variational class.
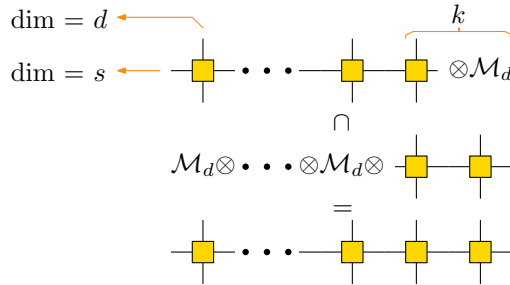
3.1. Outline of the algorithm

There are two important steps in constructing k-local parent Lindbladians:

1. For the Lindbladian $\mathcal{L} = \sum_i \mathcal{L}_{i,\dots,i+k-1}$, construct a local term $\mathcal{L}_{i,\dots,i+k-1}$ with stable space
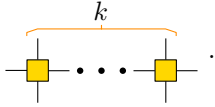
2. Check that $\mathcal{L} = \sum_{i=1}^{L-k-1} \mathcal{L}_{i,\dots,i+k-1} \equiv \sum_i \mathcal{L}^i$ does not have any additional stable states:

Here $\mathcal{M}_d$ denotes an algebra of $d \times d$ complex matrices.

Any of these two steps may return a negative result, i.e. either $\mathcal{L}_{i,\dots,i+k-1}$ or $\mathcal{L}$ have extra stable states in addition to desired stable state space of MPDOs. This means that there is no frustration free k-local parent Lindbladian for a given space of MPDOs.

3.2. Constructing local term in a Lindbladian

In this section we construct a local term in Lindbladian such that a given length-k matrix product operators (MPOs) (see the first step in 3.1) form the stable space of such term or determine that such local term does not exist. Note that given a length-L MPDO, the length-$k < L$ MPOs are not necessary density operators.

To construct a local term with a given stable space $\mathcal{F}$ we first construct a quantum channel with the fixed space $\mathcal{F}$. For every CPTP map T there exists a Lindbladian $\mathcal{L}$ such that $T[\rho] =$

$\rho \Rightarrow \mathcal{L}[\rho] = 0$, for example defined via

$$\mathcal{L}[\rho] = \text{const} \cdot (T[\rho] - \rho) \text{ for some const} > 0. \tag{3.1}$$

Moreover, for the k-local and frustration free Lindbladian $\mathcal{L}$, namely

$$\mathcal{L} = \sum_i \mathcal{L}_{i,\dots,i+k-1} \text{ and } \mathcal{L}[\rho] = 0 \iff \forall i \ \mathcal{L}_{i,\dots,i+k-1}[\rho] = 0, \tag{3.2}$$

it is possible to associate a k-local CPTP map T via

$$T = \frac{1}{L-k-1} \sum_{i=1}^{L-k-1} T_{i,\dots,i+k-1}, \text{ where } T_{i,\dots,i+k-1} = e^{\mathcal{L}_{i,\dots,i+k-1}}, \tag{3.3}$$

such that $\mathcal{L}[\rho] = 0 \Rightarrow T[\rho] = 0$. This means that a construction of a parent Lindbladian and a question of its existence can be concluded from the study if the space of interest is a fixed space of a quantum channel.

A fixed space of a quantum channel T is of the form (see Subsection 2.1.3)

$$\mathcal{F}_T = U \left(\mathbb{0}_{d_0} \oplus \bigoplus_{k=1}^{K} (\rho_k \otimes \mathcal{M}_{d_k}) \right) U^\dagger, \quad UU^\dagger = U^\dagger U = \mathbb{1}, \quad \rho_k \geq 0. \tag{3.4}$$

Let the space of k-local MPOs starting from site i be spanned by matrices $\{O_b^i\}$, where different values of $b = \{l, r\}$ corresponds to different boundary conditions:

Our goal is to find the smallest fixed space $\mathcal{F}$ such that $\forall b \ O_b^l \in \mathcal{F}$. If $\dim(\mathcal{F}) = \dim\left(\text{span}\left[\{O_b^i\}\right]\right)$, the local CPTP term $T_{i,\dots,i+k-1}$ can be chosen as a projection on $\mathcal{F}$; otherwise no k-local parent Lindbladian does not exist, as for each local term the stable space is too large. Note that fixed space can be spanned by self-adjoint operators. Thus, it is sufficient to work with self-adjoint linear combination of MPOs, e.g. $\left\{O_j + O_j^\dagger, iO_j - iO_j^\dagger\right\}$.

The algorithm presented below determines the smallest fixed space $\mathcal{F}$ of any CP map such that a given set of operators are in $\mathcal{F}$. This algorithm uses subroutines discussed in Subsection 2.1.4. An alternative algorithm that can use any simultaneous block diagonalization subroutine can be found in Appendix A.

Algorithm 3 Pseudocode for the algorithm that finds the smallest fixed space of a quantum channel that contains a given set of states.

Input: A set of matrices $\{O_b^i\}$.

Output: With probability 1, the smallest fixed space (see Subsection 2.1.3) $\mathcal{F}^i$ such that $\forall b\, O_b \in \mathcal{F}^i$. With probability 0, an algebra containing $\mathcal{F}$.

1: Find self-adjoint linear independent matrices $\{S_i\}$ such that $\mathrm{span}\,(\{\zeta_i\}) = \mathrm{span}\,(\{O_i\})$.
2: Run the Algorithm 1 from Subsection 2.1.4. Store the output Q and optional outputs $\{d_i\}, \Sigma$.
3: Let $D_i = \sum_{j=0}^{i} d_i$. Construct a matrix

$$
V = \mathrm{diag}\left(\{1\}_{k=1}^{D_0}, \left\{ \sqrt{\frac{1}{\sum_{i=D_0+1}^{D_1} \Sigma_{ii}^2}} \right\}_{k=1}^{D_1}, \cdots, \left\{ \sqrt{\frac{1}{\sum_{i=D_j+1}^{D_{j+1}} \Sigma_{jj}^2}} \right\}_{k=1}^{D_j}, \cdots \right). \tag{3.5}
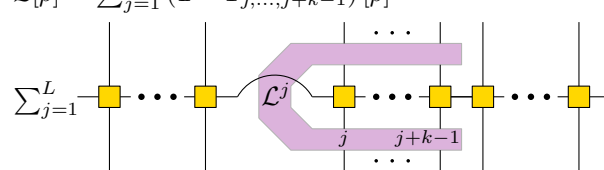$$

4: Run the Algorithm 2 from Subsection 2.1.4 with the input $\{VQ^\dagger \zeta_b Q\}$ including the optional step. As Algorithms 2 and 1 share first few steps, use the results already computed during step 2 of this algorithm. Obtain the set $\left\{ U, \{d_j\}_{j=0}^{J}, \{k_j\}_{j=1}^{J} \right\}$.
5: Output the set $\left\{ U, V, \{d_j\}_{j=0}^{J}, \{k_j\}_{j=1}^{J} \right\}$, where $\mathcal{F}^i \equiv UV^{-1}\left(0_{d_0} \oplus \bigoplus_{j=1}^{J} \left(\mathbb{1}_{k_j} \otimes \mathcal{M}_{d_j} \right) \right) U^\dagger$.

Let us run this algorithm for the space $\mathcal{F}^i$ of matrices that span the given length-k MPO to get $\mathcal{F}_{T_{i,\dots,i+l-1}}$. If the dimension of $\mathcal{F}_{T_{i,\dots,i+l-1}}$ equals to the dimension of $\mathcal{F}^i$, than we can choose $T_{i,\dots,i+l-1}$ as a projector on $\mathcal{F}_{T_{i,\dots,i+l-1}}$. The local term in a Lindbladian can be chosen as $\mathcal{L}_{i,\dots,i+l-1} = \mathbb{1} - T_{i,\dots,i+l-1}$. Otherwise, there is no k-local Lindbladian such that its stable space is the desired space $\mathcal{F}^i$.

3.3. Patching local parts - dimension of a stable space

In this section we investigate what happens when k-local parts get patched together to form the space of arbitrary long MPDOs (see the second step in 3.1).

Let us investigate the k-local Lindbladian

$$
\mathcal{L}[\rho] = \sum_{j=1}^{L} \left(\mathbb{1} - T_{j,\dots,j+k-1} \right)[\rho] =
$$

where $T_{j,\dots,j+k}$ is a CPTP map that acts non-trivially only on k neighbouring qudits. By construction, any state that is in the desired stable space of MPDO states is a fixed space of any local CPTP map $T_{j,j+1,\dots,j+k-1}$. Our goal is to find all states that are in the fixed space of $T_{j,j+1,\dots,j+k-1}$ for every $j \in \{1, 2, \cdots, L\}$.

We use a technique inspired by [405]. Suppose we have a set of s_n linearly independent solutions for the first $l \geq k$ sites in the form

$$\rho^{[1,l]} \equiv \rho^{i_1 j_1, \ldots, i_l j_l}_{\alpha_0 \alpha_l} = R^{i_1 j_1, [1]}_{\alpha_0 \alpha_1} R^{i_2 j_2, [2]}_{\alpha_1 \alpha_2} \ldots R^{i_l j_l, [l]}_{\alpha_{l-1} \alpha_l}$$
$$|i_1, i_2, \ldots i_l\rangle\langle j_1, j_2, \ldots j_l|$$

with $i_n, j_n = 1, \ldots d$ and $\alpha_n = 1, \ldots, s_n$; here and below all the repeated indices are summed over. The R's satisfy the linear independence conditions

$$R^{i_n j_n, [n]}_{\alpha_{n-1} \alpha_n} x_{\alpha_n} = 0, \ \forall i_n, j_n, \alpha_{n-1} \ \Leftrightarrow \ x_{\alpha_n} = 0, \ \forall \alpha_n \tag{3.6}$$

We now add one more site to the chain and look for eigenvalue zero states of $\mathcal{L}$ in the form

$$
\begin{aligned}
\rho^{i_1 j_1, \ldots, i_l j_l}_{\alpha_0 \alpha_l} &= \rho^{i_1 j_1, \ldots, i_{l-k+1} j_{l-k+1}}_{\alpha_0 \alpha_{l-k+1}} \cdot \\
&\quad R^{i_{l-k+2} j_{l-k+2}, [l-k+2]}_{\alpha_{l-k+1} \alpha_{l-k+2}} \ldots R^{i_{l+1} j_{l+1}, [l+1]}_{\alpha_{l+1} \alpha_{l+1}}
\end{aligned}
\tag{3.7}
$$

The unknown tensor $R^{i_{l+1} j_{l+1}, [l+1]}_{\alpha_l \alpha_{l+1}}$ must satisfy a system of linear equations

$$C_{\sigma, i_{l+1} j_{l+1} \alpha_l} R^{i_{l+1} j_{l+1}, [l+1]}_{\alpha_l \alpha_{l+1}} = 0, \tag{3.8}$$

where $\sigma = \alpha_{l-k+1} i_{l-k+1} j_{l-k+1} \ldots i_l j_l$ and

$$
\begin{aligned}
\equiv C = \mathcal{L}^{l-k+1} \Big(&|i_{l-k+2}, \ldots, i_{l+1}\rangle\langle j_{l-k+2}, \ldots, j_{l+1}| \cdot \\
&R^{i_{l-k+2} j_{l-k+2}, [l-k+2]}_{\alpha_{l-k+1} \alpha_{l-k+2}} \ldots R^{i_l j_l, [l]}_{\alpha_{l-1} \alpha_l} \Big)
\end{aligned}
$$

is a matrix with dimension $r s_{l-k} \times d^2 s_l$. The number of solutions is exactly $s_{l+1} = d^2 s_l - \mathrm{rank}(C)$. In particular, this gives a necessary and sufficient condition for the whole stable space to be an space of MPDO form with bond dimension s, that is $\mathrm{rank}(C) = s(d^2 - 1)$.

3.4. Examples

Let us first illustrate the algorithm outlined in Section 3.1 on the simplest possible non-trivial sets of states. Consider a set of MPDOs

$$\rho(\{\alpha_j\}) = \left(\frac{\mathbb{1}_2}{2}\right)^{\otimes L} + \sum_{i \in J} \alpha_i \sigma_i^{\otimes L}, \tag{3.9}$$

where σ_i are Pauli matrices, $J \subset \{1, 2, 3\}$ and $\{\alpha_j\}_{j \in J}$ are numbers such that $\rho(\{\alpha_j\})$ is a state. The bond dimension of $\rho(\{\alpha_j\})$ can be chosen as $D = |J| + 1$ with the corresponding tensors

$$1 \; \square \; 1 \; = \; \frac{\mathbb{1}_2}{2} \qquad i \; \square \; i \; = \sigma_i.$$

Let us first look at the case $J = \{3\}$. There is a two-local parent Lindbladian for $\rho(\{\alpha_3\})$. Indeed, the first stage of our algorithm outputs a local term $\mathcal{L}_{i,i+1}$ as an identity minus the projector on the algebra $\mathbb{1}_2 \otimes (\mathcal{M}_1 \oplus \mathcal{M}_1)$ and the second stage proves that the sum of local terms has the same number of stable states as one local term.

However, if $|J| = 2$ there is no k-local parent Lindbladian for any k. Indeed, the space spanned by $\mathbb{1}^{\otimes k}, \sigma_i^{\otimes k}, \sigma_j^{\otimes k}$ does not form an algebra for $i \neq j$, and already the first stage of the algorithm outputs that no desired Lindbladian exists. This situation is rather generic for MPDOs and is drastically different to the case of MPS, where a parent Hamiltonian exists for large enough k, see Theorem 2.4.7.

Beforehand we obtained a parent Lindbladian for a set of density matrices that can be simultaneously diagonalised. If we now look at the case $|J| = 3$, the local term of a Lindbladian will be an identity minus the projector on the non-commuting algebra $\mathbb{1}_2 \otimes \mathcal{M}_2$.

Similar examples can be constructed using higher-dimensional representations of $su(2)$.

We will now look at the thermal states of local Hamiltonians, $\rho = \frac{e^{-\beta H}}{\mathrm{tr}[e^{-\beta H}]}$. Let us start with the Ising model $H = \sum_i \sigma_z^i \sigma_z^{i+1}$. This state can be described as a MPDO with bond dimension 2:

We are interested in the parent Lindbladian of the space of MPDOs that correspond to the Ising model without the boundary terms $\begin{array}{c}\square \cdots \square\end{array}$. Let us first try to construct the two-local Lindbladian. The local term can be chosen as an identity minus the projector on the algebra $\bigoplus_{i=1}^4 \mathcal{M}_1$. However, the second stage of the algorithm shows that $\mathrm{rank}(C) = 4 < 6$ and thus, no 2-local parent Lindbladian exists. The situation changes when we increase the locality - the algorithm finds the 4-local parent Lindbladian for the Ising model as a sum of projectors on algebra $U \left(\bigoplus_{i=1}^4 (\tilde{\rho}_i \otimes \mathcal{M}_1) \right) U^\dagger$. Here $\tilde{\rho}_1 = \mathrm{diag}(e^{-8b}, 1, 1, 1)$, $\tilde{\rho}_2 = \mathrm{diag}(1, e^{8b}, 1, 1)$, $\tilde{\rho}_3 = \mathrm{diag}(1, 1, e^{8b}, 1)$, $\tilde{\rho}_4 = \mathrm{diag}(1, 1, 1, e^{-8b})$, $U_{ij} = 1$ if $j = 2i - 1$ or $i = 8 + j/2$ and 0 otherwise.

While the Ising model plays an important role in physics, it is classical. One way to proceed is too build excitations on top of an entangles state, the other approach is to add non-commuting terms to the Hamiltonian. Let us first proceed with the later option. The Hamiltonian for the Ising model in a transverse field is $H = \sum_i \sigma_z^i \sigma_z^{i+1} + h\sigma_x^i$. The thermal state of this model is entangled, but it might be not so straightforward to write it down in a MPDO form. Thus, we will study the first Suzuki-Trotter decomposition of the thermal state. There are two obvious possibilities for such decompositions: $\rho_h = \frac{1}{Z_h} \left(e^{-\beta \sum_i \frac{h}{2} \sigma_x^i} e^{-\beta \sum_i \sigma_z^i \sigma_z^{i+1}} e^{-\beta \sum_i \frac{h}{2} \sigma_x^i} \right)$ and $\rho_\beta =$

$\frac{1}{Z_\beta} \left(e^{-\frac{\beta}{2} \sum_i \sigma_z^i \sigma_z^{i+1}} e^{-\beta \sum_i h \sigma_x^i} e^{-\frac{\beta}{2} \sum_i \sigma_z^i \sigma_z^{i+1}} \right)$, where Z_h and Z_β are normalizations. If we denote $\varepsilon = e^{-\beta \frac{h}{2} \sigma_z}$, we can draw the corresponding tensors as
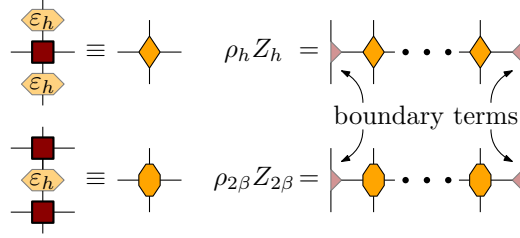

We are interested in the space of MPDOs without the boundary therms.

However, the ρ_h is classically correlated, as the bond dimension is 2.[152] For the parameters tested ($\beta = 1, h \in \{1/10, 1/2, 1, 2, 10\}, k \leq 6$) neither of these MPDOs have parent Lindbladians. Every time the first step of the algorithm failed - the stable space of a local Lindbladian is too large.

The other option to generate a quantum model related to Ising model is to start with any wave function $|\psi\rangle$ spanned by $|0\rangle^{\otimes L}$ and $|1\rangle^{\otimes L}$ and add domain walls at each site with probability p – for every n starting from 1 to L we define

$$\rho_{n-1} = (1-p)\rho_n + p\mathbb{1}^{\otimes n} \otimes \sigma_x^{\otimes(L-n)} \rho_n \sigma_x^{\otimes(L-n)} \otimes \mathbb{1}^{\otimes n}$$

starting with $\rho_0 = |\psi\rangle\langle\psi|$. The space V_L of all ρ_L can be written in MPDO form as
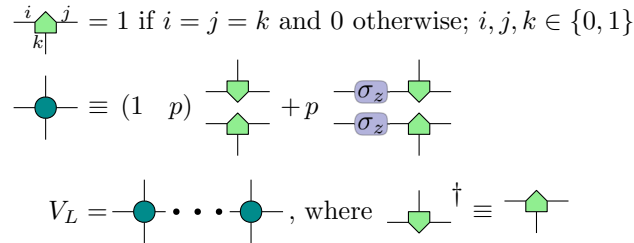
The state ρ_L can be entangled (for $p = 0$ the ρ_L can be GHZ). Different boundary conditions correspond to different choices of $|\psi\rangle$ from a 4-dimensional space and to mixtures of odd or even number of domain walls. Correspondingly, ρ_L is an element of 8-dimensional space. There is a 2-local parent Lindbladian MPDOs that give rise to ρ_L and it is a sum of identities minus the projectors on algebra $U(\mathcal{M}_2 \otimes \mathcal{M}_2)U^\dagger$, where U is a permutation $(2, 3, 1, 4)$ in one-line notation. Note that this algebra is non-commuting.

The code for all of the examples can be found at [3].

3.5. Conclusions and outlook

We developed an algorithm that determines if a given (small) linear space of MPDOs can be a stationary space for some k-local Lindbladian and, if so, outputs such a Lindbladian. Such Linbladian exist if and only if a local frustration-free quantum channel with given MPDOs as the

fixed space exists. This gives a recipe for an experimental preparation of MPDOs via dissipative engineering. It is also a good starting point to invent new integrable systems.

The most computationally expensive step of the presented algorithm is simultaneous block-diagonalization of s^2 matrices of dimension $d^k \times d^k$, where s is the bond dimension, and d is the dimensionality of the underlying elementary physical systems. It scales quadratically with the bond dimension, but exponentially with the locality of the desired Lindbladian.

The interesting remaining questions are what happens to the gap of such Lindbladian in thermodynamic limit and what is a sufficient locality parameter k such that a parent Lindbladian exists. Moreover, it would be useful to understand what kind of MPDOs have parent Lindbladian; it was conjectured by Sofyan Iblisdir that this question might be connected to ranks of different decompositions [154, 153]. It would also be interesting to study how the ideas from this work can be applied to the random MPDOs and random Lindbladians.

4. Quantum neural networks for denoising

Virtually every experimental preparation of a quantum state introduces noise. Usually, it is hard to design a denoising protocol. First, one has to identify and characterize all noise sources. Second, one has to invent a protocol that corrects the noise without affecting any relevant features of the quantum state. Machine learning (ML) can automate this task.

We use the quantum neural networks (QNNs) described in Section 2.8 to train denoisers for simulated quantum sources. As current quantum hardware is not fault-tolerant and has intrinsic noise (see Section 2.5), we verify that learning (in Section 4.1) and denoising (in Subsection 4.3.4) is possible even with imperfect devices.

For state-of-the-art experiments, there is usually no source that exhibits lower noise and can be used for supervision. Thus, in this chapter we focus on data sets that are obtained from a single source. Successful learning can be facilitated either by a clever data acquisition or by the use of network architectures appropriate for unsupervised learning.

While increasing the quality of an experimental output is hard, decreasing it is usually easy. In Section 4.2, we train a network to map from a parameter region of decreased performance to the region of best available performance. After training, the QNN is able to extrapolate from the region of available performance to a region of yet unseen quality. Due to the experimental relevance of thermal noise, we use this approach for extrapolation to lower temperatures.

In Section 4.3, we construct undercomplete quantum autoencoders (AEs) to combat shot-to-shot noise. Suppose that a tunable data source produces a certain class of states that are subject to random noise. Just like in classical AEs (see Subsection 2.6.4), a bottleneck allows quantum AEs to filter out irrelevant information and preserve only relevant features that are present in every member of a class. We use this technique to successfully denoise GHZ, W, Dicke and cluster states subject to spin-flips, dephasing errors and random unitary noise.

Finally, in Section 4.4 we teach a source to shield itself from noise by learning quantum error correction codes. This also does not require a supervising source—the training just has to ensure that the encoded and decoded states match.

A supervising source might be available if a post-processing device is trained for a source with significant constraints affecting its performance, such as cost, transportability or fast output rate. We discuss this situation in Chapter 5. There, we also study denoising protocols that require memory. In the present chapter we concentrate on protocols where feed-forward networks are sufficient.

See Fig. 4.1 for a schematic representation of the tasks studied in this chapter.
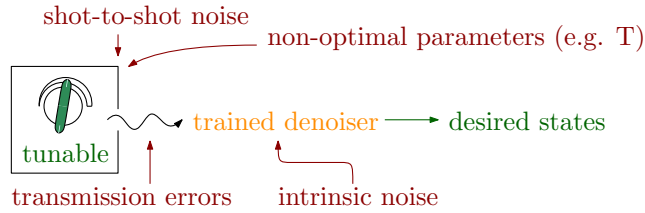
Figure 4.1.: The setting studied in Chapter 4. We use QNNs to train denoisers for quantum sources. The source may be tunable, but we do not assume access to an additional supervisory source. We concentrate on noise cancellation that does not require memory, such as dealing with non-optimal parameters, shot-to-shot noise and transmission errors. Denoising can be successful even if the QNNs are implemented on hardware with some intrinsic noise.

4.1. Are QNNs suitable for NISQ devices?

A practical ML protocol has to be robust to noise. Most data sets are not perfect. Thus, it is crucial to verify that QNNs can work even with some of the data is corrupted.

The cost function that must be evaluated for training can be measured only with finite precision. Moreover, current NISQ devices are by definition noisy. Thus, it is important to examine whether QNNs can still be useful if implemented on realistic devices.

This section shows that QNNs can tackle noisy data even when being implemented on a noisy device. The presented results were first obtained for a bachelor's thesis co-supervised by the author [516] and were included in [56].

4.1.1. Model for noisy neurons

We model the noise in the network by evolving the state with a random time-dependent Hamiltonian $H(\tau)$ before and after each operation. As a result the noisy unitary that corresponds to the jth neuron in the lth layer is

$$U_j^l(t) = R(t) U_j^l \tilde{R}(t), \tag{4.1}$$

where $R(t)$ and $\tilde{R}(t)$ are two different realisations of a random unitary generated by the same probabilistic process. The update rule is also modified into

$$U_j^l \to R(t) \left(e^{i\epsilon K_j^l} U_j^l \right) \tilde{R}(t). \tag{4.2}$$

The evolution with $H(\tau)$ can be constructed via a quantum Brownian circuit [652, 353]. Let us consider a family of Hamiltonians $\{H_j = H(j\Delta\tau)\}_{j=1}^n$, $\Delta\tau = T/n$, where every H_j is Hermitian and its entries are Gaussian distributed with zero mean and a standard deviation of $2\pi\hbar\nu\sqrt{\frac{n}{2^m}}$. The noise strength is captured by a dimensionless parameter $t = \nu T$. We model both $R(t)$ and $\tilde{R}(t)$ as

$$\overset{(\sim)}{R}(t) = \prod_{j=1}^n \exp(iH_j\Delta\tau/\hbar). \tag{4.3}$$

By Itô's calculus, there exists a $H(t)$ such that

$$\overset{(\sim)}{R}(t) = \mathcal{T} \exp\left(i/\hbar \int_0^T H(\tau)\, d\tau\right) + O\left(\frac{1}{\sqrt{n}}\right),\tag{4.4}$$

where $\mathcal{T}$ is the time ordering operator. We use $n = 20$. To assess the effect of this noise, we study the fidelity of random 3-qubit states $|\psi\rangle$ with $R(t)\tilde{R}(t)|\psi\rangle$, see Fig. 4.2.
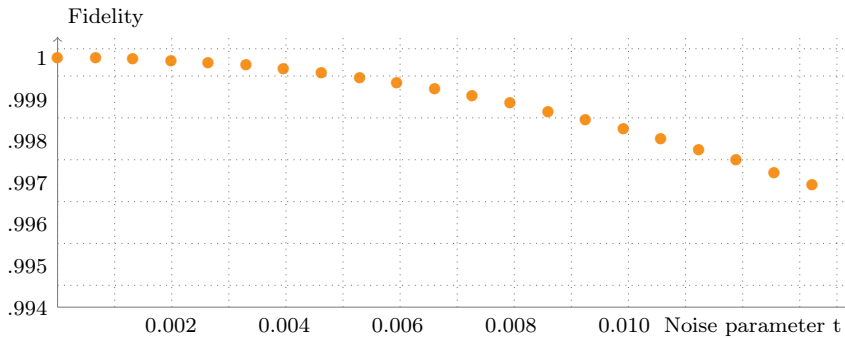
Figure 4.2.: Noisy identity operation. Relation between the fidelity $|\langle\psi|R(t)\tilde{R}(t)|\psi\rangle|^2$ and the noise strength t for 500 random 3-qubit states $|\psi\rangle$.

We return to Example 2.8.1 and study how the performance of the network deteriorates with noise. While the performance of the QNN indeed falls as the noise strength t increases, the change is smooth. The results for the $[3, 3, 3] \sim$ ▨▨▨ network can be seen in Fig. 4.3.
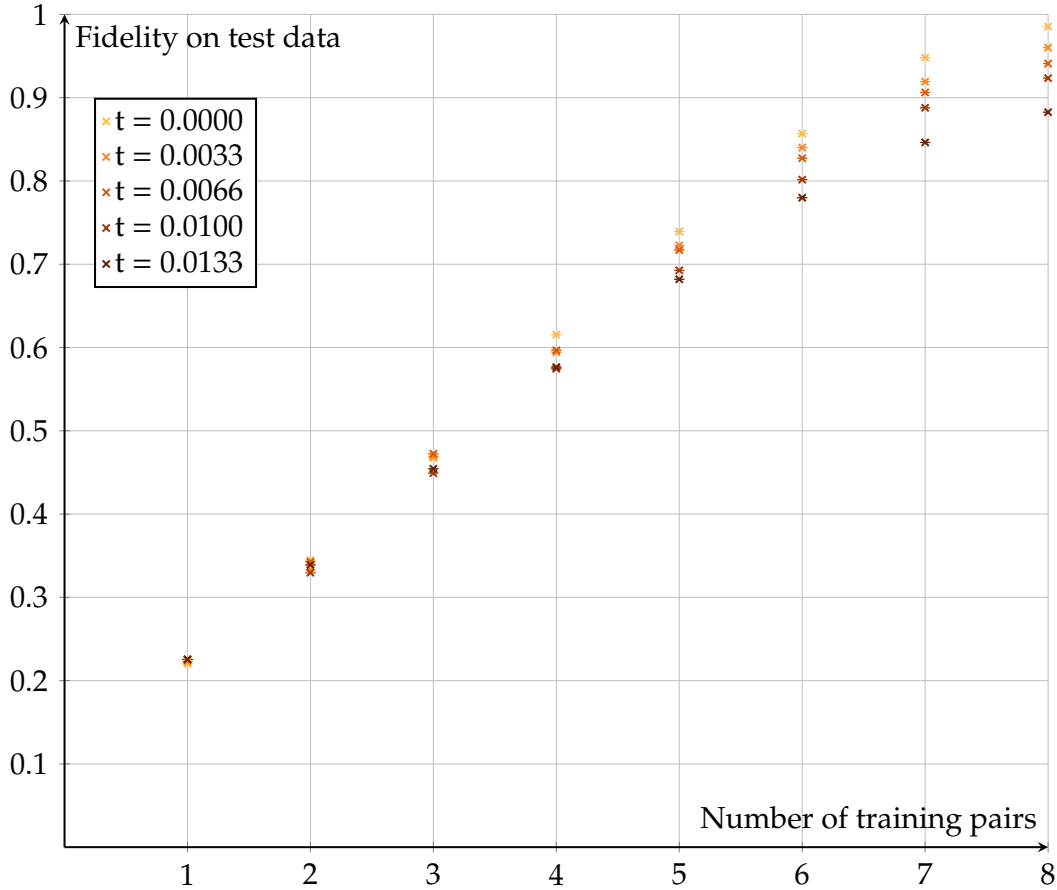
Figure 4.3.: A $[3, 3, 3]$ network intended to reproduce a unitary evolution, see Example 2.8.1; trained for 1000 rounds with $n = 1, 2, \ldots, 8$ training pairs and noise strength t. The cost function is evaluated for a set of 10 test pairs afterwards. The results are averaged over 20 rounds. Image taken from [516].

4.1.2. Model for corrupted data

We model corrupted data by substituting a subset of labels in the training set with random states.

We concentrate on learning unitary evolutions, see Example 2.8.1. The training data is $\{|\psi_i\rangle, |\phi_i\rangle\}_{i=1}^{L}$, $|\psi_i\rangle, |\phi_i\rangle \in \mathcal{H}$, where $|\phi_i\rangle$ is $U|\psi_i\rangle$ with probability $1 - p$ and a random pure state sampled from a uniform distribution with probability p. The objective is to maximize the fidelity of the output with $U|\phi\rangle$ for any input $|\phi\rangle$, see Fig. 2.39. We train the network $[2, 3, 2] \sim$ ⬡ with noisy neurons (see Subsection 4.1.1) for 300 rounds on 100 noisy training pairs. Figure 4.4 shows that noisy training data almost does not deteriorate the performance of the QNN up to $p \sim \frac{2}{3}$. As expected, as long as the fraction of corrupted training data is small, a stronger noise affecting the neurons decreases the fidelity on the test data. However, the latter relation can get reversed if a big fraction—around $\frac{9}{10}$—of the data is corrupted. If low-noise devices are available
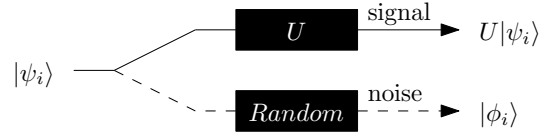
Figure 4.4.: We assume that the signal in data labels is produced by an unknown unitary evolution. With some probability the data is corrupted - the labels are random with a uniform distribution.

for a task with highly corrupted data, the optimal amount of noise can be achieved by injecting noise artificially, e.g. by modifying the update direction of gradient descent steps.
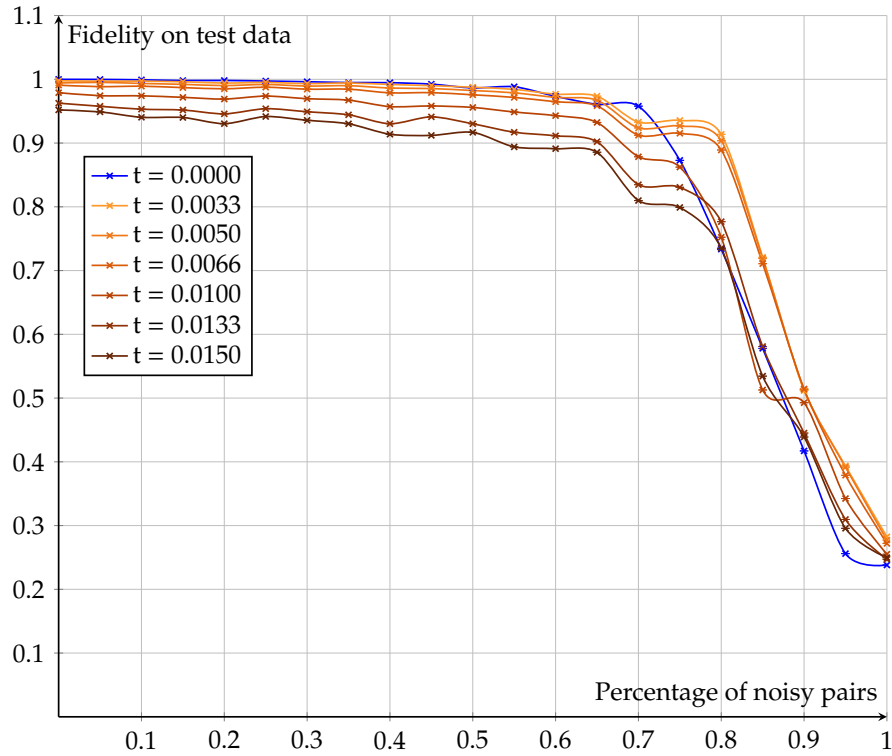
Figure 4.5.: Robustness of QNNs. A $[2, 3, 2]$ network trained for 300 rounds with 100 training pairs, where some of the labels were replaced by random uniformly distributed pure states. The neurons were subject to approximately depolarizing noise with noise parameter t (see Subsection 4.1.1). Image taken from [516].

4.1.3. Conclusion

We conclude that QNNs, for example the ones proposed in [56], can be successfully implemented on NISQ devices. There are intermediate-scale learning tasks for which one can still train networks of imperfect neurons and obtain valuable results even from very noisy data. Moreover, we

observe that the imperfections of neurons are less of an issue if the available data is very noisy. This leads to the speculation that NISQ devices are especially promising for ML task where perfect data is not available.

4.2. Extrapolation of parameters

In this section we work with sources that produce states that can be described by a parameter. We deal with the situation that the source produces states in an undesired parameter region. We use QNNs trained on the undesired parameter interval to extrapolate quantum states to more desired parameter values. While the techniques discussed in this section are applicable to any parameter of interest, we concentrate on the case of temperature.
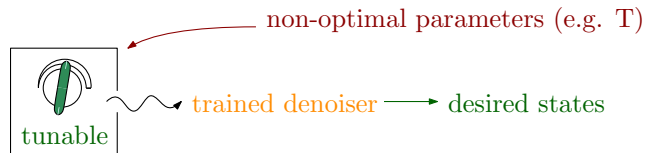

It should be noted that extrapolation is a hard [47] and actively researched [582, 51, 384] problem for classical neural networks (NNs). Addressing this task, NNs tend to suffer from overfitting [294] or to generate oversimplified predictions [249]. Sometimes, this oversimplification is a provably generic behaviour [644]. There are alternative methods for extrapolating properties of quantum systems via classical ML, for example Gaussian processes [587, 588]. However, there are classes of functions that NNs can provably extrapolate; moreover, there are subfamilies of NNs that do not seem to suffer from the aforementioned drawbacks [644]. NNs have been successfully used to extrapolate properties of systems as complex as nuclei [294, 418].

The results in this section were produced as part of a bachelor's thesis [179] supervised by Tobias Osborne and the author and have greatly benefited from a collaboration with Robert Salzmann.

We assume that a source can prepare states in a parameter range $[T_{\min}, T_{\max}]$ and that it is desirable to obtain states with parameters $T < T_{\min}$. We choose a function $f(T) < T$ and generate training data as pairs of parametrised states

$$\left\{ \rho\left(T_{in}^i\right), \rho\left(f\left(T_{in}^i\right)\right) \right\}_{i=1}^{N_{\text{data}}} \tag{4.5}$$

such that $\forall i \ f\left(T_{in}^i\right) \in [T_{\min}, T_{\max}]$. We proceed by choosing $f(\cdot)$ and training an FFQNN $\mathcal{N}$ on this data set so that the lower part of the available parameter region is extrapolated by $\mathcal{N}$ to states inaccessible to the source, see Fig. 4.6. Thus, the goal is that

$$\exists T_l, T_h : \ T_{\min} \in [T_l, T_h], \ \forall T \in [T_l, T_h] \ \exists \tilde{T} < T : \mathcal{N}\left(\rho(T)\right) \approx \rho(\tilde{T}). \tag{4.6}$$
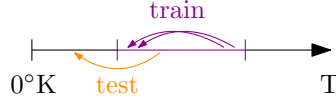
Figure 4.6.: We assume that the source is able to produce states in some temperature interval. We train a QNN to cool states inside this interval. The network should learn to extrapolate, that is input states from the lower-temperature part of the available interval are mapped to colder states that are outside of the training interval.

We choose $f(T) = \gamma T$ with $\gamma < 1$. This choice is, as of now, not well-motivated. It is worthwhile to check for which choices of f the corresponding quantum channel actually exists, see [234, Theorem 1 and Corollary 1] for a starting point. We set $\gamma = 0.8$. For numerical examples with different γ, see [179, Section 7.3]. Note that $\gamma > 1$ correspond to heating. Unsurprisingly, heating is easier to achieve than cooling.

There is a long history of attempts to produce colder states. Thermal noise is an undesirable feature in many experiments (see e.g. [515]), and decreasing the temperature tends to increase the sensitivity [312, 626, 280, 538]. Low enough temperatures can reveal novel phases of matter, exemplified by the discovery of superconductivity [298] and superfluidity [299, 23]. Reaching the extreme cold needed for this can require hard work of multiple generations. For example, the prediction of Bose-Einstein condensation (BEC) in weakly interacting atomic gases was published in 1925 [180], yet pioneering observations in ^{23}Na [148], ^{7}Li [94] and ^{87}Rb [28] occurred only in 1995 (to be fair, BECs in superfluid helium and semiconductor excitons [363, 239] were obtained earlier). Nevertheless, even judging by the few examples listed above, this hard work often pays off.

We concentrate on thermal states $\rho = \frac{e^{-\frac{H}{T}}}{\text{Tr}\left[e^{-\frac{H}{T}}\right]}$ of the transverse-field Ising model

$$H = H\left[\{J_{ij}, h_i\}\right] = \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^x, \quad J_{ij}, h_i \in \mathbb{R}. \tag{4.7}$$

This is the simplest and, via D-Wave [87], currently the only commercially available model used in quantum adiabatic computers (see Subsection 2.5.7). The transverse-field Ising model is not known to be QMA-complete, yet it is a complete problem of a complexity class that contains classical NP [98]. If $J_{ij} = J\delta_{j,i+1}$ and $h_i = h$, this system exhibits in the thermodynamic limit a quantum phase transition at $J = h = 1$ [453, 186].

As the labels in our training data are thermal states and mixed, our initial choice for the cost function is

$$C_{\text{HS}}\left(\left\{\rho\left(T_{in}^i\right), \rho\left(f\left(T_{in}^i\right)\right)\right\}_{i=1}^{N_{\text{data}}}\right) = 1 - \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \|\mathcal{N}\left(\rho\left(T_{in}^i\right)\right) - \rho\left(f\left(T_{in}^i\right)\right)\|_2^2, \tag{4.8}$$

see Subsection 2.8.2 for a discussion of alternative cost functions.

To evaluate the quality of the extrapolated states, we employ a function that compares a given output to the thermal state with temperature T.

Definition 4.2.1 (Comparison function). *For a given state ρ, the comparison function $F_\rho^{comp}(T)$ is defined via*

$$F_\rho^{comp}(T) = \mathcal{D}\left(\rho, \frac{e^{-\frac{H}{T}}}{\text{Tr}\left[e^{-\frac{H}{T}}\right]}\right), \tag{4.9}$$

where $\mathcal{D}$ is some distance measure. We pick the fidelity-based distance measure $D(\rho, \sigma) = 1 - \left(\text{Tr}\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2$.

Note that the comparison function is not used for training or validation, but only for the visualization. While the choice of distance measure in the cost function is relevant for the performance, the one employed in the comparison function is not; we could as well use the Hilbert-Schmidt norm both times.

4.2.1. Need for mini-batches

We consider sources that can provide training data with input states in the temperature region $T_{in}^i \in [T_{\min}/\gamma, T_{\max}] = [4, 8]$. In other words, we assume that the temperature region $[T_{\min}, T_{\max}] = [3.2, 8]$ is accessible. Temperatures that are significantly below $\gamma T_{\min} = 2.56$ lead to numerical instabilities in our simulation, while temperatures significantly above $T_{\max} = 8$ yield differences in the cost function that are too small for successful learning. We show results for two qubits with $J = J_{12} = 1$ and $h = h_1 = h_2 = 1$. We do not observe any significant sensitivity to the Hamiltonian parameters in the training outcomes, with a somewhat better performance for $h/J > 1$. Similar results were obtained for $J, h \in \{1, 2, 10\}$ in [179, Section 7.4].

Unlike for the other tasks considered in this thesis, the batch gradient descent is not suitable for cooling thermal states. While the cost function rapidly decreases (see Fig. 4.7), the learning outcomes tightly cluster around a value within the label parameter range, see Fig. 4.8.
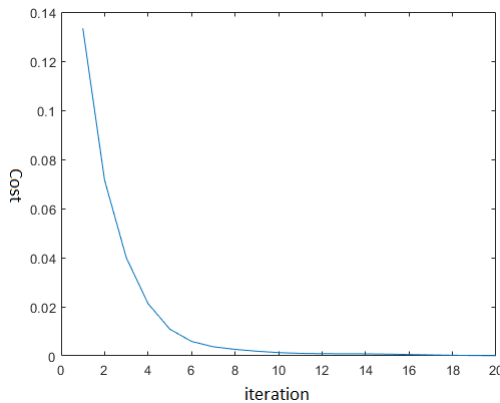

Figure 4.7.: The cost function while training a $[2, 2, 2]$ network on 1000 data pairs with the batch gradient descent. Figure taken from [179].

This clustering suggests that it can be beneficial to use mini-batches. Indeed, the concentration around a single point is not a low-cost solution if the training minimizes a cost function for

Figure 4.8.: The comparison functions (see Definition 4.2.1) for the post-processed states (black to lime) and for the corresponding label states (red to pink). Left: a $[2, 2, 2]$ network trained with 1000 data pairs for 20 training rounds. Right: a $[2, 5, 2]$ network trained with 1000 data pairs for 1000 training rounds. Figure taken from [179].

two sets of states belonging to parameter regions centered at separate points. For every training mini-batch i we fix ΔT_{mb_i} and randomly choose T_{mb_i} according to a uniform distribution on $[T_{\min}/\gamma, T_{\max} - \Delta T_{\mathrm{mb}_i}]$. We generate training data such that the input temperatures are uniformly distributed on

$$[T_{\mathrm{mb}_i}, T_{\mathrm{mb}_i} + \Delta T_{\mathrm{mb}_i}]. \tag{4.10}$$

Mini-batches indeed slightly improve the performance. However, the outputs are still too clustered, see Fig. 4.9. This can be attributed to the last mini-batch having the highest impact on the end result of the training, as can be observed from Fig. 4.10. It can be resolved by adjusting the training hyperparameters of the gradient descent for each mini-batch, as we discuss in Subsection 4.2.2.
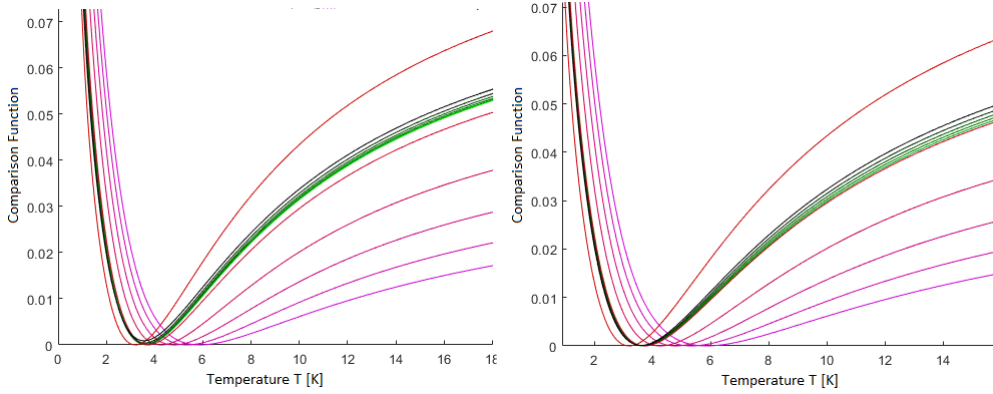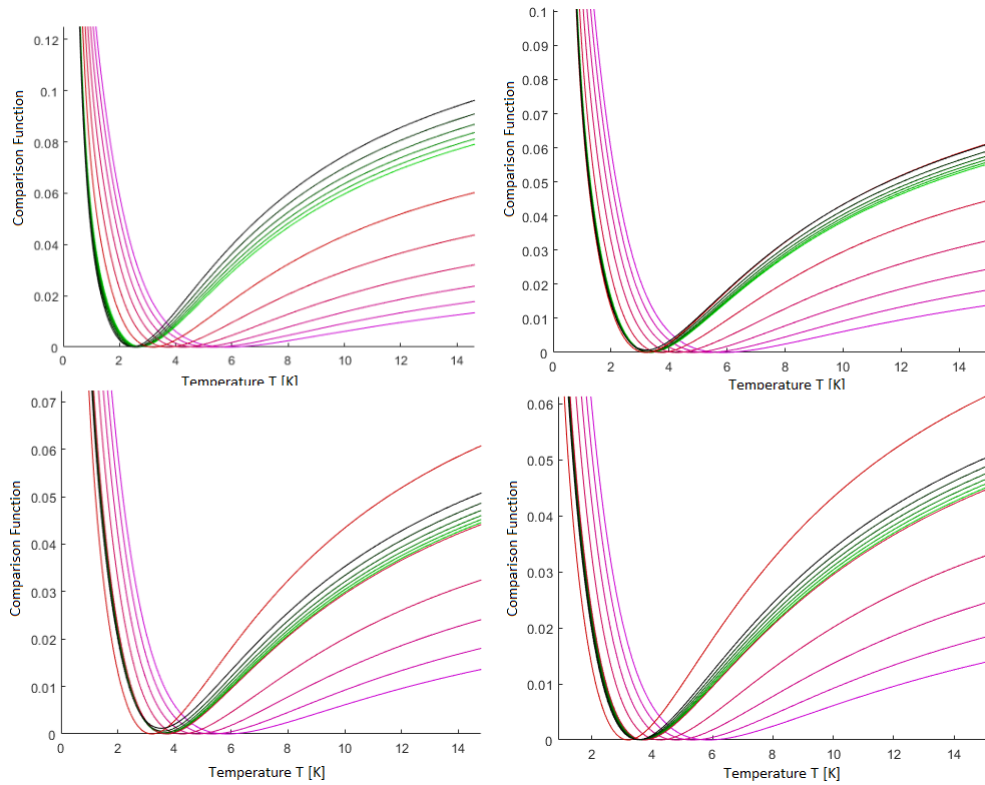
Figure 4.9.: The comparison functions (see Definition 4.2.1) for the post-processed states (black to lime) and for the corresponding label states (red to pink) for the examples listed in Table 4.1. Top left: mb1. Top right: mb2. Bottom left: mb3. Bottom Right: mb4. Figure taken from [179].
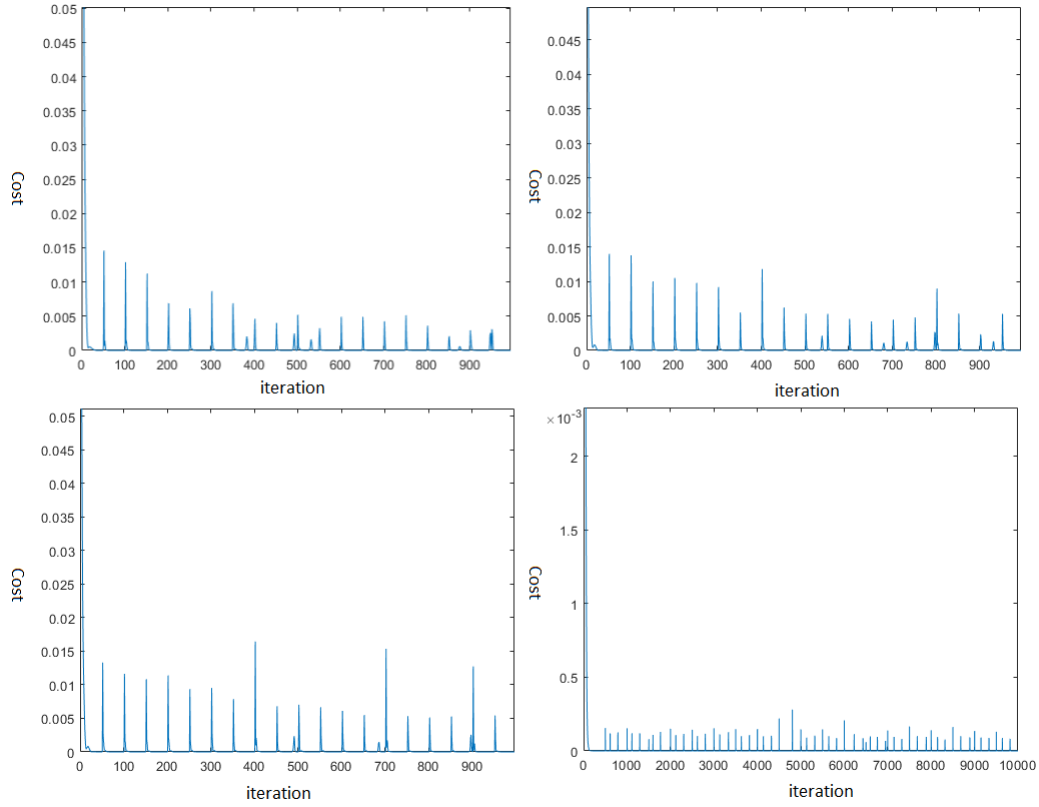
Figure 4.10.: The cost function during training for the examples listed in Table 4.1. The spikes stem from the sharp increase of the cost function when training with a new mini-batch begins. This signals that the training overfits to the current mini-batch. Top left: mb1. Top right: mb2. Bottom left: mb3. Bottom Right: mb4. Figure taken from [179].

	Example			
Parameter	mb1	mb2	mb3	mb4
ΔT_{mb}	1	2.5	4	4
Training pairs per mini-batch	6	6	6	60
Number of mini-batches	20	20	20	20
Training rounds per mini-batch	50	50	50	500
Network	[2 2 2]	[2 2 2]	[2 2 2]	[2 5 2]

Table 4.1.: Parameters for the training with mini-batches. Comparison functions for post-processed states are depicted in Fig. 4.9 and the cost function during training in Fig. 4.10.

4.2.2. Decreasing learning rate, recursive cost and random batches

A smaller learning rate can help to avoid the overfitting to the latest mini-batch, which we have observed in Fig. 4.10, but the optimization time significantly increases. Even more disappointingly, small learning rates can cause the optimization to get stuck in a shallow local minimum without converging even for a single mini-batch.

A good strategy to enjoy the benefits of both a large and a small learning rate is to make it mini-batch dependent and decrease it as training progresses. We use the Nadam optimizer and a learning rate η_j that polynomially decreases (see Equations 2.69) with the mini-batch index j:

$$\eta_j = \frac{1}{\frac{1}{\eta_{\max}} + \left(\frac{j}{N_{\text{mb}}}\right)^p \left(\frac{1}{\eta_{\min}} - \frac{1}{\eta_{\max}}\right)}, \tag{4.11}$$

where N_{mb} is the number of mini-batches and $\eta_{\min}$, $\eta_{\max}$ and p are hyperparameters.

We add an extra term to the cost function. Instead of the cost $C\left(\left\{\rho\left(T_{in}^i\right), \rho\left(f\left(T_{in}^i\right)\right)\right\}_{i=1}^{N_{\text{data}}}\right)$, we use what we call a recursive cost C^k of depth k,

$$
\begin{aligned}
C^k\left(\left\{\left\{\rho\left(T_{in,j}^i\right)\right\}_{j=0}^k\right\}_{i=1}^{N_{\text{data}}}\right) &= \frac{1}{k}\left[C\left(\left\{\rho\left(T_{in,0}^i\right), \rho\left(T_{in,1}^i\right)\right\}_{i=1}^{N_{\text{data}}}\right)\right. \\
&\quad + C\left(\left\{\mathcal{N}\left(\rho\left(T_{in,0}^i\right)\right), \rho\left(T_{in,2}^i\right)\right\}_{i=1}^{N_{\text{data}}}\right) + \ldots \\
&\quad \left. + C\left(\left\{\mathcal{N}\circ\cdots\circ\mathcal{N}\left(\rho\left(T_{in,0}^i\right)\right), \rho\left(T_{in,k}^i\right)\right\}_{i=1}^{N_{\text{data}}}\right)\right],
\end{aligned}
\tag{4.12}
$$

$$\text{where}\quad T_{in,j}^i = f\left(T_{in,j-1}^i\right), \quad j = 1,\ldots,k, \quad \text{and}\quad T_{in,0}^i = T_{in}^i. \tag{4.13}$$

The recursive cost of depth k requires the ability to produce $\left\{\left\{\rho\left(T_{in,j}^i\right)\right\}_{j=0}^k\right\}_{i=1}^{N_{\text{data}}}$ as training data. If the input parameter range remains unchanged, C^k requires a wider temperature range to be accessible to the source. If an efficient algorithm to compute C exists, C^k can also be computed efficiently. Note that this construction is not limited to the particular cost function used.

While the recursive cost gives a mild increase in training performance, the main benefit of this technique is that it facilitates overfitting detection. Each term in the cost function is computed separately, such that we have access to each of the $C^{j\leq k}$. If C^2 is much larger than C^1, it is a sign of overfitting. This greatly simplifies hyperparameter tuning. We use a depth of $k = 2$.

We proceed by randomizing the size of the mini-batch temperature intervals $\{\Delta T_{\text{mb}_j}\}$. The intuition behind this is two-fold. First, different widths of temperature regions have different benefits and drawbacks. Large regions suffer less from overfitting, yet smaller regions provide better learning outcomes for the states with parameters in this region. Second, if the learning rate is not too large, the overfitting to the data seen last is reduced if consecutive mini-batches are substantially different.

Combining all the techniques listed in this section, we achieve the learning outcomes depicted in Fig. 4.11. While the outputs of the QNN do not match the labels perfectly, the resemblance is close enough to hope for a successful extrapolation. To verify this hope, we first need to check that the outputs of the QNN are close to thermal states. This can be done by observing whether

the minima of the comparison functions 4.2.1 for different outputs of the trained network get close to zero. In Fig. 4.12 we see that the minimum values of the comparison functions are either on the order of 10^{-3} or smaller even outside the temperature region the QNN was trained on. Thus, to each output of the network we can assign the temperature corresponding to the value of T where the comparison function reaches its minimum. By comparing the temperatures of the input states with the temperatures assigned to the outputs, we confirm that the ⬖ $\sim [2, 5, 2]$ QNN successfully learned to extrapolate to colder temperatures, see Fig. 4.13.

We observe that the extrapolation performance deteriorates as the width of the network decreases. Unfortunately, the performance also deteriorates as the depth **increases**. This is due to the problem of outcome clustering described in Subsection 4.2.1. Each extra layer tends to increase clustering (as also happens in extrapolation with classical NNs [294]). We conjecture that random dropping out connections and neurons during training can solve this problem.
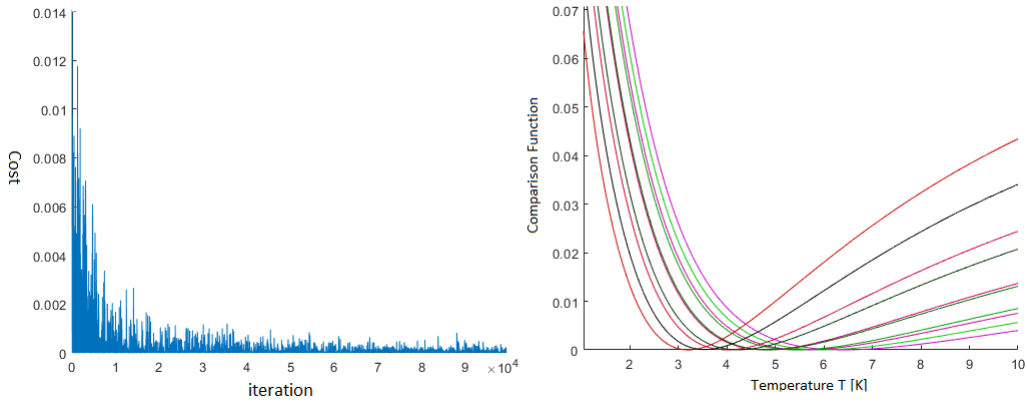
Figure 4.11.: A $[2, 5, 2]$ network trained with 2000 mini-batches, ΔT_{mb} (see Equation 4.10) uniformly distributed on $(0, 1)$, and 6 data pairs and 100 training rounds per mini-batch. Left: recursive cost function (see Equation 4.12) of depth 2 with the Nadam learning rate falling as a degree 3 polynomial (see Equation 4.11). Right: the comparison function 4.2.1 for the post-processed states (black to lime) and of the corresponding label states (red to pink). Figure taken from [179].
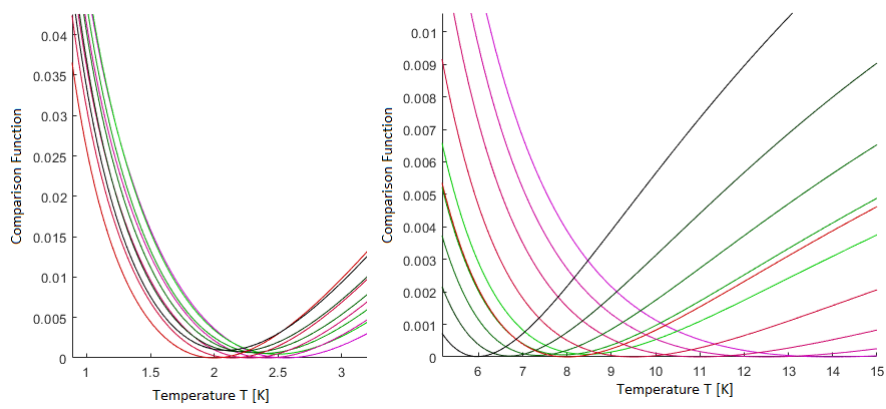
Figure 4.12.: The comparison functions for the output states (black to lime) and the corresponding input states (red to pink) for the QNN trained as depicted in Fig. 4.11. Left: input states are colder than the temperature region the network was trained on. Right: input states are hotter than the temperature region the network was trained on. Figure taken from [179].
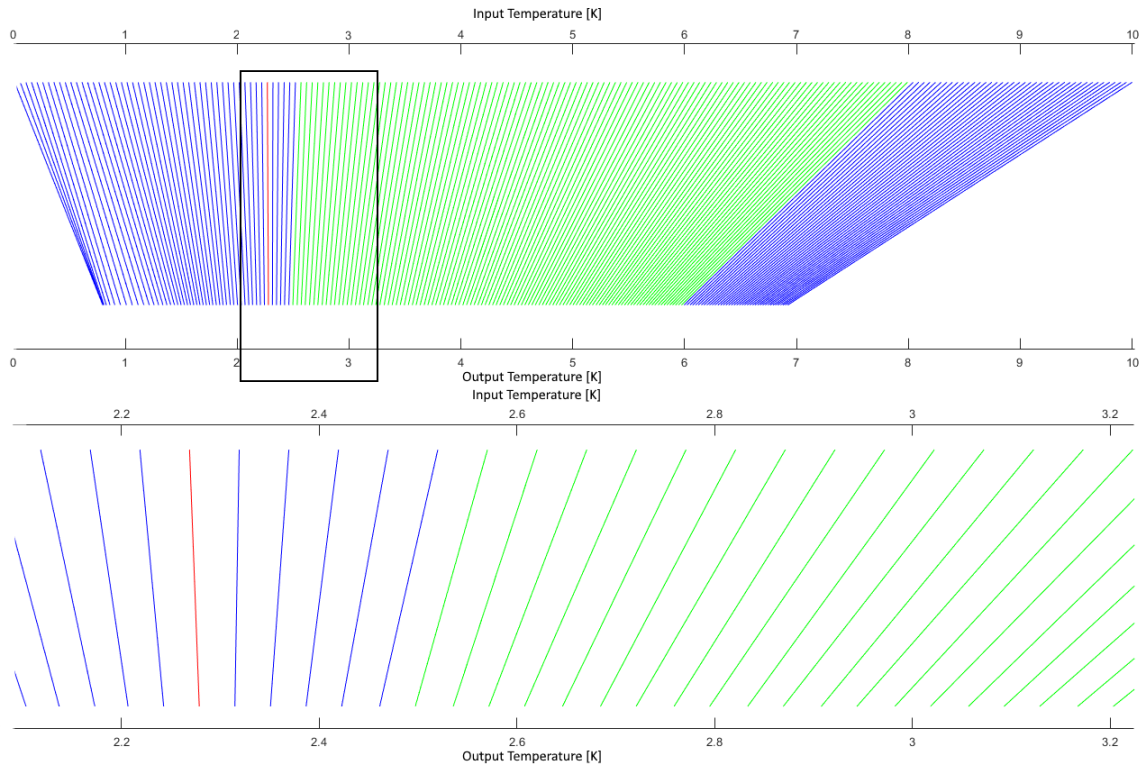
Figure 4.13.: Extrapolation with the $[2, 5, 2]$ network trained as depicted in Fig. 4.11. The lines visualize the effect the network has on the input thermal states with temperatures given by the x-coordinate of the upper end of each line. The assigned temperatures of the corresponding output states equal the x-coordinate of the lower end of each line. The blue connections correspond to input states outside of the training region. The lime connections correspond to input states inside the training region. The red connection corresponds to the state that has the highest temperature out of the states not cooled by the network. The network cools every state that is hotter than the "red" state. Upper plot: an overview of the temperature mapping of the network. Lower plot: the marked area in detail. Figure taken from [179].

4.2.3. Discussion

We have successfully trained a QNN to extrapolate thermal states to lower temperatures. However, there are a number of different cooling quantum algorithms on the market, e.g. [461, 476, 477, 404, 522, 347, 393] (and schemes to access low-temperature properties of a system, e.g. [573, 387, 138, 394]). Why QNNs?

The benefit of QNNs is their flexibility and extendability. Unfortunately, in comparison to [461, 476, 477], our proposal needs vastly greater resources. It is no wonder that minimalist approaches like [476, 477] are the ones that are experimentally implemented!

Our training algorithm produces better results for wider networks and tends to produce oversimplified outputs if the depth is increased (a problem analogous to classical NNs [644, 294]). Can width be efficiently traded for depth (see Subsection 2.6.2 and [572])? We hypothesize that drop out, i.e. random omission of connection and neurons, can help in this. It is also worthwhile to check if a different cost function (see Subsection 2.8.2) or a different function $f(\cdot)$ used for data generation (see Equation 4.5) is better suited for the task. As graph neural networks seem to suffer less from extrapolation problems [644], it might be worthwhile to combine our method with [58].

The picture gets brighter for our approach if we consider that QNNs are universal and are capable of much more than cooling. Its flexibility means that the ML approach to extrapolation presented here trivially generalizes to other parameters, such as interaction strength, conserved charges or amount of squeezing; see also [587, 588, 294] for some other applications of extrapolation for physical systems.

We develop a variety of denoising algorithms in this thesis, and QNNs can deal not only with undesired parameter ranges, but also mitigate shot-to-shot noise, drifts and transmission errors. Extendability means that there is no reason why a single QNN cannot learn to cancel multiple noise sources or to do extra analysis and processing.

4.3. Autoencoders for unsupervised denoising

In this section we train denoisers to combat shot-to-shot errors.

There is often no denoised reference state to compare with, thus unsupervised learning is required. Autoencoder neural networks (AEs) can train unsupervised and are widely used for denoising, see Subsection 2.6.4.

We construct quantum AEs capable of quantum advantage (see Definition 2.5.15) for the purpose of denoising quantum data. We apply them to single and continuously parameterized sets of small highly entangled states subject to different kinds of noise. We observe excellent

denoising without fine tuning of the hyperparameters. The results in this section were first presented in [85, 516].

Classically simulable quantum AEs have been studied in [235]. In [498, 606] shallow quantum AEs have been introduced for data compression. The QNNs in [498, 606] are closely related to the neurons from [56]. Contrary to a claim in [498] they are universal. However, the authors of [498] restrict the class of operations to get polynomial complexity scaling with the width of the network. Data compression via AEs has been demonstrated with photons [446]. In [350], it has been proposed to train AEs for quantum data compression using genetic algorithms on a classical computer. After [85] was published, an algorithm that combines training task from [85] with the network architecture from [498] appeared in [650]. The trained AEs have been implemented on superconducting qubits [168]. Classical ML techniques have been used to design experiments that produce entangled states [391, 421] that are useful and robust to noise [322, 421]. The general setting of quantum unsupervised ML has been studied in [537].
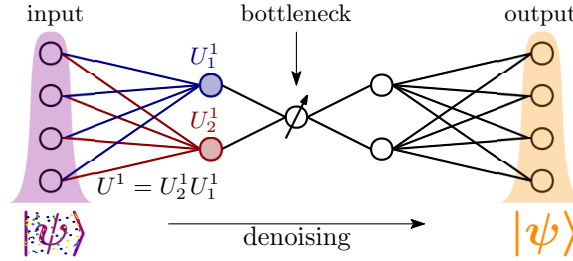

Figure 4.14.: Network architecture of an AE. The bottleneck prevents the AE from just copying the input data (violet) to the output (orange) so that it has to extract relevant features—denoised states. Each neuron unitary acts on its qubit and the connected qubits in the previous layer (e.g. navy or dark red).

We construct undercomplete quantum AEs: the bottleneck is represented by a layer of smaller width than the (equal) input and output layers. Due to the no-cloning (Theorem 2.5.10), it is impossible to use copies of the training inputs $|\psi_i^{\mathrm{in}}\rangle$ as reference outputs $|\psi_i^{\mathrm{ref}}\rangle$. Instead, these states have to be prepared independently. If the data source is noisy, the paired states will be different due to different noise realizations. However, if these states share essential features, the AE can still be trained. Below, we use half of the noisy training data as input and half as reference output in unsupervised learning.

Our distance measure is one minus the fidelity F. For training data $\{\rho_i^{\mathrm{in}}, |\psi_i^{\mathrm{ref}}\rangle\}_{i=1}^L$ with pure desired outputs, $F(\rho, |\psi\rangle) = \langle\psi|\rho|\psi\rangle$ and the cost function reads

$$C\Big(\{\rho_i^{\mathrm{in}}, |\psi_i^{\mathrm{ref}}\rangle\}_{i=1}^L\Big) = 1 - \bar{F}\Big(\{\mathcal{N}(\rho_i^{\mathrm{in}}), |\psi_i^{\mathrm{ref}}\rangle\}_{i=1}^L\Big), \tag{4.14}$$

where $\bar{F}(\{\rho_i, |\psi_i\rangle\}_{i=1}^L) = \frac{1}{L}\sum_{i=1}^L F(\rho_i, |\psi_i\rangle) \leq 1$. In the following, we abbreviate pure $\rho_i^{\mathrm{in}} = |\psi_i^{\mathrm{in}}\rangle\langle\psi_i^{\mathrm{in}}|$ by $|\psi_i^{\mathrm{in}}\rangle$.

While, in practice, one has no access to the desired outputs of the NN $\{|\psi_i^{\mathrm{id}}\rangle\}_{i=1}^L$, the performance of AEs is best studied in a setting where these target states are known. We call the

learning process successful if the mean *validation function*

$$\bar{F}_{\text{val}}\Big(\{\rho_i^{\text{in}}, |\psi_i^{\text{id}}\rangle\}_{i=1}^L\Big) = \bar{F}\Big(\{\mathcal{N}(\rho_i^{\text{in}}), |\psi_i^{\text{id}}\rangle\}_{i=1}^L\Big) \tag{4.15}$$

is large, particularly, as compared to $\bar{F}(\{\rho_i^{\text{in}}, |\psi_i^{\text{id}}\rangle\}_{i=1}^L)$ before the NN is applied. We define

$$
\begin{aligned}
F_{\text{val}}^{(i)}(\{\rho_i^{\text{in}}, |\psi_i^{\text{id}}\rangle\}_{i=1}^L) &= F(\mathcal{N}(\rho_i^{\text{in}}), |\psi_i^{\text{id}}\rangle), \\
F^{(i)}(\{\rho_i^{\text{in}}, |\psi_i^{\text{id}}\rangle\}_{i=1}^L) &= F(\rho_i^{\text{in}}, |\psi_i^{\text{id}}\rangle).
\end{aligned}
\tag{4.16}
$$

Note that the validation function, which compares $\{\mathcal{N}(\rho_i^{\text{in}})\}_{i=1}^L$ with the target states, differs from the fidelity entering the cost function for training, which compares $\{\mathcal{N}(\rho_i^{\text{in}})\}_{i=1}^L$ with the noisy data.

For the classical simulation of the quantum AE we have upgraded the MATLAB code from [56]. Most importantly, we now use the Nadam (see Section 2.6.3 or [172, 505]) gradient descent algorithm. The updated code is available at [2].

A Qiskit implementation of the quantum AEs was developed in [7].

Noisy test states

We consider highly entangled Greenberger-Horne-Zeilinger (GHZ), W, Dicke, and cluster states. Such states are important for quantum information and quantum enhanced metrology [452, 177, 482].

For practical applications the states have to be protected from experimental noise. We investigate two complementary noise processes—spin-flip errors and small random unitary transformations (see e. g. [73, 424, 639])—and show how quantum AEs can be used to denoise small GHZ, W, Dicke, and cluster states.

4.3.1. Denoising a single state

GHZ states

We call

$$|\text{GHZ}_\phi\rangle = \frac{1}{\sqrt{2}}\left(|\uparrow\rangle^{\otimes m} + \text{e}^{i\phi}\,|\downarrow\rangle^{\otimes m}\right) \tag{4.17}$$

an m-qubit GHZ state with phase ϕ or a GHZ-ϕ state. GHZ states are macroscopic superposition states with maximal entanglement depth.

Spin flip noise

For spin-flip errors we assume that for a time T all qubits are flipped back and forth at some rate Γ. Thus each qubit has a probability of $p = (1 - \text{e}^{-2\Gamma T})/2 \le 0.5$ to end up in a flipped state. The flips of the jth qubit affect the density matrix ρ of the initial, noiseless, m-qubit state according to

$$\mathcal{E}_j(p, \rho) = p\sigma_j^x \rho \sigma_j^x + (1-p)\rho. \tag{4.18}$$

The total noise channel is obtained by concatenating $\mathcal{E}_j$ for all qubits $j \in \{1, \ldots, m\}$:

$$\mathcal{E}(p, \rho) = \mathcal{E}_m(p, \mathcal{E}_{m-1}(p, \cdots \mathcal{E}_1(p, \rho) \cdots)) \qquad (4.19)$$

We assume that in each experimental shot a subset $J \subseteq \{1, 2, \ldots, m\}$ of a total of m qubits is flipped. The probability of $\rho_J = \prod_{j \in J} \sigma_j^x \rho \prod_{j \in J} \sigma_j^x$ is $P_p(J) = p^{|J|}(1-p)^{m-|J|}$. Note that states ρ_J with different J may coincide or have non-orthogonal supports.
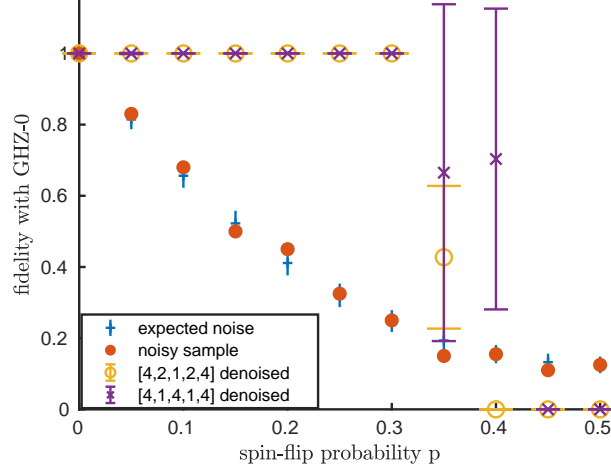

Figure 4.15.: Quantum AEs removing spin-flip errors from the GHZ-0 state. We show the average fidelity of noisy test states with the GHZ-0 state before denoising (red dots, $\bar{F}$) and after denoising (yellow circles / violet crosses, $\bar{F}_{\text{val}}$). Error bars display standard deviations. Blue plus signs show $\bar{F}^\infty \pm \Delta F^\infty$. The arrays $[4, 2, 1, 2, 4]$ and $[4, 1, 4, 1, 4]$ indicate different AE topologies. 200 noisy training pairs, training rounds, and noisy test states per p.

First, we show how well an AE can denoise 4-qubit GHZ states with zero phase. We employ two AE topologies. One is the deep QNN ⤬⤫ denoted by $[4, 2, 1, 2, 4]$ and the other one is a stacked QNN: we train the AE ⤫ $\sim [4, 1, 4]$ but denoise with ⤬⤫ $\sim [4, 1, 4, 1, 4]$ by applying $[4, 1, 4]$ twice. Each training employs 200 training pairs and takes 200 steps of the gradient descent algorithm (200 *training rounds*). We test the trained AEs on 200 GHZ-0 states exposed to the respective noise. The validation function, which, ideally, should reach one, is the fidelity between the denoised output of the AE and the GHZ-0 state.

Fig. 4.15 summarizes our results in the case of spin-flip errors. For each spin-flip probability p we, first, draw the training data and one set of $L = 200$ noisy test states $\{|\psi_i\rangle\}_{i=1}^L$ according to the probability distribution $P_p(J)$. We independently train both AE topologies. For each topology, we apply the respective AE to every $|\psi_i\rangle$ and get outputs ρ_i. To assess the performance of the AE, we evaluate the mean validation function after denoising—$\bar{F}_{\text{val}}(\{|\psi_i\rangle, |\text{GHZ}_0\rangle\}_{i=1}^L)$ $F_{\text{val}}^{(i)}(S) = \langle \text{GHZ}_0 | \rho_i | \text{GHZ}_0 \rangle$ (yellow circles / violet crosses)—and compare it to its value before denoising—$\bar{F}(\{|\psi_i\rangle, |\text{GHZ}_0\rangle\}_{i=1}^L)$ (red dots). We find that up to $p = 0.3$ both AE topologies remove the spin-flip errors almost ideally.

The error bars of $\bar{F}_{\text{val}}$ indicate the standard deviation $\Delta F_{\text{val}} = \sqrt{\sum_i (F_{\text{val}}^{(i)} - \bar{F}_{\text{val}})^2}$. Note that, contrary to $F_{\text{val}}^{(i)}$, $\bar{F}_{\text{val}} + \Delta F_{\text{val}}$ can exceed one. For the input, $\Delta F = \sqrt{\bar{F}(1 - \bar{F})}$ is large since $F^{(i)} \in \{0, 1\}$. Instead of adding error bars to $\bar{F}$, we show how $\{|\psi_i\rangle\}_{i=1}^L$ compares to the ideal probability distribution $P_p(J)$ of spin-flipped GHZ-0 states. The blue plus signs mark the expectation value of F, $\bar{F}^\infty = (1 - p)^4 + p^4$. Their vertical bars indicate the standard deviation $\Delta F^\infty / \sqrt{L} = \sqrt{\bar{F}^\infty (1 - \bar{F}^\infty)/L}$, which characterizes the spread of the average $\bar{F}(\{|\psi_i\rangle, |\text{GHZ}_0\rangle\}_{i=1}^L)$ for independent draws of L noisy states.

Spin-flip errors: limitations of denoising

A simple argument suggests that an AE might denoise GHZ-0 states all the way up to $p = 0.5$. For all $p < 0.5$ the most probable of all spin-flipped GHZ-0 states is the GHZ-0 state itself. All (non-identical) flipped GHZ-0 states are orthogonal to each other. Hence, the state ρ which maximizes the average fidelity with the ideally distributed flipped GHZ-0 states, $\text{argmin}_\rho \sum_J P_p(J) \langle \text{GHZ}_0 | \prod_{j \in J} \mathcal{F}_j \rho \prod_{k \in J} \mathcal{F}_k | \text{GHZ}_0 \rangle$, is the original, noiseless GHZ-0 state.

Why do our AEs fail to denoise GHZ-0 states beyond $p \approx 0.3$? For $p \to 0.5$ all flipped GHZ states become equally probable. On a finite training sample the ordering by probability can get misrepresented. Furthermore, the small difference in cost corresponding to a small difference in probabilities can be missed due to a finite training precision. Finally, an actual AE can learn more than a single state. Note also that, so far, we have not optimized the hyperparameters of the gradient descent for individual noise strengths.

For Fig. 4.19 we, again, train our stacked $[4, 1, 4, 1, 4]$ AE to remove spin-flip errors from the GHZ-0 state. As compared with Fig. 4.14a, we increase the number of training pairs from 200 to 1500 and optimize the gradient descent for a large spin-flip probability p. As a result, denoising succeeds up to $p = 0.4$ instead of $p = 0.3$.

Unitary noise

For unitary errors we assume that any one-qubit error can occur with probability p_u on every qubit. The noise channel for the jth qubit is a depolarizing channel and can be written as

$$
\begin{aligned}
\mathcal{E}_j(p_u, \rho) &= p_u \text{tr}_{m+1} \left[\text{SWAP}_{j,m+1} \left(\rho \otimes \frac{Id}{2} \right) \text{SWAP}_{j,m+1}^\dagger \right] + (1 - p_u)\rho \\
&= \left(1 - \frac{3p_u}{4} \right) \rho + \frac{p_u}{4} \left(\sigma_j^x \rho \sigma_j^x + \sigma_j^y \rho \sigma_j^y + \sigma_j^z \rho \sigma_j^z \right)
\end{aligned}
\tag{4.20}
$$

The total noise channel is obtained by concatenating $\mathcal{E}_j$ for all qubits. Here $\text{tr}_j(\cdot)$ is a partial trace and $\text{SWAP}_{j,m}$ swaps the jth with the mth qubit. A random error can be attributed to the evolution with a random time-dependent Hamiltonian $H(t)$, where the error probability p_u is a monotonic function of the interaction strength and the evolution time T. The evolution with $H(t)$ can be constructed just like in the Subsection 4.1.1. We consider a family of Hamiltonians $\{H_j = H(j\Delta t)\}_{j=1}^n$, $\Delta t = T/n$, such that the entries of every Hermitian H_j are Gaussian distributed with zero mean and a standard deviation of $\pi \hbar \nu \sqrt{\frac{n}{2^m}}$. The dimensionless parameter

$q = \nu T$ captures the noise strength. We assume that in each experimental shot the initial state evolves with the unitary operator

$$U = \prod_{j=1}^{n} \exp(iH_j \Delta t/\hbar). \tag{4.21}$$

We use $n = 20$.

Just as for spin-flip errors, we employ the $[4, 2, 1, 2, 4]$ and $[4, 1, 4, 1, 4]$ quantum autoencoders (AEs) to denoise 4-qubit GHZ states with zero phase. Each training involves 200 training pairs and takes 200 training rounds. We test the trained AEs on 200 noisy GHZ-0 states.

Random unitary noise gives Fig. 4.16a. To get a train or test state, we evolve the GHZ state with a random unitary drawn according to the respective noise strength q. We, again, compare the outcomes of the validation function before and after denoising. This time, we add error bars of size $\Delta F_{(\text{val})} = \sqrt{\sum_i (F_{(\text{val})}^{(i)} - \bar{F}_{(\text{val})})^2}$ to both $\bar{F}$ and $\bar{F}_{\text{val}}$. Virtually perfect denoising succeeds up to a noise strength of $q = 0.375$.

Next, we combine the two noise models. Spin-flip errors with $p = 0.2$ are followed by random unitary transformations with $q = 0.3$. We train and test the $[4, 2, 1, 2, 4]$ AE on the combined noise. Fig. 4.16b shows that the AE impressively increases the fidelity of each noisy test state with the GHZ state.
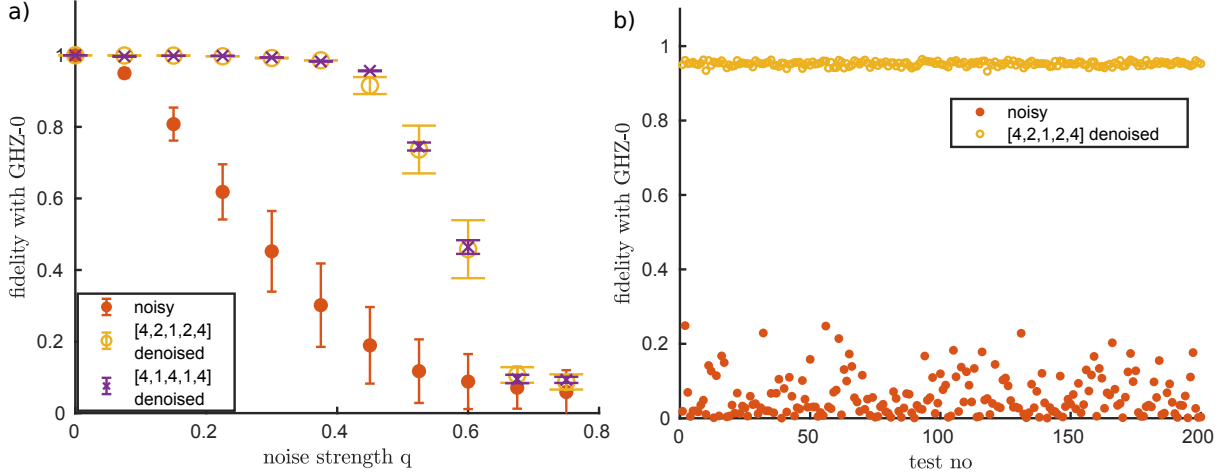
Figure 4.16.: Quantum AEs denoising GHZ-0 states. The arrays $[4, 2, 1, 2, 4]$ and $[4, 1, 4, 1, 4]$ indicate different AE topologies. 200 noisy training pairs, training rounds, and noisy test states per q. (a) Random unitary noise. We show the average fidelity of noisy test states with the GHZ-0 state before denoising (red dots, $\bar{F}$) and after denoising (yellow circles / violet crosses, $\bar{F}_{\text{val}}$). Error bars display standard deviations. (b) Combined noise: spin-flip errors with $p = 0.2$ followed by random unitary transformations with $q = 0.3$. We show the fidelity of each noisy test state with the GHZ-0 state before denoising (red dots, $F^{(i)}$) and after denoising (yellow circles, $F_{\text{val}}^{(i)}$). The respective average fidelities and standard deviations are $\bar{F} = 0.058$, $\Delta F = 0.056$, and $\bar{F}_{\text{val}} = 0.953$, $\Delta F_{\text{val}} = 0.006$.

Dephasing

For a single qubit dephasing introduces a random relative phase between the states $|\uparrow\rangle$ and $|\downarrow\rangle$ [424, 468]. This is described by the channel

$$\mathcal{E}(p_-, \rho) = p_- \sigma^z \rho \sigma^z + (1 - p_-)\rho, \tag{4.22}$$

where p_- is the probability of a phase jump of π. In a system with $m > 1$ qubits we can distinguish two limiting cases. First, all qubits can dephase independently, which corresponds to a concatenation of $\mathcal{E}$ acting on each individual qubit. Second, the qubits can dephase collectively, such that σ^z in Eq. (4.22) has to be replaced by $S_z = \bigotimes_{j=1}^{m} \sigma_j^z$.

For the first case we assume that in each experimental shot the noiseless target state gets evolved with $U = \prod_{j=1}^{m} e^{i\phi_j \sigma_j^z}$, where ϕ_j are independently and identically normally distributed around zero. This is a special case of the unitary noise discussed in section 4.3.1, in the sense that it is (up to a global phase) equivalent to restricting the random Hamiltonians in Eq. (4.21) to diagonal operators. Similarly, in the second case we set $U = e^{i\phi S_z}$ with ϕ normally distributed around zero, which corresponds to a further restriction of the Hamiltonian operators in Eq. (4.21). However, since GHZ states are particularly sensitive to dephasing, we discuss it here.

In any case a GHZ-ϕ_0 state evolves into GHZ-ϕ states with a phase ϕ which is normally distributed around ϕ_0 with some standard deviation of σ. We employ our AEs to denoise 4-qubit GHZ-0 states subject to dephasing in the same way as for removing spin-flip errors and unitary noise (see Fig. 4.15 and Fig. 4.16a). Fig. 4.17 shows excellent denoising up to $\sigma \approx \pi/2$.
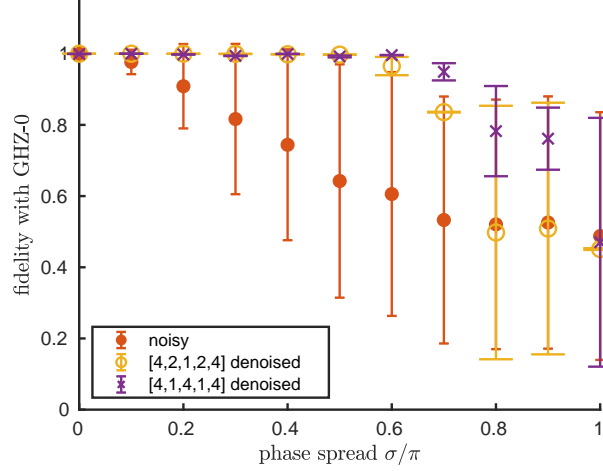
Figure 4.17.: Quantum AEs denoising GHZ-0 states after dephasing. We show the average fidelity of noisy test states with the GHZ-0 state before denoising (red dots) and after denoising (yellow circles / violet crosses). Error bars display standard deviations. The arrays $[4, 2, 1, 2, 4]$ and $[4, 1, 4, 1, 4]$ indicate different AE topologies. 200 noisy training pairs, 200 training rounds, and 500 noisy test states per σ.

W, Dicke, and cluster states

To prove that our AEs are not limited to GHZ states, we train them to denoise highly entangled W, Dicke, and cluster states. Each pure 3-qubit state with genuine 3-partite entanglement belongs to one of two entanglement classes [177]. One class is represented by the GHZ state, and the other one by the W state,

$$|\text{W}\rangle = \frac{1}{\sqrt{3}} \left(|\uparrow\downarrow\downarrow\rangle + |\downarrow\uparrow\downarrow\rangle + |\downarrow\downarrow\uparrow\rangle \right). \tag{4.23}$$

Dicke states are eigenstates of the magnitude $\mathbf{J}^2$ and z-component J_z of a collective angular momentum $\mathbf{J} = (J_x, J_y, J_z)$, which have been originally introduced for modeling spontaneous radiation [165]. For an m-qubit state $J_i = \frac{1}{2} \sum_{j=1}^m \sigma_j^i$. We focus on m-qubit Dicke states with a maximal magnitude, $|k, m - k\rangle$ with $\mathbf{J}^2 |k, m - k\rangle = \frac{m}{2}(\frac{m}{2} + 1)|k, m - k\rangle$ and $J_z|k, m - k\rangle = (k - \frac{m}{2})|k, m - k\rangle$. These states are called symmetric Dicke state because they are bosonic 2-mode Fock states,

$$|k, m - k\rangle = \frac{1}{\sqrt{m! k! (m - k)!}} \sum_{\pi \in S_m} \bigotimes_{j=1}^m |\updownarrow_{\pi(j)}\rangle, \tag{4.24}$$

where S_m is the symmetric group, and $|\updownarrow_j\rangle$ denotes $|\uparrow\rangle$ for $j \in \{1, \ldots, k\}$ and $|\downarrow\rangle$ for $j \in \{k+1, \ldots, m\}$. Note that $|1, 2\rangle = |W\rangle$. Symmetric Dicke states have been proposed as probe states for quantum enhanced metrology (see e. g. [452, 579]) and for experimentally studying multi-partite entanglement [578]. Twin Fock states $|k, k\rangle$ are particularly promising for these applications and have already been prepared in various experiments (see e. g. [373, 374]). Therefore our second example is the Dicke or twin Fock state $|2, 2\rangle$.

Cluster states [103] are an instance of graph states. A graph consists of vertices and links. Here, the links are undirected, simple (two vertices can be linked only once), and without loops (from a vertex to itself). Such a graph with m vertices can be described by an adjacency matrix $\Gamma \in \{0, 1\}^{m \times m}$, where $\Gamma_{ij} = 1$ if the vertices i and j are linked, and $\Gamma_{ij} = 0$ otherwise. The corresponding graph state consists of one qubit per vertex and is defined as

$$|G_\Gamma\rangle = \frac{1}{2^{m/2}} \sum_{b \in \{0,1\}^m} (-1)^{\frac{1}{2}b^T \Gamma b} |b\rangle, \qquad (4.25)$$

where $|0\rangle \equiv |\uparrow\rangle$, and $|1\rangle \equiv |\downarrow\rangle$ [240]. The graph of a cluster state belongs to a d-dimensional lattice. Graph states can be used for quantum error correction [240, 482, 483, 262, 520]. Cluster states have, in general, a high persistency of entanglement—disentangling them requires a large number of local measurements [103]. Furthermore, cluster states can serve as a universal resource for measurement-based quantum computation [481, 482]. We consider the square state $|G_{sq}\rangle$ whose underlying graph is a square with the adjacency matrix

$$\Gamma_{sq} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}. \qquad (4.26)$$

The quantum error-correcting code based on the square graph has a code distance of two [240].

Fig. 4.18 summarizes our results for the W state, the Dicke state $|2, 2\rangle$, and the square state. For each of the states we separately study both spin-flip errors and unitary noise. We employ a $[3, 2, 1, 2, 3]$ and a stacked $[3, 1, 3, 1, 3]$ AE topology for the 3-qubit W state, and a $[4, 2, 1, 2, 4]$ topology for the 4-qubit Dicke and square states. For the latter target states the stacked $[4, 1, 4, 1, 4]$ AE, which successfully denoises GHZ states, has turned out to be unsuitable. Like for the GHZ-0 state, we use 200 training pairs and up to 200 training rounds for each spin-flip probability p and unitary noise strength q. All target states get denoised well in a wide range of p and q. The denoising performance is similar for the different target states, including the GHZ-0 state (see Fig. 4.15 and Fig. 4.16a).

4.3.2. Denoising multiple states

So far we have demonstrated that an AE can denoise the state on which it has been trained. But it can do better. An AE can learn to denoise multiple target states, including ones not contained in the training data. It is crucial, though, that the noise process is sufficiently different from

the transformations connecting the target states. Otherwise, the attribution of noisy states to target states becomes ambiguous. Assume that an experiment encodes some information into the phase of a GHZ state, and that this GHZ state is affected by spin-flip errors. We show that an AE can denoise the output of such an experiment. We consider 3-qubit states and employ the simplest possible AE topology: ⧳⧲ ∼ [3, 1, 3].

As a first example, we imagine an experiment that outputs either a GHZ state with zero phase, GHZ-0, or with phase π, GHZ-π. The AE is trained for 200 rounds on 100 pairs of noisy GHZ-0 states and 100 pairs of noisy GHZ-π states. To test the performance of the trained AE, we apply it to 100 noisy GHZ-0 states and 100 noisy GHZ-π states and compare each output to the respective noiseless state. Fig. 4.20a shows that the AE excellently denoises the two GHZ states up to a spin-flip probability of $p = 0.4$. Note that the AE deduces whether the experiment has given a phase of zero or π from the particular noisy input state alone.

Our second example is even more demanding. We assume that the experiment can output a GHZ state with any phase $\phi \in [0, \pi]$. We restrict the phase to $[0, \pi]$ because it is impossible to distinguish a GHZ-ϕ state with $|J|$ flipped qubits from a GHZ-$(-\phi)$ state with $3 - |J|$ flipped qubits. The training involves only four equidistant training phases ϕ_i between $\phi_0 = 0$ and $\phi_4 = \pi$. It, again, employs 100 training pairs per ϕ_i and takes 200 training rounds. We test the AE on 200 noisy GHZ-ϕ states with randomly chosen phases $\phi \in (0, \pi)$.

Considering a GHZ-ϕ state with $\phi \notin \pi\mathbb{Z}$ roughly doubles the number of different spin-flipped states as compared to $\phi \in \pi\mathbb{Z}$. Only for $\phi \in \pi\mathbb{Z}$, the flipped m-qubit states $\prod_{j \in J} \sigma_j^x |\text{GHZ}_\phi\rangle$ and $\prod_{j \in M \setminus J} \sigma_j^x |\text{GHZ}_\phi\rangle$ with $J \subseteq M = \{0, 1, \ldots, m\}$ are, up to a global phase, identical. As a consequence, for 3-qubit GHZ-ϕ states with $\phi \in \pi\mathbb{Z}$, correcting spin-flip errors with $|J| = 1$ suffices for perfect denoising. For $\phi \notin \pi\mathbb{Z}$, errors with $|J| = 2$ and $|J| = 3$ need to be regarded separately.

Fig. 4.20b displays the capability of the AE to denoise GHZ states with a random phase. Note that for $p = 0$ the fidelity of the outputs with the test states reaches one. Because of the bottleneck, the AE cannot learn the identity operation; nevertheless, it correctly reproduces GHZ states with phases not contained in the training data. The AE improves the average value of the validation function for $p \leq 0.35$, but it leaves a considerable variance (yellow circles). However, if we keep only the test states with $|J| \leq 1$, we observe excellent denoising up to $p = 0.2$ (violet crosses).
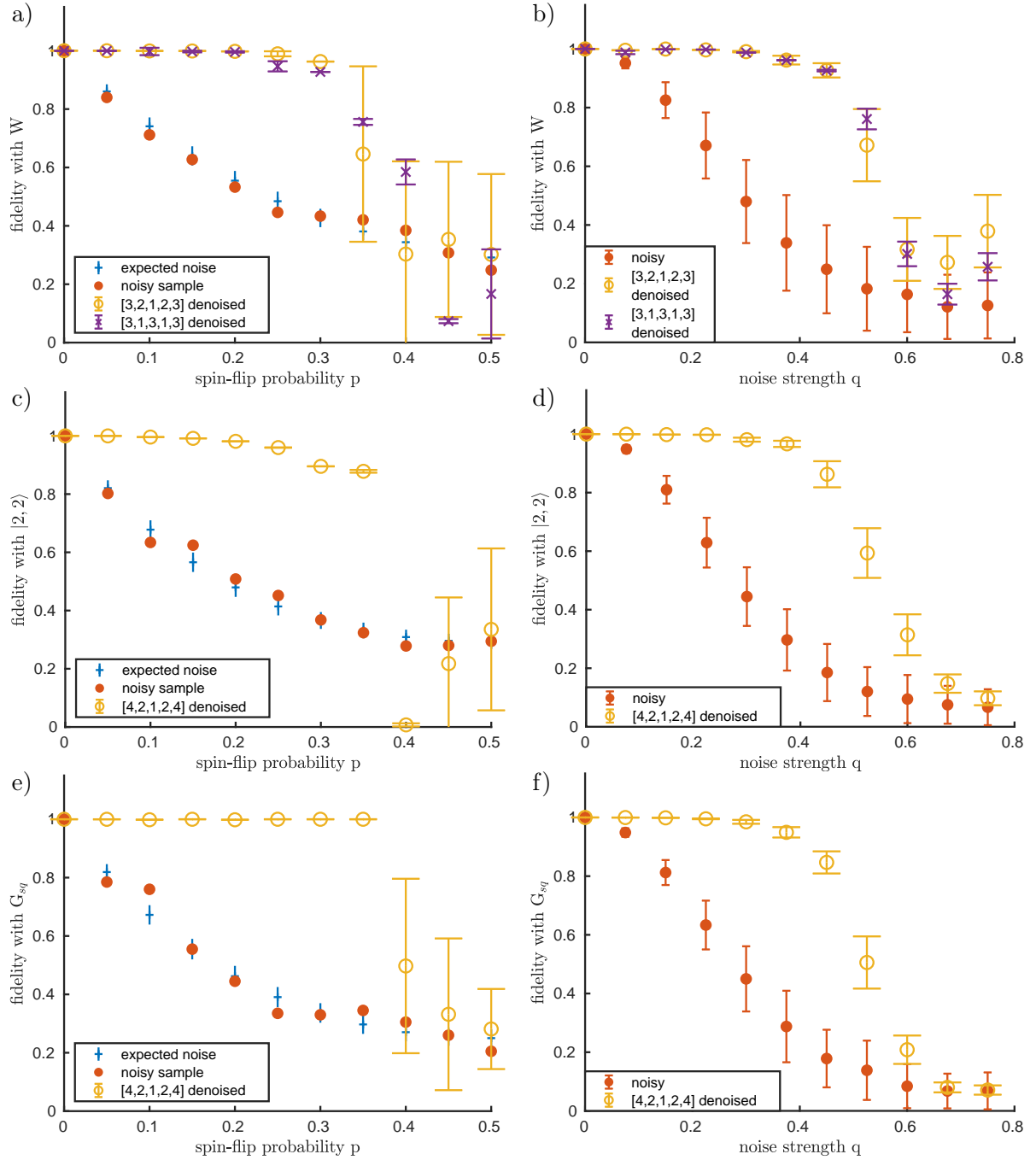
Figure 4.18.: Quantum AEs denoising various states: the W state (a, b), the Dicke state $|2, 2\rangle$ (c, d), and the square state $|G_{sq}\rangle$ (e, f). We show the average fidelity of noisy test states with the target states before denoising (red dots, $\bar{F}$) and after denoising (yellow circles / violet crosses, $\bar{F}_{\text{val}}$). Error bars display standard deviations. (a, c, e) Spin-flip errors. Blue plus signs show $\bar{F}^{\infty} \pm \Delta F^{\infty}$. (b, d, f) Random unitary noise. The arrays $[3, 2, 1, 2, 3]$, $[3, 1, 3, 1, 3]$, $[4, 2, 1, 2, 4]$, and $[4, 1, 4, 1, 4]$ indicate different AE topologies. 200 noisy training pairs, 200 noisy test states, and up to 200 training rounds per p and q.
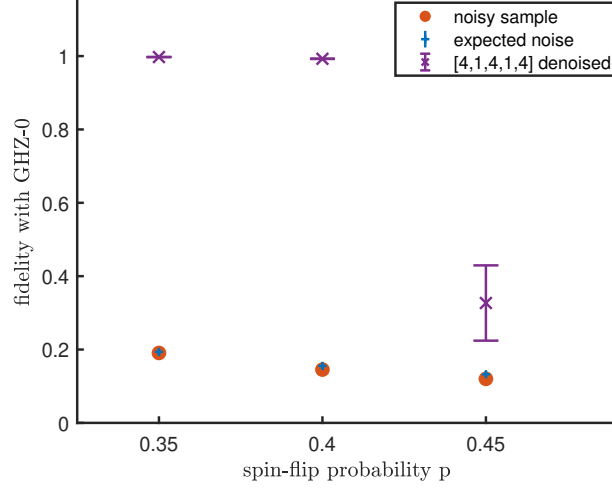
Figure 4.19.: A stacked $[4, 1, 4, 1, 4]$ quantum AE removing spin-flip errors from the GHZ-0 state. 1500 noisy training pairs, 1500 noisy test states, and 75 training rounds per p. We show the average fidelity of noisy test states with the GHZ-0 state before denoising (red dots, $\bar{F}$) and after denoising (violet crosses, $\bar{F}_{\mathrm{val}}$). Error bars display standard deviations. Blue plus signs show $\bar{F}^\infty \pm \Delta F^\infty$.
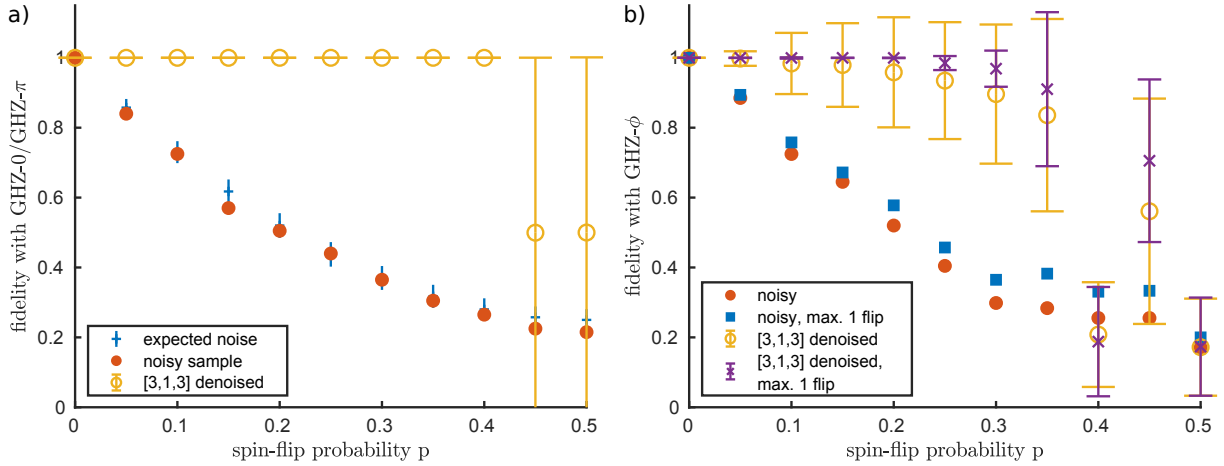
Figure 4.20.: $[3, 1, 3]$ quantum AEs correcting spin-flip errors in mixtures of GHZ-ϕ states with different phases ϕ. We show the average fidelity of noisy test states with the respective noiseless GHZ-ϕ states before denoising (red dots) and after denoising (yellow circles). Error bars display standard deviations. For each p: 100 training pairs per training phase, 200 training rounds, and 200 test pairs. (a) Fifty-fifty mixture of GHZ-0 and GHZ-π states, both for training and testing. Blue plus signs show $\bar{F}^\infty \pm \Delta F^\infty$. (b) Training phases $\{0, \pi/3, 2\pi/3, \pi\}$, and testing on random phases $\phi \in (0, \pi)$. Blue squares before denoising and violet crosses after denoising are obtained for the test states with $|J| \leq 1$.

4.3.3. Sparse quantum autoencoders
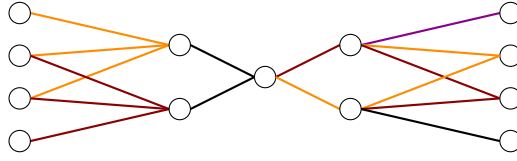
Figure 4.21.: A sparse $[4, 2, 1, 2, 4]$ quantum AE. In each layer, different colors highlight different neurons.
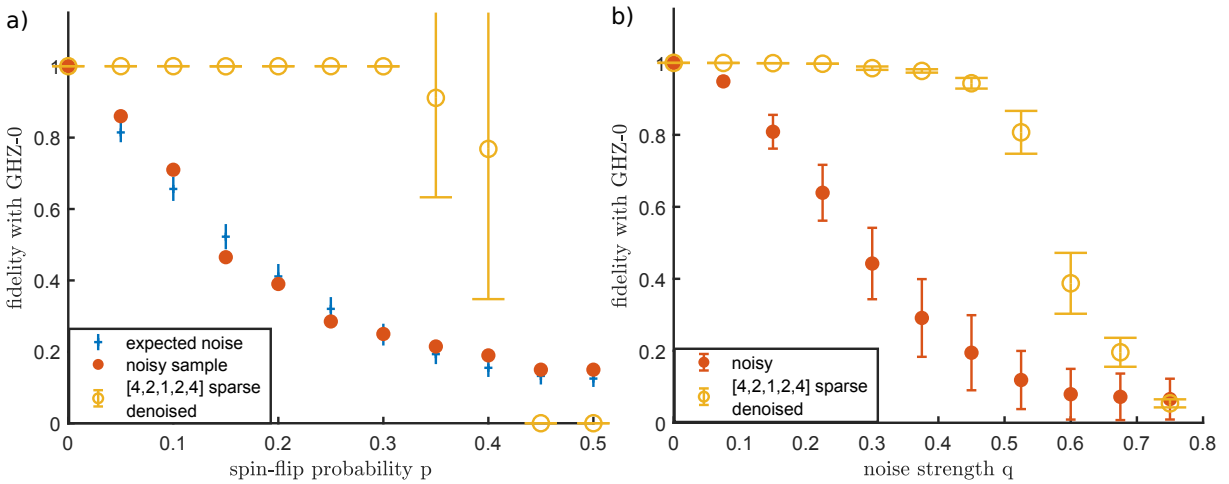
Figure 4.22.: Sparse $[4, 2, 1, 2, 4]$ quantum AE (see Fig. 4.21) denoising GHZ-0 states. We show the average fidelity of noisy test states with the GHZ-0 state before denoising (red dots, $\bar{F}$) and after denoising (yellow circles, $\bar{F}_{\mathrm{val}}$). Error bars display standard deviations. 200 noisy training pairs, training rounds, and noisy test states per p and q. (a) Correcting spin-flip errors. Blue plus signs show $\bar{F}^\infty \pm \Delta F^\infty$. (b) Correcting for random unitary noise.

In general, a neuron does not have to be connected to all the neurons in the preceding layer. An NN containing neurons with fewer connections is called sparse. Sparse networks depend on less variational parameters. On one hand, this reduces the variational class. Eventually, such a QNN may become classically simulable (see e. g. [235]). On the other hand, this speeds up both training and application, and makes the network less prone to overfitting. Recall that the number of gates needed for one application of a fully connected network scales exponentially with its width. If the number of connections per neuron is kept constant, this scaling becomes linear.

We observe that full connectivity is not essential for the success of our quantum AE. Fig. 4.22 shows the denoising capability of the sparse $[4, 2, 1, 2, 4]$ AE depicted in Fig. 4.21. The results turn out to be compatible with the corresponding fully connected network, see Fig. 4.15. To the

advantage of experiments, our sparse topology is local—the retained connections are immediately adjacent.

4.3.4. Noisy networks

To show that quantum AEs can be successfully used for denoising even if implemented using non-perfect neurons, we subject the neurons to approximate depolarizing noise of strength t as described in Subsection 4.1.1. We train the network $\rightthreetimes\!\!\!\!\bowtie\!\!\!\!\lefttwotimes \sim [3, 1, 3]$ to denoise GHZ-0 state subject to spin-flip (see Fig. 4.23) and unitary (see Fig. 4.24) errors.

We observe that quantum AE are efficient at denoising if the intrinsic noise of constituent neurons is small in comparison to the noise present in the data. We also observe that if the data is very noisy (see e.g. $q = 0.5$ on Fig. 4.24 and $p \geq 0.3$ on Fig. 4.23), the small intrinsic noise does not lead to deterioration of denoising.

Deeper networks have higher intrinsic noise but often are more capable if implemented on noiseless devices. This entails a practical limit for the depth of networks implemented on NISQ devices (see Subsection 2.5.6).

We conclude that unsupervised denoising via quantum AE is an efficient method suitable for currently available quantum computers. It especially fits data sets with significant, in comparison to device AEs are implemented on, amount of noise.

Our results are confirmed by the analysis using Qiskit and more realistic noise models for neurons found in [7].

The results in this subsection were produced as a part of bachelor's thesis [516] co-supervised by the author, Ramona Wolf and Tobias Osborne with valuable input from Polina Feldmann.
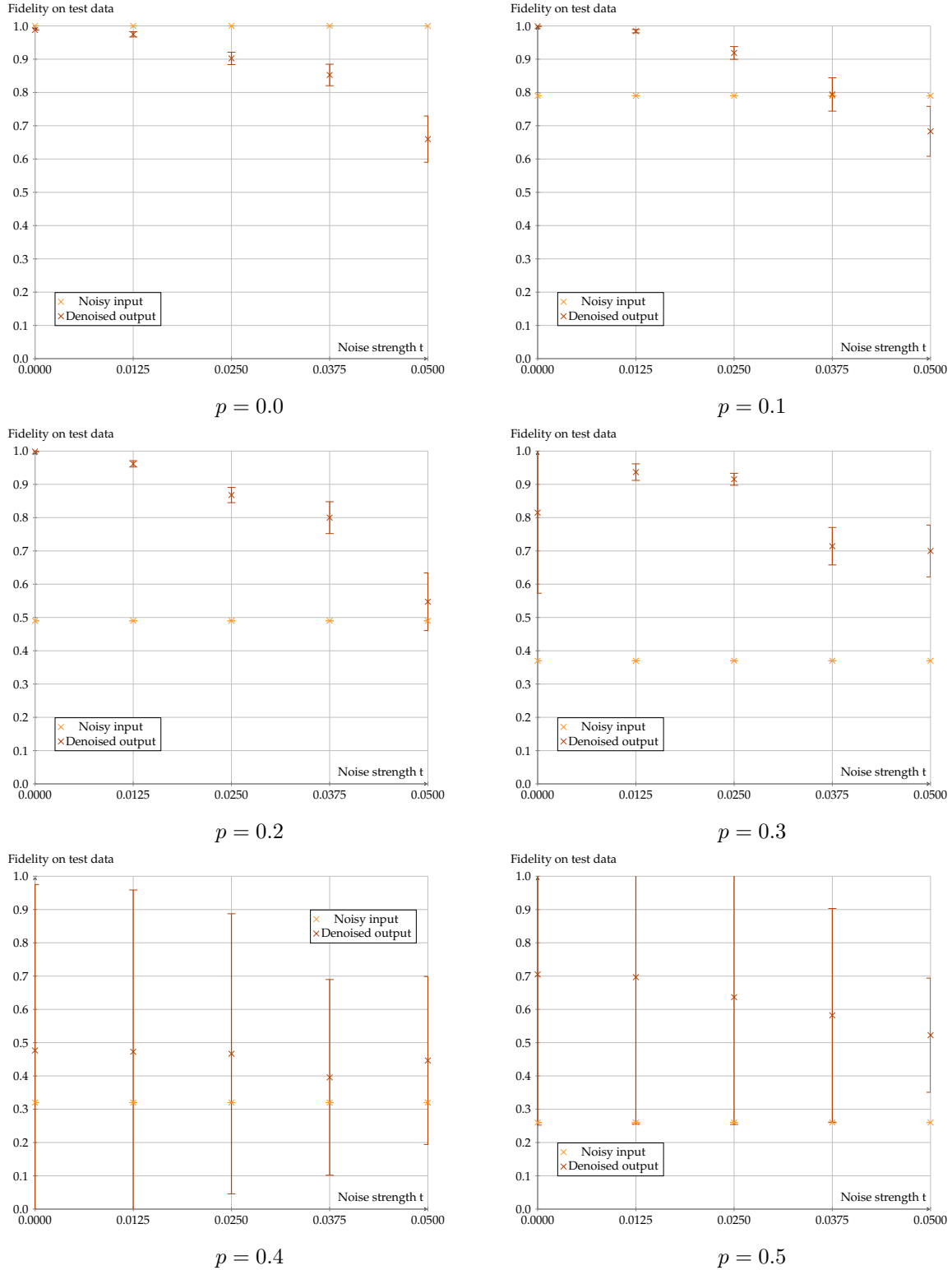
Figure 4.23.: Quantum $[3, 1, 3]$ AEs consisting of noisy qubits removing spin-flip errors. Inputs are GHZ-0 states subjected to spin flips with probability of p per qubit. Neurons and update matrices are evolved with a random time-dependant Hamiltonian with noise strength t like in Subsection 4.1.1. Error bars for denoised outputs display standard deviations. 200 noisy training pairs, training rounds, and noisy test states per p and t. Figure taken from [516].

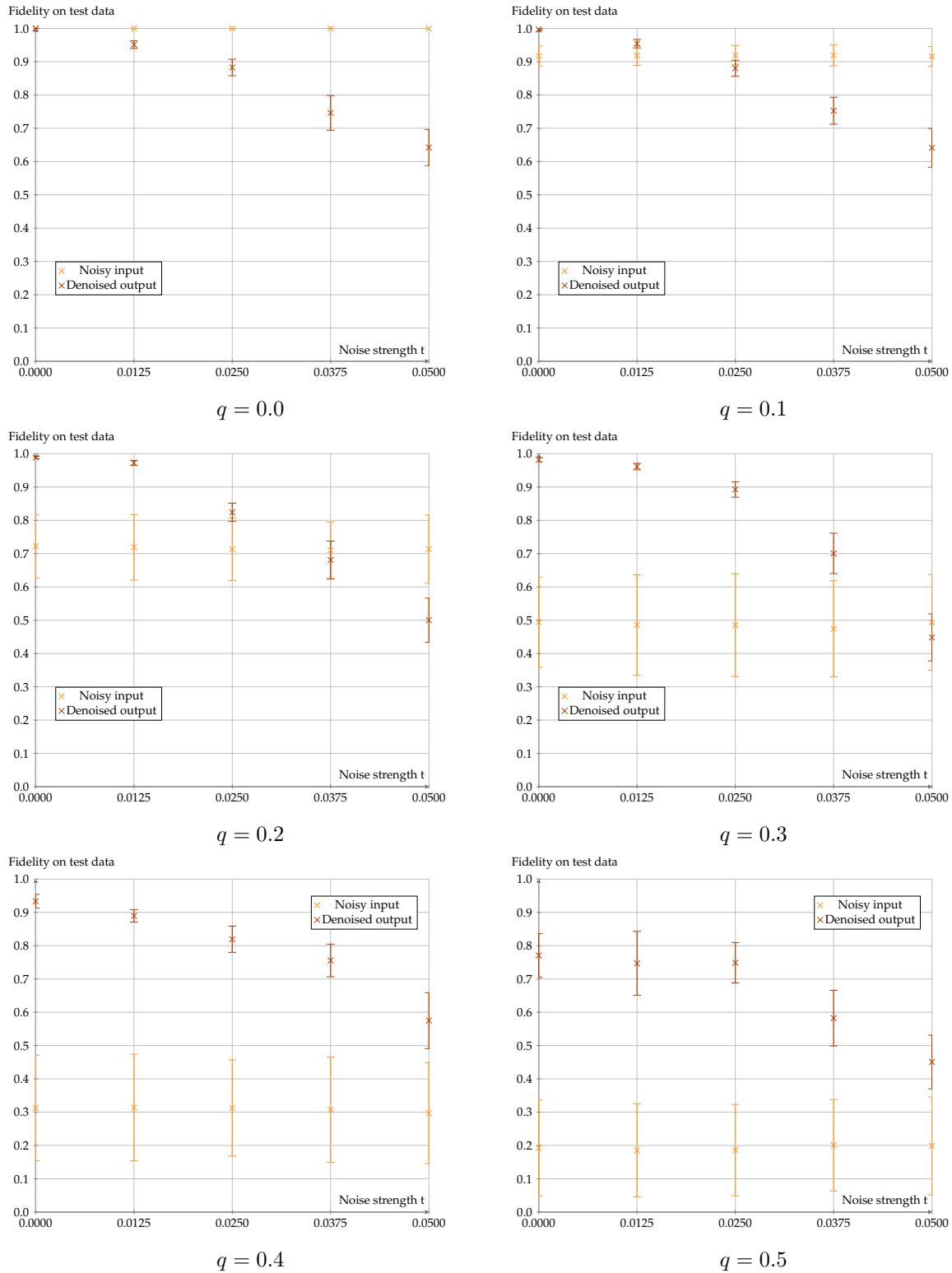Figure 4.24.: Quantum $[3, 1, 3]$ AEs consisting of noisy qubits removing unitary errors. Inputs are GHZ-0 evolved with random time-dependant Hamiltonian with noise strength q, neurons and update matrices—with noise strength t like in Subsection 4.1.1. Error bars display standard deviations. 200 noisy training pairs, training rounds, and noisy test states per q and t. Figure taken from [516].

4.3.5. Discussion

We have constructed quantum AEs and have shown that these AEs can remove spin-flip errors and random unitary noise from small GHZ, W, Dicke, and cluster states. Particularly, correcting spin-flip errors has succeeded for a set of GHZ states parameterized by a continuous phase parameter. Thus, AEs for denoising can be used not only for state preparation but also for metrology. In principle, our method can be applied to any set of quantum states subject to any kind of noise. Further possible applications of quantum AEs include data compression, quantum error correction (see discussion at the end of Section 4.4), enhancing quantum repeaters and memory (see Section 2.7) and parameterized state preparation.

We expect that larger input states will require deeper networks. The number of quantum gates needed for one application of the fully connected AE scales exponentially with the width but only linearly with the depth of the network. The exponential scaling can be avoided by constraining sparse networks.

Small universal quantum computers (see Section 2.5) have been realized on several physical platforms, e. g. superconducting qubits and trapped ions [316, 104]. If the state to be denoised is prepared on the same platform as the AE, both may be affected by equal noise, and the AE may become too noisy for denoising. However, there is a great interest in hybrid systems, see Subsection 2.7.4. Our proposal can help to denoise states from a noisy platform using a well-controlled one, or to remove deteriorating effects introduced at the interface between the coupled platforms. Alternatively, an AEs can be used to improve the output of a subroutine that takes much longer to execute than the AE. Conveniently, quantum AEs are efficient at denoising if the noise affecting the AE itself is small in comparison to the noise present in the data.

Training an AE requires much more computational resources than testing it. To approach the experimental implementation, a small AE trained on a classical computer can be tested on a quantum computer, as has been done in [168] for data compression. Moreover, the photonic realization [446] of a compressing quantum AE suggests that also the training of our AE is within reach of current quantum technology.

4.4. QNNs for error correction

The standard method to mitigate noise is to redundantly encode information into a bigger space, see Subsection 2.5.5. However, different devices and environments may lead to significantly different optimal encoding and decoding scenarios. Machine learning (ML) can aid in developing adaptable quantum error correction (QEC) codes that can be tailored for the circumstances of a particular installation. Not only can such codes be optimized for errors occurring in real, rather than idealized, devices, but we can also expect an ML algorithm to balance out an approximate correction of many errors instead of perfect correction of few [72].

The results in this section were obtained in collaboration with Robert Salzmann with valuable input from Terry Farrelly and Ramona Wolf.

Let us consider two parties, Sender and Receiver, that are separated by a noisy channel. They

can train an error correcting QNN by first agreeing about a sequence of states $\{|\psi_i\rangle\}$. These states can be chosen as random and uniformly distributed. Sender possesses the encoder, and the Receiver—the decoder. They together train the network consisting of the encoder and decoder separated by the noisy channel on the training data $\{|\psi_i\rangle, |\psi_i\rangle\}$.

This protocol is connected to unsupervised denoising via quantum autoencoders (see Section 4.3) but is, in a sense, an opposite to it.

We have trained a ⫷⚬~ $[1, 3]$ encoder and a ⫸⚬~ $[3, 1]$ decoder. The noise is a spin flip with probability p on one random of the qubit. The network has successfully trained to fidelity 1, see Fig. 4.25. Even though this approach to QEC can produce functional codes, there are,
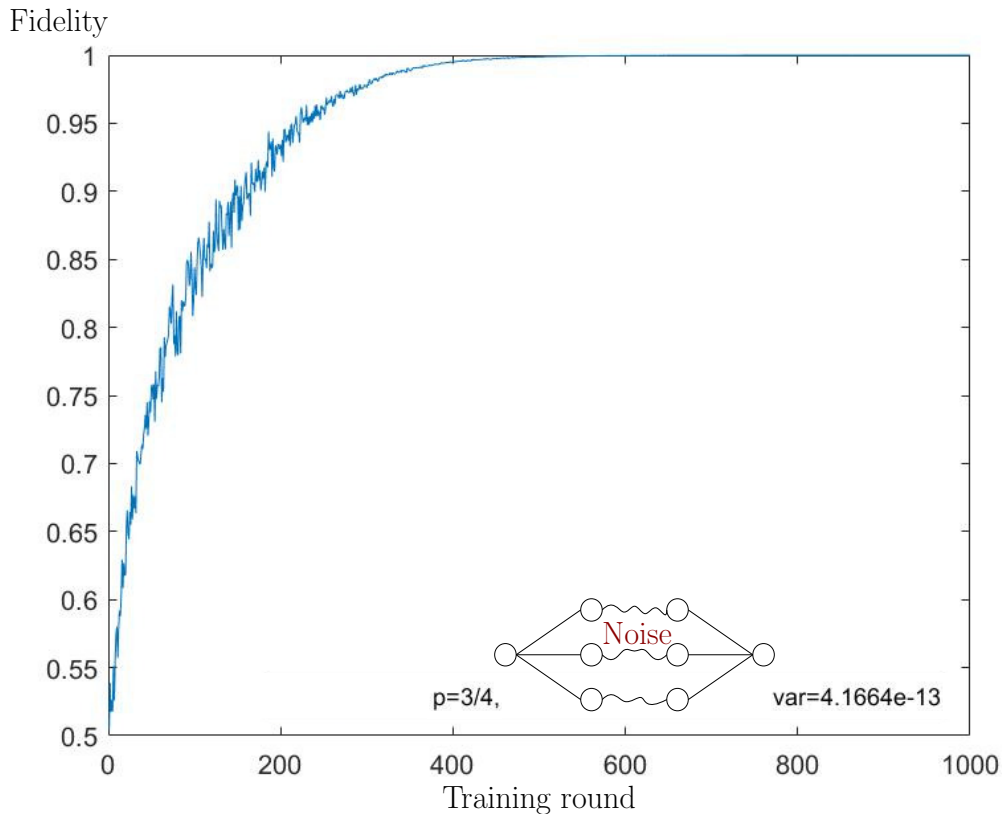
Figure 4.25.: A network consisting of $[1, 3]$ encoder and $[3, 1]$ decoder trains to counter one spin flip at a random location. Spin flips occur with probability $p = 3/4$. The fidelity reaches one, with variance for the last 50 training rounds of $4.1664 \cdot 10^{-13}$.

unfortunately, several problems in this approach.

A logical niche for quantum machine learning (QML) are high-dimensional QECC. Any device-specific QEC code should outperform general purpose codes and the code the device was originally designed around. Such codes can be pretty robust and hard to improve upon. Nevertheless, large quantum devices can often have complicated noise sources (see e.g. [613, 284, 561, 383]) that were not accounted for during their design, making them a conducive target for ML. Classical optimisation and ML algorithms are used to find QEC protocols, especially decoders. As classical

computers can not simulate large quantum devices efficiently, QML is a logical choice for the design of high-dimensional QECC. However, the success in learning a three-qubit and other small codes is by no means a proof that our approach will be successful in producing codes that are too big to be accessed classically.

A more conceptual problem is that a simple approach outlined in this section is not guaranteed to produce fault-tolerant codes. We can expect QNN to avoid noisy gates, but errors can spread through the network. The error propagation can be to some extent mitigated by reducing connectivity of the network, yet this has to be balanced out with the expressive power of the network.

A promising way to circumvent these problems is to use QNNs in subroutines for already existing QECC. In particular, Markus Müller and Lorenzo Cardarelli have recently suggested employing quantum autoencoders as a stage in QEC protocols.

4.5. Conclusions and outlook

In this chapter we have successfully shown that QNNs can successfully learn to denoise quantum sources without additional supervision. This can be achieved through either the design of data sets obtained from a tunable source or AE architectures. Learning can occur if some of the data is corrupted and the QNN is implemented on a noisy device. As such, QNNs can be trained to improve state preparation protocols.

In this chapter we have considered noise sources that are uncorrelated in time. Such noise sources include the preparation of states with non-optimal parameters, shot-to-shot noise and transmission errors.

We provide a proof-of-principle demonstration for ML systems that are able to extrapolate quantum states to lower temperatures and learn QECC. For quantum extrapolation, it would be nice to understand how to trade width for the depth of the network. It is also worthwhile to explore extrapolation beyond the case of temperature considered in this thesis. For learning QECC, it is important to develop methods that would ensure fault-tolerance of the obtained protocol.

We show that quantum AEs can denoise highly entangled quantum states from generic shot-to-shot errors. The protocol works for significant noise strength, for denoising of multiple states parameterized by e.g. a phase and if the AE is implemented on non-perfect hardware. As such, it is a compelling tool for practical tasks spanning data compression, optimization of quantum circuits, noise mitigation at interfaces, design of quantum repeaters without memory and information retrieval from quantum memory. However, in order to use quantum autoencoders in practice, it is important to understand how to implement QNNs and facilitate their learning with as little hardware resources as possible.

Each approach discussed in this chapter can benefit from a more in-detail understanding of the theoretical limitations of learning and denoising protocols. Finally, it would be nice to combine the building blocks presented in this chapter into one more powerful system.

5. Quantum neural networks meet tensor networks: recurrent quantum neural networks

In this chapter we train recurrent quantum neural networks (RQNNs). They are a general-purpose quantum machine learning method for data with sequential structure, just like classical RNNs (see Subsection 2.6.4). In particular, we use RQNNs to design low- and high-pass filters and combat drifts. This work was done in collaboration with Tobias Osborne, Robert Salzmann and Victoria Schmiesing, with valuable comments from Kerstin Beer and Polina Feldmann.

For simplicity, we assume access to a supervisory source (see Fig. 5.1). There are many scenarios when this supervision is available, see Section 5.1.
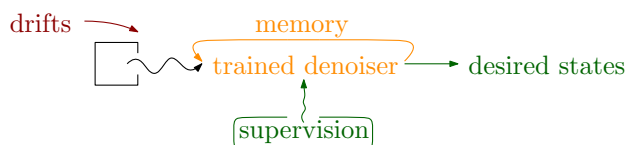
Figure 5.1.: A schematic representation of scenario considered in Chapter 5. We use supervisory source and recurrent RNNs to train denoisers for processes that require memory, such as drifts.

Classical RNNs is a valuable tool for quantum physics. Examples of applications include optimizing measurements for state tomography [474] and modelling quantum optical [13] and superconducting qubit [210] experiments, as well as optimizing dynamical decoupling protocols for quantum memories [38] and magnetic field estimation in atomic magnetometry [307].

There were several proposals to use quantum hardware for learning with memory. In [59] an evolution with non-linear Schrödinger equation was proposed as an activation function, and applications to classical data filtering were discussed. In [502, 503, 486] quantum Hopfield networks were introduced. [124] presents an RQNN closely inspired by the LSTM. It can be implemented on a quantum computer but is designed to work with classical data. [124, 53] introduced networks composed of highly structured neurons designed to mimic polynomial non-linearity; only learning with classical data is presented, even though these methods can, in principle, operate quantum data. Previously introduced RQNNs significantly differ from the approach presented in this chapter, both in neurons used and in tasks considered. In particular, the author is not aware of any other RQNN used to denoise quantum data.

5.1. Supervising sources

Unlike in Chapter 4, we assume access to a supervisory source. In this section we discuss a typical use case for learning with supervision.

If a low noise level is the only requirement taken into account during source development, such supervision might be unavailable (see Chapter 4). However, in practice this is often not the case and every machine is a compromise. Given two machines optimized for different requirements, one of them can supervise the post-processor for the other one. Such situations often arise in quantum metrology (see Subsection 2.7.3) as demonstrated by an example below.

There might be a need for transportable or/and cheap measurement apparatus. Often the signal of such apparatus can be improved by a compact post-processing device at a fraction of the cost of a more advanced analog sensor. For example, a suitable low-pass filter can cancel high-frequency noise. A post-processing device can be trained using a stationary high-precision supervisory apparatus.

Let us discuss an application of supervised denoisers in the field of geodesy. Modern atomic clocks are excellent gravimeters, as they are sensitive to gravitational time dilation. Their accuracy is sufficient to resolve gravity changes due to elevation shift by as little as $1cm$ [390, 389]. Comparison of the clock frequency with a reference in a well-measured location (alternatively, one can use an apparatus with two atomic species on-site [504]) yields gravitational potential. However, the most precise atomic clocks are assembled in laboratories and are too fragile to be transported. Applied geodesy would benefit from measurements in a large number of sometimes inhospitable locations, and a cost is always a consideration. As such, transportable atomic clocks that fit in a trailer are being developed [253, 202].

Conveniently, transportable quantum processing units are becoming available. Some state-of-the-art clocks have built-in quantum logic—a well-controlled additional quantum system that is coupled to atoms with clock transition and is typically used for cooling [100, 253, 522, 477]. Simultaneously, quantum computers are also experiencing minimization in order to fit into standard data center racks [459] (which are much smaller than a typical trailer). Finally, relevant quantum interfaces exist (see Subsection 2.7.4; although modern-day interfaces are far from perfect). As such, all the components needed for improved measurements are present: a low-noise reference clock used for supervision; a noisy but transportable clock—a source to be denoised; either built-in quantum logic or a compact quantum computer used as a hardware platform for a QNN; if applicable, an interface between the atomic clock and the quantum computer.

5.2. RQNNs: training algorithm for quantum channels with memory

To construct an RQNN, we first consider an FF QNN $\mathcal{N}$ with $i + m$ input qubits and $m + o$ output qubits. Consider L copies of $\mathcal{N}$. Let the output number $k \leq m$ of the jth copy be connected by a single link to the $i + k$ input of the copy number $j + 1$. For the first copy, let the

last m inputs be some fixed quantum state σ_0, for example $\sigma_0 = |\downarrow\rangle\langle\downarrow|^{\otimes m}$. Assume that L may vary. We have arrived at RQNNs, see Fig. 5.2.
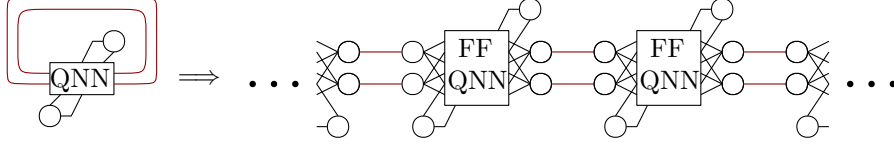
Figure 5.2.: An RQNN is constructed from a chain of FF QNNs such that some of the outputs of a previous unit are attached (dark red links) to the inputs of the next one.

Just like for the classical RNNs (see Subsection 2.6.4), the quantum learning algorithm can be directly lifted from the FF case. However, even for pure labels, there are many possible choices for the cost function. We are using the fidelity of labels with the outputs averaged over the sequence. We assume that each output of the network is consumed and measured by another device that does not communicate measurement outcomes to the RQNN. If the last layer of jth recurrent cell is $\rho_{[j]} \in \mathcal{T}(\mathcal{H}_o \otimes \mathcal{H}_m)$, $\mathcal{H}_o = \mathbb{C}^o, \mathcal{H}_m = \mathbb{C}^m$, its output is $\mathrm{Tr}_{\mathcal{H}_o}(\rho_{[j]})$ and the memory qubits are in the state $\mathrm{Tr}_{\mathcal{H}_m}(\rho_{[j]})$. Given a sequences of outputs

$$\left\{ \left\{ \rho_j^{(k)} \right\}_{j=1}^{T_k} \right\}_{k=1}^{N}, \quad \rho_j^{(k)} \in \mathcal{T}(\mathbb{C}^o), \tag{5.1}$$

and the corresponding labels

$$\left\{ \left\{ |\psi_j^{(k)}\rangle \right\}_{j=1}^{T_k} \right\}_{k=1}^{N}, \quad |\psi_j^{(k)}\rangle \in \mathbb{C}^o, \tag{5.2}$$

the cost function is

$$C(\mathcal{S}) = 1 - \frac{1}{\sum_k T_k} \sum_{k,j} \langle \psi_j^{(k)} | \rho_j^{(k)} | \psi_j^{(k)} \rangle. \tag{5.3}$$

If we do not assume that the output of each recurrent cell is measured before the next cell is applied, the memory and outputs may be entangled. It might be worthwhile to use a cost function that averages fidelity of a joint (and possibly entangled) output state of l cells with the corresponding l-cell label state. This is especially worthwhile if l-cell label states are entangled. Indeed, even pure labels cannot be written as a Sequence 5.2 if they are correlated. Generally, we can write pure labels as

$$\left\{ |\psi^{(k)}\rangle \right\}_{k=1}^{N}, \quad |\psi^{(k)}\rangle \in \mathbb{C}^{o \cdot T_k}. \tag{5.4}$$

For a l-local cost function, it is a good idea to have a relatively small l, as the fidelity tends to vanish as $l \to \infty$.

Just like for FF QNNs, RQNNs can also be trained with mixed labels.

As FF QNNs are universal, RQNNs are parameterizations of channels with memory defined in [337]. Classical RNNs are universal in the sense that they can simulate Turing machines with

linear overheads (see Subsection 2.6.4). A quantum equivalent of this statement is found in [337]: any quantum process in which outputs up to some time t do not depend on inputs at times $\tilde{t} > t$ can be decomposed into a concatenated memory channel.

RQNNs can also be regarded as QNNs with some parameters being quantum states. Quantum parameters can give at most a marginal improvement for ML tasks without memory [399]. However, RQNNs are suitable for learning with memory. We can treat the memory as quantum parameters. Classical parameters can be used multiple times, so once the network has been trained, there is no need to change them. Quantum parameters, on the other hand, cannot be cloned (see Corollary 2.5.10), and if the quantum parameter has been measured, it has to be prepared anew. We can view RQNN cells as preparation devices of quantum parameters for the subsequent cells. Similarly, we can treat memory qubits as a system that is being controlled by the inputs.

In the same sense as [49] presents a quantum optimization algorithm for MPS, RQNNs can be regarded as quantum algorithms for training matrix product quantum channels.

5.3. Filtering of interferometric outputs

Some of the most popular signal-enhancing techniques are based on bandwidth reduction [402, Section 6.8]. Such filters can be adaptive, e.g. [259]. These techniques filter out noise by analyzing the frequency with which different events occur in the data stream. Information about frequencies can be gathered only by having access to long sequences of inputs, necessitating recurrent architectures. In this section we construct a simple toy model that mimics an interferometry experiment and train quantum bandwidth filters for this model.

Consider a quantum source that is coupled to a time-dependent field $B(t)$. At times $t = j\omega$, $j \in \mathbb{N}$ the source outputs a superposition of two pre-defined states, let us call them $|\downarrow\rangle$ and $|\uparrow\rangle$. The relative phase between these two states encodes information about $B(j\omega)$. We call such a source an interferometer.

We assume that there are two interferometers; one of them is noisy, while the other encodes the information about the field faithfully. We denote the js outputs of the noisy device by $|\phi_j\rangle \equiv |\downarrow\rangle + e^{i\phi_j}|\uparrow\rangle$, and of the faithful one by $|\psi_j\rangle \equiv |\downarrow\rangle + e^{i\psi_j}|\uparrow\rangle$. We train an RQNN on the data set

$$\left\{ \left\{ |\phi_j^{(k)}\rangle, |\psi_j^{(k)}\rangle \right\}_{j=1}^{T_k} \right\}_{k=1}^N \tag{5.5}$$

consisting of N input-output sequences of lengths $\{T_k\}$. The trained RQNN generalizes well if given a sequence of inputs up to an arbitrary time t, $\{|\phi_j\rangle\}_{j=1}^t$, it can approximate $|\psi_t\rangle$ better than an FF QNN with the input $|\phi_t\rangle$.

Suppose that the desired output is related to the current input and past label by some function f:

$$\psi_j^{(k)} = f\left(\phi_j^{(k)}, \psi_{j-1}^{(k)}, j \right). \tag{5.6}$$

Given enough resources, for classical data $\left\{ \{\phi_j^{(k)}, \psi_j^{(k)}\}_{j=1}^{T_k} \right\}_{k=1}^{N}$ we can expect f to be learned perfectly. This follows form the universality of RNNs.

However, this is not the case for the quantum Data 5.5 even if f is affine.

Proposition 5.3.1 (No perfect quantum multiplication)**.** *There is no CPTP ε such that for a given number $a \notin \{0, 1\}$*

$$\forall \theta \; \varepsilon \left(|\theta\rangle\langle\theta| \right) = |a \cdot \theta\rangle\langle a \cdot \theta| \tag{5.7}$$

Proof. The Operation 5.7 is non-linear and, thus, not a quantum channel. $\qquad\square$

Corollary 5.3.1 (No perfect quantum adders)**.** *There is no CPTP ϵ such that for any $|x\rangle, |y\rangle \in \mathcal{H}$, with different values of continuous parameters x, y corresponding to different quantum states, $\epsilon \left(|x\rangle\langle x| \otimes |y\rangle\langle y| \right) = \rho_{x,x+y}$ such that $\mathrm{Tr}_1(\rho_{x,x+y}) = |x+y\rangle\langle x+y|$ and $\mathrm{Tr}_2(\rho_{x,x+y}) = |x\rangle\langle x|$.*

Proof. If such ϵ would have existed, we could have broadcasted x by choosing $y = 0$. By encoding into a tensor product of a sufficient number of states in the parameterized subspace, this would lead to a possibility of broadcast an arbitrary state—a contradiction to the Theorem 2.5.9. $\quad\square$

For a numerical search of approximate quantum adders, see also [168].

This dramatically complicates the design of quantum filters and manifests itself in RQNNs not reaching fidelity 1 for Data 5.5 with the relation between inputs and labels 5.6.

Example 5.3.2 (High- and low-pass filters)**.** Suppose that the input phases are uniformly distributed on the interval $[-\pi, \pi]$. Let us start with the simplest model for a low-frequency noise—a drift with constant velocity v. The training data can be described as

$$\psi_j = \phi_j - v \cdot j. \tag{5.8}$$

This noise is ubiquitous in both technological applications and every-day life. Classically, a rather simple high-pass filter cancels linear drifts, such as the one depicted in Fig. 5.3. Animals that fly in a windy environment routinely solve a more complicated version of this task (see [323] for how flies solve this problem).
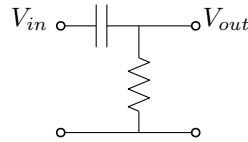
Figure 5.3.: A simple classical passive high-pass filter that is capable of removing linear drifts.

Let us assume that the signal changes slowly but is corrupted by a high-frequency noise. This noise is mitigatable by a low-pass filter that smoothens the variations. A simple model for the training data is

$$\psi_j = \alpha\phi_j + (\alpha - 1)\psi_j, \quad 0 < \alpha < 1. \tag{5.9}$$

This filtering can be familiar to the reader from the discussion of the gradient descent with adaptable learning rate, see Subsection 2.6.3. We use it as a part of the Nadam optimizer.

Let us combine the two models to obtain

$$\psi_j = \alpha \cdot (\phi_j - v \cdot j) + (1 - \alpha) \cdot \psi_{j-1}. \tag{5.10}$$

We observe that even a small amount of memory qubits greatly improves the performance of the trained filter. First, we choose $\alpha = 3/5$ and $v = 0$ (see Fig. 5.4) and train an RQNN. We see that memory can decrease the error $(1 - \langle \mathcal{F} \rangle$, where $\langle \mathcal{F} \rangle$ is the average fidelity) almost twofold. Even greater boosts in performance are available for $v \neq 0$. [1]
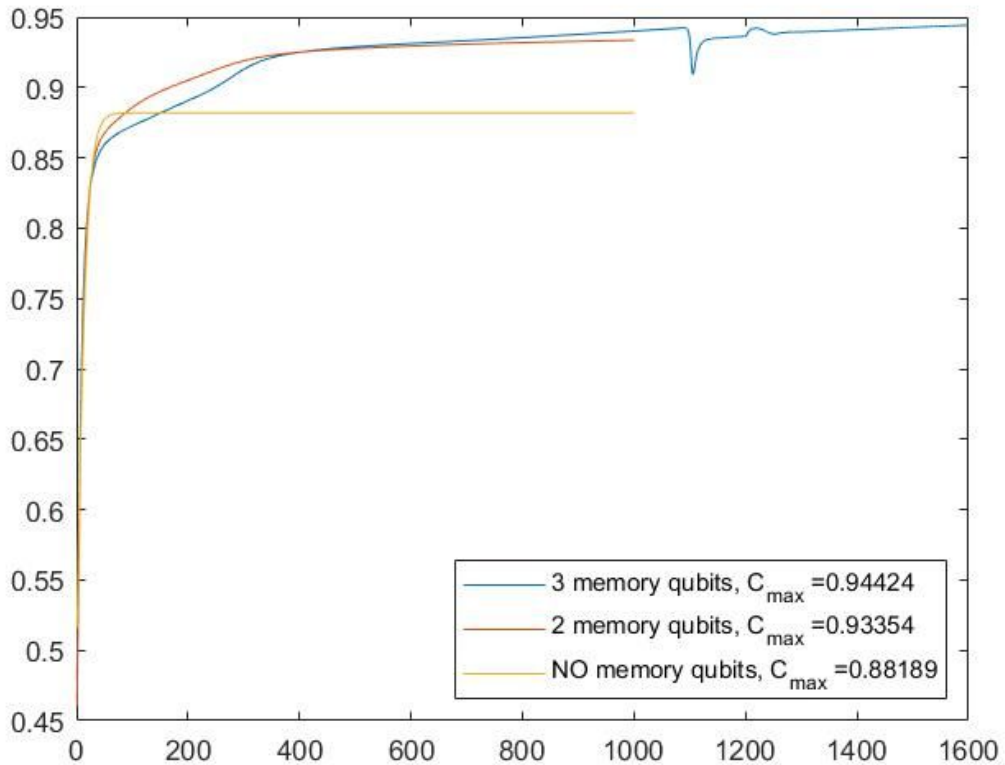
Figure 5.4.: Mitigating high frequency noise with RQNN and FF QNN. Vertical axis: cost function. Horizontal axis: training round. Parameters: $\alpha = 3/5$ and $v = 0$.

5.4. Conclusions

We have proposed a novel architecture for an RQNN. Our RQNN can work with quantum data and is efficiently implementable on quantum hardware. The training algorithm is identical to

---

[1]The plot for $v \neq 0$ that I have submitted my thesis with turned out to contain an error. While I have located and fixed the error and observed the performance boost, I cannot put it into this version of the thesis due to legal reasons. A version of this thesis with plots for $v \neq 0$ as well as estimates for FF QNN performance will be available on arXiv shortly after this version will be published. Sorry for the inconvenience.

that of FF QNNs discussed in Section 2.8.

The RQNNs we propose are parameterizations of quantum channels with memory. They can also be regarded as a quantum learning algorithm that varies over matrix product quantum channels.

We use RQNNs to train low-pass and high-pass filters that benefit from memory access. We see that RQNNs can significantly outperform FF QNNs.

Nevertheless, it is important to establish bounds on the performance of FF QNNs and to verify that the performance boost due to memory is not due to a poor training algorithm for FF QNNs.

It is also worthwhile to devise practical tasks that require entangled training data and check that RQNN can learn successfully. It is interesting to test various cost functions and understand how to choose their locality to maximize performance.

In this chapter we have considered supervised learning. It is desirable to develop unsupervised RQNN algorithms.

Memory qubits in the RQNN can be regarded as a system under control via interactions with input qubits. It is interesting to understand how to train the network to prepare a desired state in the memory register.

6. Conclusions and outlook

In this thesis we have investigated quantum channels that are composed of small building blocks. We have developed a framework for robust state preparation using these building blocks, see Fig. 6.1.
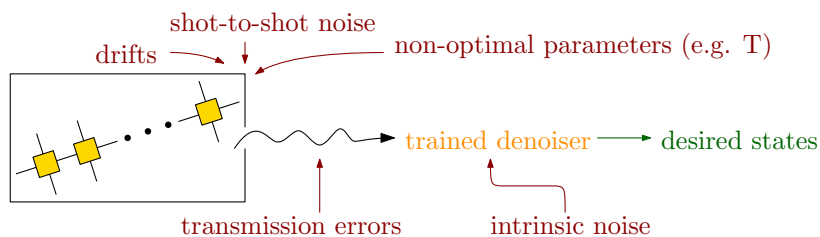
Figure 6.1.: A schematic representation of scenarios considered in this thesis. We have designed dissipative sources of MPDOs in Chapter 3. We have proceeded to develop denoisers that extrapolate states to more desirable parameter regions, cancel shot-to-shot noise and shield against transmission errors in Chapter 4. We have continued with the construction of denoisers that require memory, such as bandwidth filters.

We have developed an algorithm that designs local dissipative sources for an expressive class of physically relevant states—MPDOs. Our algorithm determines if a given linear space of MPDOs can be a stationary space for some k-local Lindbladian and, if so, outputs such a Lindbladian. Such a Linbladian can be constructed if and only if a local frustration-free quantum channel with given the MPDOs as the fixed space exists. This gives a recipe for the experimental preparation of MPDOs via dissipative engineering. It is also a good starting point to invent new integrable systems.

We have proceeded to train post-processors that mitigate the imperfections a source of states can have. We use machine learning with networks of quantum channels, also known as QNNs, to design post-processing devices.

In Chapter 4 we have focused on learning without a supervisory source. Successful learning can occur either through the design of data sets obtained from a tunable source, or through AE architectures. We have provided a proof-of-principle demonstration for ML systems that are able to extrapolate quantum states to lower temperatures and learn QECCs. We have introduced novel quantum AEs and use them to combat shot-to-shot errors. We have shown that these AEs can remove spin-flip errors and random unitary noise from small GHZ, W, Dicke, and cluster states. The protocol works for significant noise strength, for denoising multiple states parameterized by e.g. a phase, and even if the AE is implemented on non-perfect hardware.

As such, it is a compelling tool for practical tasks spanning data compression, optimization of quantum circuits, noise mitigation at interfaces, design of quantum repeaters without memory and information retrieval from quantum memory.

In Chapter 5 we have focused on denoising that requires memory. For this, we have introduced recurrent networks of quantum channels, also known as RQNNs. They can be interpreted as learning algorithms for matrix product quantum channels implemented on quantum hardware. We have observed that these networks can learn to implement quantum low- and high-pass filters.

Still, many open questions remain.

For one, it is unclear what kind of MPDOs have a parent Lindbladian. Moreover, it is both theoretically and practically interesting to understand how to construct an approximate parent Lindbladian and when it exists.

We have demonstrated proof-of-principle extrapolation and QECC search. It is worthwhile to bring this algorithm close to practice. For that, one needs to reduce the amount of resources required for successful extrapolation. Perhaps, deeper networks with drop out can help? To benefit from learned QEC protocols, one needs to find a training procedure that ensures that the learned code is fault-tolerant.

Considering the success of MPS and MPDO methods, it is easy to imagine many applications of matrix product quantum channels. As such, it is desirable to search for other tasks where RQNNs can shine, such as the control of a system by probing particles.

This thesis is theoretical in nature. How can we turn the proposed methods into an everyday tool? How to efficiently take into account that, even for local operations, the difficulty of implementation can vary significantly? What physical platforms and interfaces are mature enough? For the RQNNs, how can we avoid (by e.g. working with high-frequency sources) hitting the technological bottlenecks associated with memory?

There are several choices for a suitable cost function. For learning with mixed states, we have used the Hilbert-Schmidt norm. Should we switch to the fidelity now that there is a polynomial-time algorithm to estimate it? How local should the cost function for RQNNs be? Does it depend on the entanglement properties of training data?

We have discussed how to use QML to cancel paradigmatic examples of noise. However, several mechanisms are usually responsible for real-life noise. Can we design an out-of-the-box denoiser? Is a recurrent extrapolating autoencoder feasible?

Finally, there is a great need to understand and prove the bounds on the performance of quantum denoisers. Unfortunately, this thesis barely scratches the surface of such understanding.

A. Constructing local terms in a parent Lindbladian: an alternative algorithm outline

Let us outline an algorithm determines the smallest fixed space of any CP map such that ε is its fixed subspace and outputs a CP map that has such fixed space (see Subsection 2.1.3). It relies on any simultaneous block-diagonalization subroutine. We will denote a space of fixed points of some map ε as $fix(\varepsilon)$.

- Obtain finest block-diagonalization of ρ_0 and $\{\lambda_\alpha\}_{\alpha=1}^n$

- Some blocks can have further dependencies thus giving ρ_k in the resolution of $fix(\varepsilon)$. If two sets of blocks $\{A_\alpha\}_{\alpha=0}^n$ and $\{B_\alpha\}_{\alpha=0}^n$ are dependent, there exists a unitary U such that $A_\alpha = \kappa U B_\alpha U^\dagger$ for every α and $\kappa = \sqrt{\frac{\operatorname{tr}[A_0^2]}{\operatorname{tr}[B_0^2]}}$.

- Sequentially solve $A_\alpha = \kappa U B_\alpha U^\dagger$ with $U = \bigoplus_{i=1}^D U(d_i)$ starting from $\alpha = 0$, $D = 1$, $d_1 = dim(A_0)$. After each step choose the proper ordering of remaining degeneracies, update the list $\{d_i\}_{i=1}^D$ and perform a rotation $\{B_\beta\}_{\beta>\alpha} \to \{U B_\beta U^\dagger\}_{\beta>\alpha}$. If at some step there is no such U then the blocks are independent, otherwise the corresponding eigenvalue of ρ_k is κ.

The pseudocode is:

Algorithm 4 A map that has a given stabilizer code as its fixed point. Part 1 - input and output

---

Input:

1. a self-adjoint matrix $\rho_0 \in \mathcal{M}_d(\mathbb{C})$ such that $\operatorname{tr}[\rho_0] = 1$. Alternatively, we will denote it as $\lambda_0 \equiv \rho_0$.

2. a set of self-adjoint traceless matrices $\{\lambda_i \in \mathcal{M}_d(\mathbb{C})\}, i = 1, \ldots n$.

We would like $\Lambda = \{\rho_0 + \sum_{i=1}^n s_i \lambda_i | \{s_i\} \in S \subseteq \mathbb{R}^n\}$ to be a stabilizer space of interest. We will assume that ρ_0 is chosen to be as an inner point of this space.

Output: A space fix_Λ such that $\forall$ CPTP $\varepsilon(\rho) : \mathcal{M}_d(\mathbb{C}) \to \mathcal{M}_d(\mathbb{C})$ we have $\Lambda \subseteq fix(\varepsilon) \Rightarrow \Lambda \subseteq fix_\Lambda \subseteq fix(\varepsilon)$ and $\exists$ a CPTP $\tilde\varepsilon(\rho) : \mathcal{M}_d(\mathbb{C}) \to \mathcal{M}_d(\mathbb{C})$ such that $fix(\tilde\varepsilon) = fix_\Lambda$.

Algorithm 5 A map that has a given stabilizer code as its fixed point. Part 2 - support procedures

---

procedure BLOCKDIAGONALIZE($\{\lambda_\alpha \in \mathcal{M}_d(\mathbb{C})\}_{\alpha=0}^n$)   ▷ Outputs the finest possible block-diagonalization of a set of matrices

    return $\{\tilde{\lambda}_\alpha \in \mathcal{M}_d(\mathbb{C})\}_{\alpha=0}^n, B \in U(d)$; $D = \{\{d_0, k+1\}, \{d_i, m_i\}_{i=1}^k\}$ ▷ For $\alpha = 0, \ldots n$, the block-diagonal matrix $\tilde{\lambda}_\alpha = B\lambda_\alpha B^\dagger$. This unitary B block-diagonalizes a set of matrices, such that the first block is of dimension d_0 and is composed entirely of zeros, and blocks of dimension d_{i+1} follow blocks of dimension d_i for $i = 0, \le k - 1$. There are m_i blocks of dimension d_i. Moreover, blocks of dimension $d_i, i > 0$ are non-zero and $d_{i+1} \neq d_i$ for $i = 1, \ldots, k - 1$. D is an array that stores these dimensions of the blocks.

procedure EXTRACTBLOCK($\lambda \in \mathcal{M}_d(\mathbb{C}); D \in (\mathbb{N}_0 \times \mathbb{N})^{k+1}; i, j \in \mathbb{N}_0$)   ▷ Outputs the j-th block of matrix λ of dimension d_i. The dimensions of the blocks are specified by data structure D.

    return $\tilde{\lambda} \in \mathcal{M}_{d_i}$

procedure EXTRACTBLOCK($A \in \mathcal{M}_d(\mathbb{C})$; Blocks $\in \mathbb{N}^k; i \in \mathbb{N}$)   ▷ Outputs the i-th block of matrix A. The array Blocks has dimensions of blocks as its entries.

    return $\tilde{\lambda} \in \mathcal{M}_{d_i}$

procedure DIAGONALIZE($A \in \mathcal{M}_d$, self-adjoint)   ▷ Diagonalizes A in such a way that eigenvalues are in an increasing order.

    return $\tilde{A} \in \text{diag}(\mathbb{R}^d)$; $U \in U(d)$ ▷ The diagonal matrix $\tilde{A} = UAU^\dagger$, where the unitary U diagonalizes A. We also have the property that $A_{ii} \le A_{jj}$ for $i \le j$.

procedure DENOTECONSTANT($A, B \in \mathcal{M}_d$) ▷ Outputs a feasible κ such that $A = \kappa UBU^\dagger$

    return $\kappa \in \mathbb{R}_+$

procedure COMPAREBLOCKMATRIX($A, B \in \mathcal{M}_d$, self-adjoint; Blocks $\in \mathbb{N}^k, \sum_{i=1}^k \text{Blocks}_i = d$)   ▷ Denotes if there exists a unitary U with a block structure given by Blocks such that $A = UBU^\dagger$ and, if so, outputs a U and an encoding via SubBlocks of residual free parameters in the choice of U. Zero matrices are excluded from solutions.

    return IfEquivalent $\in \{0, 1\}$; $U \in \bigoplus_{i=1}^k U(\text{Blocks}_i)$; SubBlocks $\in \mathbb{N}^{\tilde{k}}, k \le \tilde{k} \le d$ ▷ IfEquivalent equals 0 if wanted U does not exist or if A or B are zero matrices, and IfEquivalent $= 1$ otherwise. SubBlocks consists of the dimensions of invariant subspaces in each of Blocks$_i$.

procedure COMPAREBLOCKSET($\{A_\alpha, B_\alpha \in \mathcal{M}_d$, self-adjoint$\}_{\alpha=0}^n$)   ▷ Denotes if there exist $\kappa \in \mathbb{R}_+$ and unitary U such that $A_\alpha = \kappa UB_\alpha U^\dagger$ for $\alpha = 0, \ldots m$ and, if so, outputs κ and U.

    return IfEquivalent $\in \{0, 1\}$; $\kappa \in \mathbb{R}_+$; $U \in U(d)$ ▷ IfEquivalent equals 0 if wanted κ and U does not exist and 1 otherwise.

procedure ARRAYINDUCEDPERMUTATION($d, k \in \mathbb{N}_0; O \in \{\mathbb{N}_0 \times \mathbb{N}_0\}^k$)   ▷ Let us have a basis $\{e_i\}_{i=1}^{d \cdot k}$. Let us assign the vectors $\{e_i\}_{i=j \cdot d}^{(j+1) \cdot d}$ a number $O_{j,2}$. At each step the procedure permutes two blocks $\{e_i\}_{i=j \cdot d}^{(j+1) \cdot d}$ and $\{e_i\}_{i=l \cdot d}^{(l+1) \cdot d}$, and two columns $O_{j,i}$ and $O_{l,i}$ in such a way, that $O_{j,2}$ becomes ordered. The output is the permutation matrix V that orders e_i's and an ordered $O_{j,i}$.

    return $V \in S(d \cdot k)$; $\tilde{O} \in \{\mathbb{N}_0 \times \mathbb{N}_0\}^k$ ▷ $\tilde{O}$ is an ordered O.

Algorithm 6 A map that has a given stabilizer code as its fixed point. Part 3 - essential routines

1: **procedure** CompareBlockMatrix(A, B; Blocks)
2:     $\{A, U_A\} \leftarrow$ Diagonalize(A)   ▷ A convenient basis
to check if $\exists\, U = \bigoplus_{i=1}^{\text{SizeOf(Blocks)}} U(\text{Blocks}_i) :\ A = UBU^\dagger$ is the eigenbasis of A, as in this
basis B should have block-diagonal structure $U = \bigoplus_{i=1}^{\text{SizeOf(Blocks)}} B_i :\ \dim(B_i) = \text{Blocks}_i$ if
such U exists. Moreover, there is no extra increase in computational cost in moving to this
basis as we will use the spectral decompositions of A and B anyway.
3:     $B \leftarrow U_A B U_A^\dagger$
4:     $j \leftarrow 1$
5:     **for** $i \leftarrow 1$, SizeOf(Blocks) **do**   ▷ These embedded cycles are responsible
for checking that B has a correct block-diagonal structure in the eigenbasis of A – this is a
necessary condition for existence of $U = \bigoplus_{i=1}^{\text{SizeOf(Blocks)}} U(\text{Blocks}_i)$.
6:         **for** $k \leftarrow j + \text{Blocks}_i, \dim(A)$ **do**
7:             **for** $l \leftarrow j, j + \text{Blocks}_i - 1$ **do**
8:                 **if** $A_{kl} \neq B_{kl}$ or $A_{lk} \neq B_{lk}$ **then**
9:                     **return** $\{0; \mathbb{1}_{\dim(A)}; \dim(A)\}$
10:         $j \leftarrow j + \text{Blocks}_i$
11:     $j \leftarrow 0$
12:     $SubBlockCount \leftarrow 1$
13:     $\text{SubBlocks}_1 \leftarrow \dim(A)$
14:     **for** $i \leftarrow 1$, SizeOf(Blocks) **do**
15:         $a \leftarrow$ ExtractBlock(A; Blocks; i)
16:         $b \leftarrow$ ExtractBlock(B; Blocks; i)
17:         $\{b, U_b\} \leftarrow$ Diagonalize(b)
18:         **if** a=b **then**
19:             $CurrentEigenvalue \leftarrow a_{11}$
20:             $d \leftarrow 0$ ▷ Dimension of a current sub-block. The initial value is 0 as $a_{11} \not> a_{11}$ (see
line 24) and d will increase to 1 at the first iteration ($k = 1$) of the cycle on line 21.
21:             **for** $k \leftarrow 1, \text{Blocks}_i$ **do**   ▷ This cycle is responsible for writing the i-th block of U
and storing the remaining redundancies in the array SubBlocks.
22:                 **for** $l \leftarrow 1, \text{Blocks}_i$ **do**
23:                     $U_{j+k, j+l} \leftarrow (U_b)_{kl}$ ▷ Writing up $U(\text{SubBlock}_i)$ in
$U = \bigoplus_{i=1}^{\text{SizeOf(Blocks)}} U(\text{Blocks}_i)$.
24:                 **if** $a_{kk} > CurrentEigenvalue$ **then**   ▷
The redundancies correspond to degenerate eigenvalues. Let us remind that the procedure
Diagonalize($\cdot$) outputs a matrix with eigenvalues in ascending order.
25:                     $\text{SubBlocks}_{SubBlockCount} \leftarrow d$
26:                     $SubBlockCount \leftarrow SubBlockCount + 1$
27:                     $CurrentEigenvalue \leftarrow a_{kk}$
28:                     $d \leftarrow 1$
29:                 **else**
30:                     $d \leftarrow d + 1$
31:         **else**
32:             **return** $\{0; \mathbb{1}_{\dim(A)}; \dim(A)\}$ ▷ Here we return
result IfEquivalent$= 0$ if desired U does not exist. The second and the third output should
not matter and is a subject to convention.
33:         $\text{SubBlocks}_{SubBlockCount} \leftarrow d$ ▷ Here we store the last value of SubBlock that can be
deduced from $i - th$ block of A and B.
34:         $j \leftarrow j + \text{Blocks}_i$
35:     **return** $\{1, U, \text{SubBlocks}\}$

Algorithm 7 A map that has a given stabilizer code as its fixed point. Part 3 - essential routines continued

---

1: **procedure** DENOTECONSTANT(A, B)

2:     **return** $\kappa \leftarrow \sqrt{\frac{\text{tr}[A^2]}{\text{tr}[B^2]}}$ ▷ There are a lot of measures that will give out the correct output. Simply taking the trace is not a good idea as A and B are typically traceless, and just taking the determinant may lead to troubles as these matrices may have some zero eigenvalues. The constant κ can always be chosen to be non-negative as the matrices that span the stable space can be chosen to be positive semi-definite. Any function of the type $\sqrt[2n]{\frac{\text{tr}[A^{2n}]}{\text{tr}[B^{2n}]}}$ will do the job, while for odd powers one may encounter troubles with $\text{tr}[B^{2n+1}] = 0$ for $B \neq 0$. Our choice of the way to obtain κ is dictated only by simplicity and low computational cost arguments.

1: **procedure** COMPAREBLOCKSET($\{A, B\}_{\alpha=0}^n$)

2:     $\kappa \leftarrow$ DENOTECONSTANT(A_0, B_0) ▷ Let us remind that κ is the same for all A_α and B_α.

3:     Blocks $\in \mathbb{N}$, Blocks$_1 = \dim(A)$ ▷ When we determine if $\exists U : A_0 = \kappa U B_0 U^\dagger$ we permit any unitary matrices, thus we can write $U \in \bigoplus_{i=1}^1 U(\dim(A))$.

4:     $U = \mathbb{1}_{\dim(A_0)}$ ▷ Let us remind that $\dim(A_0) = \dim(A_\alpha) = \dim(B_\alpha)$ for $\alpha = 0, \dots, n$.

5:     **for** $\alpha \leftarrow 0, n$ **do**

6:         $\{$IfEquivalent; V; Blocks$\} \leftarrow$ COMPAREBLOCKMATRIX($A_\alpha, \kappa \cdot B_\alpha$; Blocks)

7:         **if** IfEquivalent **then**

8:             $U \leftarrow V \circ U$

9:             **for** $j \leftarrow \alpha + 1, n$ **do**

10:                 $B_j \leftarrow V B_j V^\dagger$ ▷ Probably some B_j, $j > \alpha$ will require extra unitary rotation so that $A_j = U B_j U^\dagger$. However, this extra rotation is guaranteed to not spoil $A_i = U B_i U^\dagger$, $i \leq \alpha$ obtained earlier thanks to the structure of the unitaries stored in the array Blocks.

11:         **else**

12:             **return** $\{0; 1; \mathbb{1}_{\dim(A_0)}\}$ ▷ Here we return result IfEquivalent$= 0$ if desired U does not exist. The second and the third output should not matter and is a subject to a convention.

13:     **return** $\{1; \kappa; U\}$

Algorithm 8 A map that has a given stabilizer code as its fixed point. Part 4 - the main routine

1: **procedure** MapWithSpecifiedFixedPoints(**Input**) ▷ main
2:     $\{\{\lambda_\alpha\}_{\alpha=0}^n; B; D\} \leftarrow$ BlockDiagonalize$(\{\lambda_\alpha\}_{\alpha=0}^n)$
3:     **for** $i \leftarrow 2, D_{1,2}$ **do** ▷ After the block diagonalization one needs to check if the blocks can be further resolved. This cycle goes through the blocks of the same size with this aim.
4:         $l \leftarrow 1$ ▷ Counter responsible for the first block for which the existence a finer resolution is not yet known.
5:         $\lambda_\alpha^r \leftarrow$ ExtractBlock$(\lambda_\alpha; D; D_{i,1}, r)$
6:         $O \in \{\mathbb{N}_0 \times \mathbb{N}_0\}^{D_{i,2}}$ ▷ This array is responsible for storing all the ρ's for block size $D_{i,1}$. The diagonal components of ρ's are stored in the sub-array $O_{l,1}$ and are normalized so that the first eigenvalue of each ρ is 1. In order to write the resolution in the form $\bigoplus_i M_{D_{i,1}} \rho_{N_i}$, The blocks have to be ordered in a specific way. The sub-array $\{O_{l,2}\}_{l=2}^{D_{i,2}}$
7:         **for** $p \leftarrow 2, D_{i,2}$ **do**
8:             $O_{j,2} \leftarrow D_{i,2}$ ▷ The value $D_{i,2}$ is just a big enough value to denote the fact that we do not yet know the finer resolution of $fix(\varepsilon)$
9:         $N_i \leftarrow 0$ ▷ Number of ρ's needed for the finest resolution of blocks of dimension $D_{i,1}$.
10:         **while** $l < D_{i,2}$ **do**   ▷ This cycle and the embedded **for** cycle are responsible for checking which of the $D_{i,2}$ blocks are linearly dependent.
11:             $N_i \leftarrow N_i + 1$
12:             $O_{l,1} = 1$
13:             $O_{l,2} = l$
14:             $d \leftarrow 1$ ▷ Number of blocks known to be linearly dependent (after appropriate unitary rotation) on l's block.
15:             **for** $j \leftarrow l+1, D_{i,2}$ **do**   ▷ Here we denote the unitary rotation U and the proportionality constant κ of l'th and j'th block and if such U and κ exist.
16:                 $\{\text{IfEquivalent}; \kappa; U\} \leftarrow$ CompareBlockSet$(\{\lambda_\alpha^l, \lambda_\alpha^j\}_{\alpha=0}^n)$
17:                 **if** IfEquivalent **then**   ▷ If the is a linear dependence, we rotate the dependent block to an appropriate basis, increase the count of blocks dependent on l'th block, store the proportionality constant and permute the blocks so that dependent ones are in the consecutive order.
18:                     $B \leftarrow \left(\mathbb{1}_{D_{0,1}+\sum_{m=2}^{i-1} D_{m,1}\cdot D_{m,2}+D_{i,1}\cdot(j-1)} \oplus U \oplus \mathbb{1}_{D_{i,1}\cdot(D_{i,2}-j)+\sum_{m=i+1}^{D_{1,2}} D_{m,1}\cdot D_{m,2}}\right) \circ B$
19:                 $d \leftarrow d + 1$
20:                 $O_{j,1} = \kappa$
21:                 $O_{j,2} = l$
22:                 $\{V; O\} \leftarrow$ ArrayInducedPermutation$(D_{i,1}, D_{i,2}; O)$
23:                 $B \leftarrow \left(\mathbb{1}_{D_{0,1}+\sum_{m=2}^{i-1} D_{m,1}\cdot D_{m,2}} \oplus V \oplus \mathbb{1}_{\sum_{m=i+1}^{D_{1,2}} D_{m,1}\cdot D_{m,2}}\right) \circ B$
24:             $\rho_{[i,N_i]} \leftarrow \text{diag}\left[\{O_{q,1}\}_{q=l}^{l+d-1}\right]$
25:             $l \leftarrow l + d$
26:     **return** $\varepsilon(\cdot) = B\left(\mathbb{0}_{D_{1,1}} \oplus \bigoplus_{i=2}^{D_{1,2}}\left[\bigoplus_{j=1}^{N_i}\left(\mathbb{1}_{D_{i,1}}(\cdot)\mathbb{1}_{D_{i,1}} \otimes \text{tr}[\cdot]\rho_{[i,j]}\right)\right]\right)B^\dagger$

Acknowledgement

Many people have, knowingly or not, made my life more pleasant and interesting during my graduate studies. There were also many who have contributed to me being here in the first place. I am very fortunate to have such people in my life.

First of all, I would like to thank my supervisor, Prof. Tobias J. Osborne. I very much enjoyed working on the projects that were initiated and supervised by Tobias. I have learned valuable lessons, and not only in physics. There always was a space for questions about things outside my research and an expectation that healthy quality of life should be maintained. I am happy that Tobias not just allowed, but supported independent research projects.

I would like to mention my gratitude to my co-supervisor, Prof. Reinhard F. Werner. Reinhard remains for me the hero of good scientific practice and great wisdom, while simultaneously being very relatable and humanly. I will try to cultivate these qualities in me. Numerous lectures by Reinhard were a great bonus during my studies in Hannover.

I would also like to thank Prof. Markus Müller and Prof. Carsten Klempt for their interest in my work, enlightening discussions and agreeing to take part in the evaluation of my thesis, as well as Professors Robert König, Nilanjana Datta and Olexander Yakimenko for helping me with the decision to start my PhD in Hannover.

I would like to thank Alexander Hahn, Kerstin Beer, Victoria Schmiesing, Taras Kucherenko and especially Polina Feldmann for catching quite a few typos and inconsistencies in the earlier version of this document.

I would like to thank my students. It is flattering when such bright people as Daniel and Jannik seem impressed by my explanations. Both of them are also easy-going characters and, despite commonly propagated myths about the supervision of bachelor students, significantly helped me with my research.

I was lucky to have great co-authors. I do feel scientifically indebted to quite a few of them. Despite no success with some of the projects, I have managed to enjoy the process.

It felt great to be among a company of interesting, bright and enthusiastic people during my PhD. Being part of the QIG and SFB DQ-Mat was fun. As was going for a hike, a party, a conference or a scientific visit with Terry. It was reassuring to feel that Denis and Nastya are always able to find some time for me. It felt refreshing to have a small random adventure with Mitya, Ivan, Dasha, Denis and Vlad, as well as solving puzzles with Max no matter the location. It was nice to go on big random adventures with Taras and feel that he will not disappear or drift apart. I also have to thank Taras for the sharp turn of my PhD, as he brought my attention to the similarities between recurrent networks and MPDOs.

I would like to express my gratitude to my family. I have always felt surrounded by love, respect and support. I love all of you with whole my heart. Thanks for making me who I am.

Finally, I would like to express my sheer excitement and joy about the fact that Polina is nearby. Who knew that a short question about QFTs on a DQ-Mat meeting would lead to such a relaxing, but life-changing coffee break. It escalated so quickly... It seems that we have barely been on our first snowy adventure in Harz, and yet we have done so much together. Rowing, BECs, gardening, quantum autoencoders, extensive travel and cozy home, cross-country skiing and SUP, photography and playing musical instruments. Thanks for being my wife, Polya.

Bibliography

[1] https://github.com/DimaBoo/Joint-Block-Diagonalization.

[2] https://github.com/qigitphannover/DeepQuantumNeuralNetworks.

[3] https://github.com/DimaBoo/Parent-Lindbladian-MPDO.

[4] K. Abed-Meraim and A. Belouchrani. Algorithms for joint block diagonalization. In *2004 12th European Signal Processing Conference*, pages 209–212. IEEE, 2004.

[5] L. Accardi. Topics in quantum probability. *Phys. Rep.*, 77(3):169 – 192, 1981.

[6] F. Acernese et al. (Virgo Collaboration). Increasing the astrophysical reach of the advanced virgo detector via the application of squeezed vacuum states of light. *Phys. Rev. Lett.*, 123:231108, Dec 2019.

[7] T. Achache, L. Horesh, and J. Smolin. Denoising quantum states with quantum autoencoders – theory and applications. arXiv:2012.14714, 2020.

[8] A. Acín, I. Bloch, H. Buhrman, T. Calarco, C. Eichler, J. Eisert, D. Esteve, N. Gisin, S. Glaser, F. Jelezko, S. Kuhr, M. Lewenstein, M. Riede, P. Schmidt, R. Thew, A. Wallraff, I. Walmsley, and F. Wilhelm. The quantum technologies roadmap: a european community view. *New J. Phys.*, 20(8):080201, 2018.

[9] A. Acín, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani. Device-independent security of quantum cryptography against collective attacks. *Phys. Rev. Lett.*, 98:230501, Jun 2007.

[10] A. Acín, R. Gill, and N. Gisin. Optimal bell tests do not require maximally entangled states. *Phys. Rev. Lett.*, 95:210402, Nov 2005.

[11] A. Acín, S. Massar, and S. Pironio. Efficient quantum key distribution secure against no-signalling eavesdroppers. *New J. Phys.*, 8(8):126–126, aug 2006.

[12] A. Acín, S. Massar, and S. Pironio. Randomness versus nonlocality and entanglement. *Phys. Rev. Lett.*, 108:100402, Mar 2012.

[13] T. Adler, M. Erhard, M. Krenn, J. Brandstetter, J. Kofler, and S. Hochreiter. Quantum optical experiments modeled by long short-term memory. arXiv:1910.13804, 2019.

[14] I. Affleck, T. Kennedy, E.. Lieb, and H. Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Phys. Rev. Lett.*, 59:799–802, Aug 1987.

[15] I. Affleck, T. Kennedy, E. Lieb, and H. Tasaki. Valence bond ground states in isotropic quantum antiferromagnets. In *Condensed matter physics and exactly soluble models*, pages 253–304. Springer, 1988.

[16] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM J. Comput.*, 2008.

[17] D. Aharonov and T. Naveh. Quantum np—a survey. *arXiv:0210077*, 2002.

[18] D. Aharonov, W. Van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Rev.*, 50(4):755–787, 2008.

[19] S. Al-Assam, S. R. Clark, and D. Jaksch. The tensor network theory library. *J. Stat. Mech.: Theory Exp.*, 2017(9):093102, Sep 2017.

[20] V. V. Albert, B. Bradlyn, M. Fraas, and L. Jiang. Geometry and response of lindbladians. *Phys. Rev. X*, 6:041031, Nov 2016.

[21] S. Alexanderson, G. E. Henter, T. Kucherenko, and J. Beskow. Style-controllable speech-driven gesture synthesis using normalising flows. *Comput. Graph. Forum*, 39(2):487–496, 2020.

[22] R. Alicki and K. Lendi. *Quantum dynamical semigroups and applications*, volume 717. Springer, 2007.

[23] J. F. Allen and A. D. Misener. Flow of liquid helium ii. *Nature*, 141(3558):75–75, 1938.

[24] M. V. Altaisky. Quantum neural network. arXiv:0107012, 2001.

[25] F. A. An, E. J. Meier, and B. Gadway. Engineering a flux-dependent mobility edge in disordered zigzag chains. *Phys. Rev. X*, 8:031045, Aug 2018.

[26] F. Anders, A. Idel, P. Feldmann, D. Bondarenko, S. Loriani, K. Lange, J. Peise, M. Gersemann, B. Meyer, S. Abend, N. Gaaloul, C. Schubert, D. Schlippert, L. Santos, E. Rasel, and C. Klempt. Momentum entanglement for atom interferometry. arXiv:2010.15796, 2020.

[27] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, Mi. Gabureac, C. Eichler, and A. Wallraff. Repeated quantum error detection in a surface code. *Nat. Phys.*, pages 1–6, 2020.

[28] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, and E. A. Cornell. Observation of bose-einstein condensation in a dilute atomic vapor. *Science*, 269(5221):198–201, 1995.

[29] P. W. Anderson. Infrared catastrophe in fermi gases with local scattering potentials. *Phys. Rev. Lett.*, 18:1049–1051, Jun 1967.

[30] P. Andreasson, J. Johansson, S. Liljestrand, and M. Granath. Quantum error correction for the toric code using deep reinforcement learning. *Quantum*, 3:183, 2019.

[31] H. Aoki, N. Tsuji, M. Eckstein, M. Kollar, T. Oka, and P. Werner. Nonequilibrium dynamical mean-field theory and its applications. *Rev. Mod. Phys.*, 86:779–837, Jun 2014.

[32] T. Aoki, G. Takahashi, T. Kajiya, J. Yoshikawa, S. L. Braunstein, P. Van Loock, and A. Furusawa. Quantum error correction beyond qubits. *Nat. Phys.*, 5(8):541–546, 2009.

[33] I. Arad, A. Yu. Kitaev, Z. Landau, and U. Vazirani. An area law and sub-exponential algorithm for 1d systems. *arXiv:1301.1162*, 2013.

[34] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran. Machine learning method for state preparation and gate synthesis on photonic quantum computers. *Quantum Sci. Technol.*, 4(2):024004, 2019.

[35] F. Arute, K. Arya, R. Babbush, D. Bacon, J. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandão, D. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. Harrigan, M. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. Humble, S. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, D. S. Mandrà, J. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Yuezhen Niu, E. Ostby, A. Petukhov, J. Platt, C. Quintana, E. Rieffel, P. Roushan, N. Rubin, D. Sank, K. Satzinger, V. Smelyanskiy, K. Sung, M. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Jamie Yao, P. Yeh, A. Zalcman, H. Neven, and J. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[36] A. Aspect, J. Dalibard, and G. Roger. Experimental test of bell's inequalities using time-varying analyzers. *Phys. Rev. Lett.*, 49:1804–1807, Dec 1982.

[37] A. Aspect, P. Grangier, and G. Roger. Experimental tests of realistic local theories via bell's theorem. *Phys. Rev. Lett.*, 47:460–463, Aug 1981.

[38] M. August and X. Ni. Using recurrent neural networks to optimize dynamical decoupling for quantum memory. *Phys. Rev. A*, 95:012335, Jan 2017.

[39] K. Azuma, K. Tamaki, and H.-K. Lo. All-photonic quantum repeaters. *Nature communications*, 6(1):1–7, 2015.

[40] M. C. Bañuls, K. Cichy, I. Cirac, K. Jansen, and H. Saito. Thermal evolution of the schwinger model with matrix product operators. *Phys. Rev. D*, 92:034519, Aug 2015.

[41] M. C. Bañuls, K. Cichy, K. Jansen, and H. Saito. Chiral condensate in the schwinger model with matrix product operators. *Phys. Rev. D*, 93:094512, May 2016.

[42] M. C. Bañuls, N. Yao, S. Choi, M. Lukin, and I. Cirac. Dynamics of quantum information in many-body localized systems. *Phys. Rev. B*, 96:174201, Nov 2017.

[43] P. Baireuther, M. D. Caio, B. Criger, C. Beenakker, and T. O'Brien. Neural network decoder for topological color codes with circuit level noise. *New J. Phys.*, 21(1):013003, 2019.

[44] P. Baireuther, T. O'Brien, B. Tarasinski, and C. Beenakker. Machine-learning-assisted correction of correlated qubit errors in a topological code. *Quantum*, 2:48, 2018.

[45] M. C. Bañuls, R. Blatt, J. Catani, A. Celi, J. I. Cirac, M. Dalmonte, L. Fallani, K. Jansen, M. Lewenstein, S. Montangero, C. A. Muschik, B. Reznik, E. Rico, L. Tagliacozzo, K. Van Acoleyen, F. Verstraete, U.-J. Wiese, M. Wingate, J. Zakrzewski, and P. Zoller. Simulating lattice gauge theories within quantum technologies. *The European physical journal D*, 74(8):1–42, 2020.

[46] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, 2014.

[47] E. Barnard and L. F. A. Wessels. Extrapolation and interpolation in neural network classifiers. *IEEE Control Syst.*, 12(5):50–53, 1992.

[48] H. Barnum, C. M. Caves, C. A. Fuchs, R. Jozsa, and B. Schumacher. Noncommuting mixed states cannot be broadcast. *Phys. Rev. Lett.*, 76(15):2818–2821, Apr 1996.

[49] F. Barratt, J. Dborin, M. Bal, V. Stojevic, F. Pollmann, and A. Green. Parallel quantum simulation of large systems on small quantum computers. arXiv:2003.12087, 2020.

[50] J. Barrett, L. Hardy, and A. Kent. No signaling and quantum key distribution. *Phys. Rev. Lett.*, 95:010503, Jun 2005.

[51] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu. Interaction networks for learning about objects, relations and physics. arXiv:1612.00222, 2016.

[52] F. Battistel. General quantum error correction for mera codes. Master's thesis, Technische Universität München, 2017.

[53] J. Bausch. Recurrent quantum neural networks. arXiv:2006.14619, 2020.

[54] S. Becker, N. Datta, and R. Salzmann. Quantum zeno effect for open quantum systems. arXiv:2010.04121, 2020.

[55] T. Becker, L.-N. Wu, and A. Eckardt. Lindbladian approximation beyond ultra-weak coupling. arXiv:2012.14208, 2020.

[56] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf. Training deep quantum neural networks. *Nat. Commun.*, 11(1):1–6, 2020.

[57] K. Beer, D. Bondarenko, A. Hahn, M. Kalabakov, N. Knust, L. Niermann, T. J. Osborne, C. Schridde, S. Seckmeyer, D. E. Stiegemann, and R. Wolf. From categories to anyons: a travelogue. arXiv:1811.06670, 2018.

[58] K. Beer, M. Khosla, J. Köhler, and T. J. Osborne. Quantum machine learning of graph-structured data. arXiv:2103.10837, 2021.

[59] L. Behera, I. Kar, and A. C. Elitzur. Recurrent quantum neural network and its applications. In *The Emerging Physics of Consciousness*, pages 327–350. Springer, 2006.

[60] J. S Bell. On the einstein-podolsky-rosen paradox. *Physics Physique Физика*, 1(3):195, 1964.

[61] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444, 1997.

[62] A. Belouchrani, M. G. Amin, and K. Abed-Meraim. Direction finding in correlated noise fields based on joint block-diagonalization of spatio-temporal correlation matrices. *IEEE Signal Processing Letters*, 4(9):266–268, 1997.

[63] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.*, 4(4):043001, nov 2019.

[64] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural. Netw. Learn. Syst.*, 5(2):157–166, 1994.

[65] C. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. Smolin, and W. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.*, 76:722–725, Jan 1996.

[66] C. B. Bennett, G. Brassard, S. Breidbard, and S. Wiesner. Quantum cryptography, or unforgeable subway tokens. In *Advances in Cryptology: Proceedings of CRYPTO '82*, pages 267–275. Plenum, 1982.

[67] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin. Experimental quantum cryptography. *Journal of cryptology*, 5(1):3–28, 1992.

[68] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, volume 175, page 8, 1984.

[69] C. H. Bennett and G. Brassard. Experimental quantum cryptography: The dawn of a new era for quantum cryptography: The experimental prototype is working]. *SIGACT News*, 20(4):78–80, November 1989.

[70] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, Mar 1993.

[71] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, Nov 1996.

[72] C. Bény and O. Oreshkov. General conditions for approximate quantum error correction and near-optimal recovery channels. *Phys. Rev. Lett.*, 104:120501, Mar 2010.

[73] C. Bény and F. Richter. Algebraic approach to quantum theory: a finite-dimensional guide. *arXiv:1505.03106*, May 2015.

[74] B. Bergh, R. Salzmann, and N. Datta. The $\alpha \to 1$ limit of the sharp quantum rényi divergence. arXiv:2102.06576, 2021.

[75] A. Bermudez, L. Tagliacozzo, G. Sierra, and P. Richerme. Long-range heisenberg models in quasiperiodically driven crystals of trapped ions. *Phys. Rev. B*, 95:024431, Jan 2017.

[76] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, F. , Schmidt-Kaler, M. Biercuk, R. Blatt, S. Benjamin, and M. Müller. Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation. *Phys. Rev. X*, 7(4):041061, 2017.

[77] M. Berta, F. J. S. L. Brandão, J. Haegeman, V. Scholz, and F. Verstraete. Thermal states as convex combinations of matrix product states. *Phys. Rev. B*, 98:235154, Dec 2018.

[78] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik. Noisy intermediate-scale quantum (nisq) algorithms. arXiv:2101.08448, 2021.

[79] K. Bharti, T. Haug, V. Vedral, and L.-C. Kwek. Machine learning meets quantum foundations: A brief survey. *AVS Quantum Sci.*, 2(3):034101, 2020.

[80] J. D. Biamonte and P. J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78(1), Jul 2008.

[81] R. Blatt, G. J. Milburn, and A. Lvovksy. The 20th anniversary of quantum state engineering. *J. Phys. B: At. Mol. Opt. Phys*, 46(100201):100201, 2013.

[82] O. Boada, A. Celi, J. I. Latorre, and M. Lewenstein. Quantum simulation of an extra dimension. *Phys. Rev. Lett.*, 108(13):133001, 2012.

[83] D. Bohm and Y. Aharonov. Discussion of experimental proof for the paradox of einstein, rosen, and podolsky. *Phys. Rev.*, 108:1070–1076, Nov 1957.

[84] H. Bombín. An introduction to topological quantum codes. *arXiv:1311.0277*, 2013.

[85] D. Bondarenko and P. Feldmann. Quantum autoencoders to denoise quantum data. *Phys. Rev. Lett.*, 124:130502, Mar 2020.

[86] L. Bonnes and A. Läuchli. Superoperators vs. trajectories for matrix product state simulations of open quantum system: A case study. arXiv:1411.4831, 2014.

[87] K. Boothby, P. Bunyk, J. Raymond, and A. Roy. Next-generation topology of d-wave quantum processors. *arXiv:2003.00133*, 2020.

[88] P. Bordia, H. P. Lüschen, S. S. Hodgman, M. Schreiber, I. Bloch, and U. Schneider. Coupling identical one-dimensional many-body localized systems. *Phys. Rev. Lett.*, 116:140401, Apr 2016.

[89] J. Borregaard, H. Pichler, T. Schröder, M. D. Lukin, P. Lodahl, and A. S. Sørensen. One-way quantum repeater based on near-deterministic photon-emitter interfaces. *Phys. Rev. X*, 10:021071, Jun 2020.

[90] D. Boschi, S. Branca, F. De Martini, L. Hardy, and S. Popescu. Experimental realization of teleporting an unknown pure quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 80:1121–1125, Feb 1998.

[91] A. Bouland, B. Fefferman, Z. Landau, and Y. Liu. Noise and the frontier of quantum supremacy. arXiv:2102.01738, 2021.

[92] J. E. Bourassa, R. Alexander, M. Vasmer, A. Patil, I. Tzitrin, T. Matsuura, D. Su, B. Baragiola, S. Guha, G. Dauphinais, K. Sabapathy, N. Menicucci, and I. Dhand. Blueprint for a scalable photonic fault-tolerant quantum computer. arXiv:2010.02905, 2020.

[93] D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger. Experimental quantum teleportation. *Nature*, 390(6660):575–579, 1997.

[94] C. C. Bradley, C. A. Sackett, J. J. Tollett, and R. G. Hulet. Evidence of bose-einstein condensation in an atomic gas with attractive interactions. *Phys. Rev. Lett.*, 75:1687–1690, Aug 1995.

[95] S. L. Braunstein and C. M. Caves. Statistical distance and the geometry of quantum states. *Phys. Rev. Lett.*, 72:3439–3443, May 1994.

[96] S. Bravyi, D. Gosset, and R. König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018.

[97] S. Bravyi, D. Gosset, R. König, and M. Tomamichel. Quantum advantage with noisy shallow circuits. *Nat. Phys.*, 16(10):1040–1045, 2020.

[98] S. Bravyi and M. Hastings. On complexity of the quantum ising model. *Comm. Math. Phys.*, 349(1):1–45, 2017.

[99] N. Breuckmann and X. Ni. Scalable neural network decoders for higher dimensional quantum codes. *Quantum*, 2:68, 2018.

[100] S. M. Brewer, J.-S. Chen, A. M. Hankin, E. R. Clements, C. W. Chou, D. J. Wineland, D. B. Hume, and D. R. Leibrandt. ^{27}al$^+$ quantum-logic clock with a systematic uncertainty below 10^{-18}. *Phys. Rev. Lett.*, 123:033201, Jul 2019.

[101] H. J. Briegel and G. De las Cuevas. Projective simulation for artificial intelligence. *Sci. Rep.*, 2(1):1–16, 2012.

[102] H. J. Briegel, W. Dür, J. I. Cirac, and P. Zoller. Quantum repeaters: The role of imperfect local operations in quantum communication. *Phys. Rev. Lett.*, 81:5932–5935, Dec 1998.

[103] H. J. Briegel and R. Raussendorf. Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett.*, 86:910–913, Jan 2001.

[104] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage. Trapped-ion quantum computing: Progress and challenges. *Appl. Phys. Rev.*, 6(2):021314, 2019.

[105] A. Burchardt and Z. Raissi. Stochastic local operations with classical communication of absolutely maximally entangled states. *Phys. Rev. A*, 102:022413, Aug 2020.

[106] D. Burgarth, G. Chiribella, V. Giovannetti, P. Perinotti, and K. Yuasa. Ergodic and mixing quantum channels in finite dimensions. *New J. Phys.*, 15(7):073045, jul 2013.

[107] D. Buterakos, E. Barnes, and S. E. Economou. Deterministic generation of all-photonic quantum repeaters from solid-state emitters. *Phys. Rev. X*, 7:041023, Oct 2017.

[108] Y. Cai, G. Cheng, and D. Shi. Solving the general joint block diagonalization problem via linearly independent eigenvectors of a matrix polynomial. *Numer. Linear Algebra Appl.*, 26(4):e2238, 2019.

[109] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane. Quantum error correction via codes over gf (4). *IEEE Trans. Inf. Theory*, 44(4):1369–1387, 1998.

[110] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54(2):1098–1105, Aug 1996.

[111] P. Campagne-Ibarcq, A. Eickbusch, S. Touzard, E. Zalys-Geller, N. E. Frattini, V. V. Sivak, P. Reinhold, S. Puri, S. Shankar, R. J. Schoelkopf, L. Frunzio, M. Mirrahimi, and M. H. Devoret. Quantum error correction of a qubit encoded in grid states of an oscillator. *Nature*, 584(7821):368–372, 2020.

[112] A. Canabarro, S. Brito, and R. Chaves. Machine learning nonlocal correlations. *Phys. Rev. Lett.*, 122:200401, May 2019.

[113] Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik. Quantum neuron: an elementary building block for machine learning on quantum computers. arXiv:1711.11240, 2017.

[114] G. Carleo, Ig. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, Dec 2019.

[115] A. Castro, J. Werschnik, and E. K. U. Gross. Controlling the dynamics of many-electron systems from first principles: A combination of optimal control and time-dependent density-functional theory. *Phys. Rev. Lett.*, 109:153603, Oct 2012.

[116] C. M. Caves. Quantum-mechanical noise in an interferometer. *Phys. Rev. D*, 23:1693–1708, Apr 1981.

[117] A. Çakan, I. Cirac, and Ba nuls. M.C. Approximating the long time average of the density operator: Diagonal ensemble. arXiv:2011.01257, 2020.

[118] K. Chabuda, J. Dziarmaga, T. J. Osborne, and R. Demkowicz-Dobrzanski. Tensor networks for quantum metrology. *Nat. Commun.*, 11(1):1–12, 2020.

[119] C. Chamberland and P. Ronagh. Deep neural decoders for near term fault-tolerant experiments. *Quantum Sci. Technol.*, 3(4):044002, 2018.

[120] H. F. Chau. Practical scheme to share a secret key through a quantum channel with a 27.6% bit error rate. *Phys. Rev. A*, 66:060302, Dec 2002.

[121] S. Chehade and A. Vershynina. Quantum entropies. *Scholarpedia*, 14(2):53131, 2019. revision #190401.

[122] C. Chen, K. Kato, and F. G. S. L. Brandão. Matrix product density operators: when do they have a local parent hamiltonian? arXiv:2010.14682, 2020.

[123] M.-C. Chen, C. Wang, F.-M. Liu, J.-W. Wang, C. Ying, Z.-X. Shang, Y. Wu, M. Gong, H. Deng, F. Liang, Q. Zhang, C.-Z. Peng, X. Zhu, A. Cabello, C.-Y. Lu, and J.-W. Pan. Ruling out real-number description of quantum mechanics. arXiv:2103.08123, 2021.

[124] S. Y.-C. Chen, S. Yoo, and Y.-L. L. Fang. Quantum long short-term memory. arXiv:2009.01783, 2020.

[125] Y.-A. Chen, Q. Zhang, T.-Y. Chen, W.-Q. Cai, S.-K. Liao, J. Zhang, K. Chen, J. Yin, J.-G. Ren, Z. Chen, S.-L. Han, Q. Yu, K. Liang, F. Zhou, X. Yuan, M.-S. Zhao, T.-Y. Wang, X. Jiang, L. Zhang, W.-Y. Liu, Y. Li, Q. Shen, Y. Cao, C.-Y. Lu, R. Shu, J.-Y. Wang, L. Li, N.-L. Liu, F. Xu, X.-B. Wang, C.-Z. Peng, and J.-W. Pan. An integrated space-to-ground quantum communication network over 4,600 kilometres. *Nature*, 589(7841):214–219, 2021.

[126] Y.-H. Chen, M.-J. Lee, I-C. Wang, S. Du, Y.-F. Chen, Y.-C. Chen, and I. A. Yu. Coherent optical memory with high storage efficiency and large fractional delay. *Phys. Rev. Lett.*, 110:083601, Feb 2013.

[127] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. B. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, R. Ozeri, and D. J. Wineland. Realization of quantum error correction. *Nature*, 432(7017):602–605, 2004.

[128] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv:1406.1078, 2014.

[129] J. Choi, S. Hild, J. Zeiher, P. Schauß, A. Rubio-Abadal, T. Yefsah, V. Khemani, D. A. Huse, I. Bloch, and C. Gross. Exploring the many-body localization transition in two dimensions. *Science*, 352(6293):1547–1552, 2016.

[130] I. L. Chuang, D. W. Leung, and Y. Yamamoto. Bosonic quantum codes for amplitude damping. *Phys. Rev. A*, 56:1114–1125, Aug 1997.

[131] I. Cirac, J. Garre-Rubio, and D. Pérez-García. Mathematical open problems in projected entangled pair states. *Revista Matemática Complutense*, 32(3):579–599, Jul 2019.

[132] Bo. S Cirel'son. Quantum generalizations of bell's inequality. *Lett. Math. Phys.*, 4(2):93–100, 1980.

[133] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.

[134] P. T. Cochrane, G. J. Milburn, and W. J. Munro. Macroscopically distinct quantum-superposition states as a bosonic code for amplitude damping. *Phys. Rev. A*, 59(4):2631–2634, Apr 1999.

[135] P. Contreras-Tejada, C. Palazuelos, and J. de Vicente. Resource theory of entanglement with a unique multipartite maximally entangled state. *Phys. Rev. Lett.*, 122:120503, Mar 2019.

[136] A. D. Córcoles, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow. Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. *Nat. Commun.*, 6(1):1–10, 2015.

[137] D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. S. Somaroo. Experimental quantum error correction. *Phys. Rev. Lett.*, 81(10):2152, 1998.

[138] J. Cotler, S. Choi, A. Lukin, H. Gharibyan, T. Grover, M. E. Tai, M. Rispoli, R. Schittko, P. M. Preiss, A. M. Kaufman, M. Greiner, H. Pichler, and P. Hayden. Quantum virtual cooling. *Phys. Rev. X*, 9:031013, Jul 2019.

[139] H. Cramér. A contribution to the theory of statistical estimation. *Scand. Actuar. J.*, 1946(1):85–94, 1946.

[140] Harald Cramér. *Mathematical methods of statistics*. Almqvist & Wiksells, 1945.

[141] J. Cramer, N. Kalb, M. A. Rol, B. Hensen, M. S. Blok, M. Markham, D. J. Twitchen, R. Hanson, and T. H. Taminiau. Repeated quantum error correction on a continuously encoded qubit by real-time feedback. *Nat. Commun.*, 7(1):1–7, 2016.

[142] T. S. Cubitt, A. Lucia, S. Michalakis, and D. Perez-Garcia. Stability of local quantum dissipative systems. *Commun. Math. Phys.*, 337:1275–1315, 2015.

[143] J. Cui, I. Cirac, and M. Banuls. Variational matrix product operators for the steady state of dissipative quantum systems. *Phys. Rev. Lett.*, 114(22):220601, 2015.

[144] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[145] A. J. da Silva, T. B. Ludermir, and W. R. de Oliveira. Quantum perceptron over a field and neural network architecture selection in a quantum computer. *Neural Networks*, 76:55–64, 2016.

[146] A. Daley. Quantum trajectories and open many-body quantum systems. *Adv. Phys.*, 63(2):77–149, Mar 2014.

[147] J. Dalibard. Introduction to the physics of artificial gauge fields. *Quantum Matter at Ultralow Temperatures*, 2015.

[148] K. B. Davis, M.-O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. M. Kurn, and W. Ketterle. Bose-einstein condensation in a gas of sodium atoms. *Phys. Rev. Lett.*, 75:3969–3973, Nov 1995.

[149] C. Dawson and M. Nielsen. The solovay-kitaev algorithm. *arXiv:0505030*, 2005.

[150] E. de Klerk, C. Dobre, and D. V. Pasechnik. Numerical block diagonalization of matrix*-algebras with application to semidefinite programming. *Mathematical programming*, 129(1):91, 2011.

[151] G. De las Cuevas, T. Cubitt, I. Cirac, M. Wolf, and D. Pérez-García. Fundamental limitations in the purifications of tensor networks. *J. Math. Phys.*, 57(7):071902, 2016.

[152] G. De las Cuevas, T. Drescher, and T. Netzer. Separability for mixed states with operator schmidt rank two. *Quantum*, 3:203, Dec 2019.

[153] G. De las Cuevas, A. Klingler, and T. Netzer. Approximate tensor decompositions: disappearance of many separations. arXiv:2004.10219, 2020.

[154] G. De las Cuevas and T. Netzer. Mixed states in one spatial dimension: Decompositions and correspondence with nonnegative matrices. *J. Math. Phys.*, 61(4):041901, Apr 2020.

[155] G. De las Cuevas, N. Schuch, D. Perez-Garcia, and I Cirac. Purifications of multipartite states: limitations and constructive methods. *New J. Phys.*, 15(12):123021, Dec 2013.

[156] M. de Leeuw, C. Paletta, and B. Pozsgay. Constructing integrable lindblad superoperators. arXiv:2101.08279, 2021.

[157] B. de Neeve, T. L. Nguyen, T. Behrle, and J. Home. Error correction of a logical grid state qubit by dissipative pumping. *arXiv:2010.09681*, 2020.

[158] I. de Vega and D. Alonso. Dynamics of non-markovian open quantum systems. *Rev. Mod. Phys.*, 89:015001, Jan 2017.

[159] N. Delfosse, C. Okay, J. Bermejo-Vega, D. Browne, and R. Raussendorf. Equivalence between contextuality and negativity of the wigner function for qudits. *New J. Phys.*, 19(12):123024, 2017.

[160] D.-L. Deng. Machine learning detection of bell nonlocality in quantum many-body systems. *Phys. Rev. Lett.*, 120:240402, Jun 2018.

[161] E. Dennis, A. Yu. Kitaev, A. Landahl, and J. Preskill. Topological quantum memory. *J. Math. Phys.*, 43(9):4452–4505, 2002.

[162] D. E. Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A*, 400(1818):97–117, 1985.

[163] D. E. Deutsch. Quantum computational networks. *Proc. R. Soc. Lond. A*, 425(1868):73–90, 1989.

[164] D. E. Deutsch, A. Barenco, and A. Ekert. Universality in quantum computation. *Proc. R. Soc. Lond. A*, 449(1937):669–677, 1995.

[165] R. H. Dicke. Coherence in spontaneous radiation processes. *Phys. Rev.*, 93:99–110, Jan 1954.

[166] P. K. Diederik and J. Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2017.

[167] S. Diehl, A. Micheli, A. Kantian, B Kraus, H. P. Büchler, and P. Zoller. Quantum states and phases in driven open quantum systems with cold atoms. *Nat. Phys.*, 4(11):878–883, 2008.

[168] Y. Ding, L. Lamata, M. Sanz, X. Chen, and E. Solano. Experimental implementation of a quantum autoencoder via quantum adders. *Adv. Quantum Technol.*, page 1800065, 2019.

[169] D. P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9-11):771–783, Sep 2000.

[170] M. Donald, M. Horodecki, and O. Rudolph. The uniqueness theorem for entanglement measures. *Journal of Mathematical Physics*, 43(9):4252–4272, Sep 2002.

[171] D. Dong and I.R. Petersen. Quantum control theory and applications: a survey. *IET Control Theory Appl.*, 4(12):2651–2671, Dec 2010.

[172] T. Dozat. Incorporating nesterov momentum into adam. `http://cs229.stanford.edu/proj2015/054_report.pdf`, 2015.

[173] W. Dür. Multipartite entanglement that is robust against disposal of particles. *Phys. Rev. A*, 63:020303, Jan 2001.

[174] W. Dür and H.-J. Briegel. Stability of macroscopic entanglement under decoherence. *Phys. Rev. Lett.*, 92:180403, May 2004.

[175] W. Dür, R. Lamprecht, and S. Heusler. Towards a quantum internet. *Eur. J. Phys.*, 38(4):043001, 2017.

[176] W. Dür, M. Skotiniotis, F. Fröwis, and B. Kraus. Improved quantum metrology using quantum error correction. *Phys. Rev. Lett.*, 112:080801, Feb 2014.

[177] W. Dür, G. Vidal, and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. *Phys. Rev. A*, 62:062314, Nov 2000.

[178] F. A. Dziemba. Robustness of qma against witness noise. *arXiv:1611.07332*, 2016.

[179] J. Eggert. Using quantum neural networks to cool down thermal states. Bachelor's thesis, Leibniz Universität Hannover, Jul 2020.

[180] A. Einstein. Quantentheorie des einatomigen idealen gases. zweite abhandlung. *Sitzungsberichte der Preussischen Akademie der Wissenschaften*, 3, 1925.

[181] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[182] J. Eisert. Entanglement and tensor network states. *arXiv:1308.3318*, 2013.

[183] J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82(1):277–306, Feb 2010.

[184] A. K. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991.

[185] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Conference on learning theory*, pages 907–940. PMLR, 2016.

[186] R. J. Elliott, P. Pfeuty, and C. Wood. Ising model with a transverse field. *Phys. Rev. Lett.*, 25:443–446, Aug 1970.

[187] J. L. Elman and D. Zipser. Learning the hidden structure of speech. *J. Acoust. Soc. Am.*, 83(4):1615–1626, 1988.

[188] T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21, 2019.

[189] G. Evenbly and G. Vidal. Algorithms for entanglement renormalization. *Phys. Rev. B*, 79(14):144108, April 2009.

[190] G. Evenbly and G. Vidal. Local Scale Transformations on the Lattice with Tensor Network Renormalization. *Phys. Rev. Lett.*, 116(4):040401, January 2016.

[191] F. Ewert, M. Bergmann, and P. van Loock. Ultrafast long-distance quantum communication with static linear optics. *Phys. Rev. Lett.*, 117:210501, Nov 2016.

[192] P Facchi and S Pascazio. Quantum zeno dynamics: mathematical and physical aspects. *J. Phys. A: Math. Theor.*, 41(49):493001, Oct 2008.

[193] M. Fanizza, M. Rosati, M. Skotiniotis, J. Calsamiglia, and V. Giovannetti. Beyond the swap test: optimal estimation of quantum state overlap. *Phys. Rev. Lett.*, 124(6), Feb 2020.

[194] M. Fannes, B. Nachtergaele, and R. F. Werner. Construction and study of exact ground states for a class of quantum antiferromagnets. *Revista Brasileira de Física*, 19(3), 1989.

[195] M. Fannes, B. Nachtergaele, and R. F. Werner. Exact antiferromagnetic ground states of quantum spin chains. *EPL*, 10(7):633, 1989.

[196] M. Fannes, B. Nachtergaele, and R. F. Werner. Finitely correlated states on quantum spin chains. *Comm. Math. Phys.*, 144(3):443–490, 1992.

[197] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475, 2001.

[198] E. Farhi and H. Neven. Classification with quantum neural networks on near term processors. arXiv:1802.06002, 2018.

[199] T. Farrelly. A review of Quantum Cellular Automata. *Quantum*, 4:368, November 2020.

[200] T. Farrelly, R. J. Harris, N. A. McMahon, and T. M. Stace. Parallel decoding of multiple logical qubits in tensor-network codes. arXiv:2012.07317, 2020.

[201] T. Farrelly, R. J. Harris, N. A. McMahon, and T. M. Stace. Tensor-network codes. arXiv:2009.10329, 2020.

[202] V. F. Fateev, V. P. Sysoev, and E. A. Rybakov. Experimental measurement of gravitational time dilation using transportable quantum clocks. *Measurement Techniques*, 59(4):402–404, 2016.

[203] P. Feldmann. *Generalized Quantum Phase Transitions for Quantum-State Engineering in Spinor Bose-Einstein Condensates.* PhD thesis, Leibniz Universität Hannover, 2021.

[204] C. Févotte and F. Theis. Pivot selection strategies in jacobi joint block-diagonalization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 177–184. Springer, 2007.

[205] R. Feynman. Simulating physics with computers. *Int. J. Theor. Phys*, 21(6/7), 1982.

[206] R. A. Fisher. Theory of statistical estimation. In *Math. Proc. Camb. Phil. Soc.*, volume 22, pages 700–725. Cambridge University Press, 1925.

[207] M. Fitzpatrick, N. M. Sundaresan, A. C. Y. Li, J. Koch, and A. A. Houck. Observation of a dissipative phase transition in a one-dimensional circuit qed lattice. *Phys. Rev. X*, 7:011016, Feb 2017.

[208] F. Flamini, N. Spagnolo, and F. Sciarrino. Photonic quantum information processing: a review. *Rep. Prog. Phys.*, 82(1):016001, 2018.

[209] C. Flühmann, T. L. Nguyen, M. Marinelli, V. Negnevitsky, K. Mehta, and J. P. Home. Encoding a qubit in a trapped-ion mechanical oscillator. *Nature*, 566(7745):513–517, Feb 2019.

[210] E. Flurin, L. S. Martin, S. Hacohen-Gourgy, and I. Siddiqi. Using a recurrent neural network to reconstruct quantum dynamics of a superconducting qubit from physical observations. *Phys. Rev. X*, 10:011006, Jan 2020.

[211] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt. Reinforcement learning with neural networks for quantum feedback. *Phys. Rev. X*, 8:031084, Sep 2018.

[212] A. G. Fowler, A. M. Stephens, and P. Groszkowski. High-threshold universal quantum computation on the surface code. *Phys. Rev. A*, 80(5), Nov 2009.

[213] Austin G. Fowler, David S. Wang, Charles D. Hill, Thaddeus D. Ladd, Rodney Van Meter, and Lloyd C. L. Hollenberg. Surface code quantum communication. *Phys. Rev. Lett.*, 104:180503, May 2010.

[214] M. Fréchet. Sur l'extension de certaines évaluations statistiques au cas de petits échantillons. *Rev. Inst. Int. Stat.*, pages 182–205, 1943.

[215] S. J. Freedman and J. F. Clauser. Experimental test of local hidden-variable theories. *Phys. Rev. Lett.*, 28:938–941, Apr 1972.

[216] F. Fröwis and W. Dür. Stable macroscopic quantum superpositions. *Phys. Rev. Lett.*, 106:110402, Mar 2011.

[217] F. Fröwis, P. Sekatski, W. Dür, N. Gisin, and N. Sangouard. Macroscopic quantum states: Measures, fragility, and implementations. *Rev. Mod. Phys.*, 90:025004, May 2018.

[218] V. Galitski, G. Juzeliūnas, and I. B. Spielman. Artificial gauge fields with ultracold atoms. *Physics Today*, 72(1):38–44, Jan 2019.

[219] A. Gangat, I.. McCulloch, and Y. Kao. Symmetry between repulsive and attractive interactions in driven-dissipative bose-hubbard systems. *Sci. Rep.*, 8(1), Feb 2018.

[220] M. Gebert, H. Küttler, and P. Müller. Anderson's orthogonality catastrophe. *Commun. Math. Phys.*, 329(3):979–998, Mar 2014.

[221] M. Gessner and A. Smerzi. Statistical speed of quantum states: Generalized quantum fisher information and schatten speed. *Phys. Rev. A*, 97:022109, Feb 2018.

[222] S. Gharibian. Strong np-hardness of the quantum separability problem. *Quantum Inf. Comput.*, 10:343–360, 2010.

[223] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum metrology. *Phys. Rev. Lett.*, 96:010401, Jan 2006.

[224] V. Giovannetti, S. Lloyd, and L. Maccone. Advances in quantum metrology. *Nat. photonics*, 5(4):222, 2011.

[225] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[226] C. P. Gonçalves. Quantum neural machine learning: Backpropagation and dynamics. *NeuroQuantology*, 15(1):22–41, 2017.

[227] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[228] V. Gorini, A. Kossakowski, and G. Sudarshan. Completely positive dynamical semigroups of n-level systems. *Journal of Mathematical Physics*, 17(5):821–825, 1976.

[229] D. Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, Caltech, 1997.

[230] D. Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57(1):127, 1998.

[231] D. Gottesman. Quantum error correction and fault-tolerance. *Quantum Information Processing: From Theory to Experiment*, 199:159, 2006.

[232] D. Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. In *Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics*, volume 68, pages 13–58, 2010.

[233] D. Gottesman, A. Yu. Kitaev, and John Preskill. Encoding a qubit in an oscillator. *Phys. Rev. A*, 64(1), Jun 2001.

[234] G. Gour, D. Jennings, F. Buscemi, R. Duan, and I. Marvian. Quantum majorization and a complete set of entropic conditions for quantum thermodynamics. *Nat. Commun.*, 9(1):1–9, 2018.

[235] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini. Hierarchical quantum classifiers. *npj Quantum Inf.*, 4(1):65, 2018.

[236] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009.

[237] D. M. Greenberger, M. A. Horne, and A. Zeilinger. Going Beyond Bell's Theorem. *arXiv:0712.0921*, December 2007.

[238] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE Trans. Neural. Netw. Learn. Syst.*, 28(10):2222–2232, 2017.

[239] A. Griffin. *High-momentum scattering and the condensate fraction*, page 67–91. Cambridge Studies in Low Temperature Physics. Cambridge University Press, 1993.

[240] Robert B. Griffiths. Graph states and graph codes. `http://quantum.phys.cmu.edu/QCQI/qitd452.pdf`, 2014. [Online; accessed 6-February-2020].

[241] C. Gross and I. Bloch. Quantum simulations with ultracold atoms in optical lattices. *Science*, 357(6355):995–1001, 2017.

[242] H. Grote, K. Danzmann, K. L. Dooley, R. Schnabel, J. Slutsky, and H. Vahlbruch. First long-term application of squeezed states of light in a gravitational-wave observatory. *Phys. Rev. Lett.*, 110:181101, May 2013.

[243] P. J. Grother. Nist special database 19. *Handprinted forms and characters database, National Institute of Standards and Technology*, page 10, 1995.

[244] L. Gurvits. Classical complexity and quantum entanglement. *Journal of Computer and System Sciences*, 69(3):448 – 484, 2004. Special Issue on STOC 2003.

[245] C. Gustiani and D. P. DiVincenzo. Blind oracular quantum computation. arXiv:2010.13840, 2020.

[246] S. Hacohen-Gourgy, V. V. Ramasesh, C. De Grandi, I. Siddiqi, and S. M. Girvin. Cooling and autonomous feedback in a bose-hubbard chain with attractive interactions. *Phys. Rev. Lett.*, 115:240501, Dec 2015.

[247] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951, 2000.

[248] R. H. R. Hahnloser, H. S. Seung, and J.-J. Slotine. Permitted and forbidden sets in symmetric threshold-linear networks. *Neural Comput.*, 15(3):621–638, 2003.

[249] P. J. Haley and D. Soloway. Extrapolation limitations of multilayer feedforward neural networks. In *IJCNN International Joint Conference on Neural Networks*, volume 4, pages 25–30 vol.4, 1992.

[250] K. Hammerer, A. S. Sørensen, and E. S. Polzik. Quantum interface between light and atomic ensembles. *Rev. Mod. Phys.*, 82:1041–1093, Apr 2010.

[251] Lisa Hänggli, Margret Heinze, and Robert König. Enhanced noise resilience of the surface–gottesman-kitaev-preskill code via designed bias. *Phys. Rev. A*, 102:052408, Nov 2020.

[252] B. Hanin and M. Sellke. Approximating continuous functions by relu nets of minimal width. arXiv:1710.11278, 2018.

[253] S. Hannig, L. Pelzer, N. Scharnhorst, J. Kramer, M. Stepanova, Z. T. Xu, N. Spethmann, I. D. Leroux, T. E. Mehlstäubler, and P. O. Schmidt. Towards a transportable aluminium ion quantum logic optical clock. *Rev. Sci. Instrum.*, 90(5):053204, 2019.

[254] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng. Deep speech: Scaling up end-to-end speech recognition. arXiv:1412.5567, 2014.

[255] A. W. Harrow, B. Recht, and I. L. Chuang. Efficient discrete approximations of quantum gates. *J. Math. Phys.*, 43(9):4445–4451, 2002.

[256] M. Hastings. Solving gapped hamiltonians locally. *Phys. Rev. B*, 73:085115, Feb 2006.

[257] M B Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08024, 2007.

[258] P. Hayden, M. Horodecki, and B. Terhal. The asymptotic entanglement cost of preparing a quantum state. *J. Phys. A*, 34(35):6891–6898, Aug 2001.

[259] S. Haykin. *Adaptive Filter Theory (rrd Ed.)*. Prentice-Hall, Inc., USA, 2013.

[260] P. He, M. A. Perlin, S. R. Muleady, R. J. Lewis-Swan, R. B. Hutson, J. Ye, and A. M. Rey. Engineering spin squeezing in a 3d optical lattice with interacting spin-orbit-coupled fermions. *Phys. Rev. Research*, 1:033075, Nov 2019.

[261] D. O. Hebb. *The organization of behavior: A neuropsychological theory.* Wiley & Sons, 1949.

[262] M. Hein, J. Eisert, and H. J. Briegel. Multiparty entanglement in graph states. *Phys. Rev. A*, 69:062311, Jun 2004.

[263] G. Heinze, C. Hubrich, and T. Halfmann. Stopped light and image storage by electromagnetically induced transparency up to the regime of one minute. *Phys. Rev. Lett.*, 111:033601, Jul 2013.

[264] G. Heinze, C. Hubrich, and T. Halfmann. Stopped light and image storage by electromagnetically induced transparency up to the regime of one minute. *Phys. Rev. Lett.*, 111:033601, Jul 2013.

[265] C. Helstrom. *Quantum detection and estimation theory*, volume 3. Academic press New York, 1976.

[266] C. W. Helstrom. Quantum detection and estimation theory. *J. Stat.l Phys.*, 1(2):231–252, 1969.

[267] K. Heshami, D. G. England, P. C. Humphreys, P. J. Bustard, V. M. Acosta, J. Nunn, and B. J. Sussman. Quantum memories: emerging applications and recent advances. *J. Mod. Opt.*, 63(20):2005–2028, Mar 2016.

[268] S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Diploma, Technische Universität München, 1991.

[269] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[270] S. Hochreiter and J. Schmidhuber. Lstm can solve hard long time lag problems. *Adv. Neural. Inform. Process. Syst.*, pages 473–479, 1997.

[271] S. G. Hofer and K. Hammerer. Entanglement-enhanced time-continuous quantum control in optomechanics. *Phys. Rev. A*, 91:033822, Mar 2015.

[272] S. G. Hofer and K. Hammerer. Chapter five - quantum control of optomechanical systems. volume 66 of *Advances In Atomic, Molecular, and Optical Physics*, pages 263–374. Academic Press, 2017.

[273] A. S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973.

[274] J. J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A*, 79(8):2554–2558, 1982.

[275] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Netw.*, 4(2):251–257, 1991.

[276] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Reviews of Modern Physics*, 81(2):865–942, Jun 2009.

[277] M. Hosseini, B. M. Sparkes, G. Campbell, P. K. Lam, and B. C. Buchler. High efficiency coherent optical memory with warm rubidium vapour. *Nat. Commun.*, 2(1):1–5, 2011.

[278] O. Hosten, N. J. Engelsen, R. Krishnakumar, and M. A. Kasevich. Measurement noise 100 times lower than the quantum-projection limit using entangled atoms. *Nature*, 529(7587):505–508, 2016.

[279] C. Howard. *An open systems approach to quantum optics.* Springer Verlag, 1991.

[280] D. Hümmer, P. Schneeweiss, A. Rauschenbeutel, and O. Romero-Isart. Heating in nanophotonic traps for cold atoms. *Phys. Rev. X*, 9:041034, Nov 2019.

[281] H. Hyötyniemi. Turing machines are recurrent neural networks. *Proceedings of STeP*, 96, 1996.

[282] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert. "event-ready-detectors" bell experiment via entanglement swapping. *Phys. Rev. Lett.*, 71:4287–4290, Dec 1993.

[283] D. Irving and F. Sorrentino. Synchronization of dynamical hypernetworks: Dimensionality reduction through simultaneous block-diagonalization of matrices. *Phys. Rev. E*, 86(5):056102, 2012.

[284] T. Itoko, R. Raymond, T. Imamichi, and A. Matsuo. Optimization of quantum circuit mapping using gate transformation and commutation. *Integration*, 70:43–50, 2020.

[285] A. G. Ivakhnenko. Heuristic self-organization in problems of engineering cybernetics. *Automatica*, 6(2):207–219, 1970.

[286] A. G. Ivakhnenko. Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(4):364–378, 1971.

[287] J. Jarkovsky, A. Molnar, N. Schuch, and J. Cirac. Efficient description of many-body systems with matrix product density operators. arxiv:2003.12418, 2020.

[288] D. Jaschke. *Many-body entangled dynamics of closed and open systems for quantum simulators.* PhD thesis, Colorado School of Mines, 2018.

[289] D. Jaschke and L. Carr. Open source matrix product states: exact diagonalization and other entanglement-accurate methods revisited in quantum systems. *J. Phys. A: Math. Theor.*, 51(46):465302, Oct 2018.

[290] D. Jaschke, S. Montangero, and L. Carr. One-dimensional many-body entangled open quantum systems with tensor network methods. *Quantum Sci. Technol.*, 4(1):013001, nov 2018.

[291] A. Jasinski, J. Montaner, R. C. Forrey, B. H. Yang, P. C. Stancil, N. Balakrishnan, J. Dai, R. A. Vargas-Hernández, and R. V. Krems. Machine learning corrected quantum dynamics calculations. *Phys. Rev. Research*, 2:032051, Aug 2020.

[292] D. Jennings, C. Brockt, J. Haegeman, T. J. Osborne, and F. Verstraete. Continuum tensor network field states, path integral representations and spatial symmetries. *New J. Phys.*, 17(6):063039, 2015.

[293] N. Jiang, Y.-F. Pu, W. Chang, C. Li, S. Zhang, and L.-M. Duan. Experimental realization of 105-qubit random access quantum memory. *npj Quantum Inf.*, 5(1):1–6, 2019.

[294] W. G. Jiang, G. Hagen, and T. Papenbrock. Extrapolation of nuclear structure observables with artificial neural networks. *Phys. Rev. C*, 100:054326, Nov 2019.

[295] M. I. Jordan. Serial order: a parallel distributed processing approach. *Technical report ICS-8604*, 5 1986.

[296] R. Jozsa. Fidelity for mixed quantum states. *J. Mod. Opt.*, 41(12):2315–2323, 1994.

[297] M. Junge and C. Palazuelos. Large violation of bell inequalities with low entanglement. *Commun. in Math. Phys.*, 306(3):695–746, 2011.

[298] H. Kamerlingh Onnes. Further experiments with liquid helium. c. on the change of electric resistance of pure metals at very low temperatures etc. iv. the resistance of pure mercury at helium temperatures. *KNAW Proc. B Phys. Sci.*, 13:1274–1276, January 1910.

[299] P. Kapitza. Viscosity of liquid helium below the λ-point. *Nature*, 141(3558):74–74, 1938.

[300] P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing.* Oxford University Press, Inc., New York, NY, USA, 2007.

[301] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, I.-C. Hoi, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, A. N. Cleland, and J. M. Martinis. State preservation by repetitive error detection in a superconducting quantum circuit. *Nature*, 519(7541):66–69, 2015.

[302] F. Kesting, F. Fröwis, and W. Dür. Effective noise channels for encoded quantum systems. *Phys. Rev. A*, 88:042305, Oct 2013.

[303] M. Keyl. Fundamentals of quantum information theory. *Physics reports*, 369(5):431–548, 2002.

[304] M. Keyl and R. F. Werner. Optimal cloning of pure states, testing single clones. *J. Math. Phys.*, 40(7):3283–3299, 1999.

[305] M. Keyl and R. F. Werner. The rate of optimal purification procedures. In *Annales Henri Poincaré*, volume 2, pages 1–26. Springer, 2001.

[306] A. A. Khajetoorians, D. Wegner, A. F. Otte, and I. Swart. Creating designer quantum states of matter atom-by-atom. *Nat. Rev. Phys.*, 1(12):703–715, Sep 2019.

[307] Maryam Khanahmadi and Klaus Mølmer. Time-dependent atomic magnetometry with a recurrent neural network. *Phys. Rev. A*, 103:032406, Mar 2021.

[308] P. Kidger and T. Lyons. Universal approximation with deep narrow networks. arXiv:1905.08539, 2020.

[309] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd. Continuous-variable quantum neural networks. arXiv:1806.06871, 2018.

[310] A. D. King, J. Carrasquilla, J. Raymond, I. Ozfidan, E. Andriyash, A. Berkley, M. Reis, T. Lanting, R. Harris, F. Altomare, P. Bunyk, J. Whittaker, L. Swenson, E. Hoskinson, Y. Sato, M. Volkmann, E. Ladizinsky, M. Johnson, J. Hilton, and M. H. Amin. Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature*, 560(7719):456–460, Aug 2018.

[311] A. D. King, J. Raymond, T. Lanting, S. V. Isakov, M. Mohseni, G. Poulin-Lamarre, S. Ejtemaee, W. Bernoudy, I. Ozfidan, A. Yu. Smirnov, and M. Reis, et al. (D-Wave). Scaling advantage over path-integral monte carlo in quantum simulation of geometrically frustrated magnets. *Nat. Commun.*, 12(1):1–6, 2021.

[312] S. A. King, L. J. Spiess, P. Micke, A. Wilzewski, T. Leopold, J. R. C. López-Urrutia, and P. O. Schmidt. Algorithmic ground-state cooling of weakly-coupled oscillators using quantum logic. arXiv:2102.12427, 2021.

[313] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv:9511026*, 1995.

[314] A. Yu. Kitaev. Quantum error correction with imperfect gates. In *Quantum Communication, Computing, and Measurement*, pages 181–188. Springer, 1997.

[315] A. Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Ann. Phys.*, 303(1):2–30, 2003.

[316] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver. Superconducting qubits: Current state of play. *Annu. Rev. Condens. Matter Phys.*, 11(1):369–395, 2020.

[317] M. Kliesch, T. Barthel, C. Gogolin, M. Kastoryano, and J. Eisert. Dissipative quantum church-turing theorem. *Phys. Rev. Lett.*, 107:120501, Sep 2011.

[318] M. Kliesch, D. Gross, and J. Eisert. Matrix-product operators and states: Np-hardness and undecidability. *Phys. Rev. Lett.*, 113:160503, Oct 2014.

[319] E. Knill and R. Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55(2):900, 1997.

[320] E. Knill, R. Laflamme, R. Martinez, and C. Negrevergne. Benchmarking quantum computers: the five-qubit error correcting code. *Phys. Rev. Lett.*, 86(25):5811, 2001.

[321] E. Knill, R. Laflamme, and W. H. Zurek. Resilient quantum computation. *Science*, 279(5349):342–345, 1998.

[322] P. A. Knott. A search algorithm for quantum state engineering and metrology. *New J. Phys.*, 18(7):073033, jul 2016.

[323] J. R. Kohn, J. P. Portes, M. P. Christenson, L. F. Abbott, and R. Behnia. State and stimulus dependence reconcile motion computation and the drosophila connectome. *bioRxiv*, 2021.

[324] R. König, B. Reichardt, and G. Vidal. Exact entanglement renormalization for string-net models. *Phys. Rev. B*, 79:195123, May 2009.

[325] R. König and V. Scholz. Matrix product approximations to multipoint functions in two-dimensional conformal field theory. *Phys. Rev. Lett.*, 117:121601, Sep 2016.

[326] R. König and V. Scholz. Matrix product approximations to conformal field theories. *Nuclear Physics B*, 920:32 – 121, 2017.

[327] B. Korzh, C. C. W. Lim, R. Houlmann, N. Gisin, M. J. Li, D. Nolan, B. Sanguinetti, R. Thew, and H. Zbinden. Provably secure and practical quantum key distribution over 307 km of optical fibre. *Nat. Photonics*, 9(3):163–168, 2015.

[328] A. Kossakowski. On quantum statistical mechanics of non-hamiltonian systems. *Reports on Mathematical Physics*, 3(4):247 – 274, 1972.

[329] G. Kotliar, S. Y. Savrasov, K. Haule, V. S. Oudovenko, O. Parcollet, and C. A. Marianetti. Electronic structure calculations with dynamical mean-field theory. *Rev. Mod. Phys.*, 78:865–951, Aug 2006.

[330] N. Kouda, N. Matsui, H. Nishimura, and F. Peper. Qubit neural network and its learning efficiency. *Neural Comput. Applic.*, 14(2):114–121, 2005.

[331] A. Kouzelis, K. Macieszczak, J. Minář, and I. Lesanovsky. Dissipative quantum state preparation and metastability in two-photon micromasers. *Phys. Rev. A*, 101:043847, Apr 2020.

[332] S. Krastanov and L. Jiang. Deep neural network probabilistic decoder for stabilizer codes. *Sci. Rep.*, 7(1):1–7, 2017.

[333] A. Kratsios and E. Bilokopytov. Non-euclidean universal approximation. arXiv:2006.02341, 2020.

[334] B. Kraus, H. P. Büchler, S. Diehl, A. Kantian, A. Micheli, and P. Zoller. Preparation of entangled states by quantum markov processes. *Phys. Rev. A*, 78:042307, Oct 2008.

[335] H. Krauter, C. A. Muschik, K. Jensen, W. Wasilewski, J. M. Petersen, J. I. Cirac, and E. S. Polzik. Entanglement generated by dissipation and steady state entanglement of two macroscopic objects. *Phys. Rev. Lett.*, 107:080503, Aug 2011.

[336] R. V. Krems. Bayesian machine learning for quantum molecular dynamics. *Phys. Chem. Chem. Phys.*, 21:13392–13410, 2019.

[337] D. Kretschmann and R. F. Werner. Quantum channels with memory. *Phys. Rev. A*, 72:062323, Dec 2005.

[338] M. Kritsotakis, J. A. Dunningham, and S. A. Haine. Spin squeezing of a bose-einstein condensate via quantum non-demolition measurement for quantum-enhanced atom interferometry. arXiv:2005.00299, 2020.

[339] I. Kruse, K. Lange, J. Peise, B. Lücke, L. Pezzè, J. Arlt, W. Ertmer, C. Lisdat, L. Santos, A. Smerzi, and C. Klempt. Improvement of an atomic clock using squeezed vacuum. *Phys. Rev. Lett.*, 117:143004, Sep 2016.

[340] T. Kucherenko, J. Beskow, and H. Kjellström. A neural network approach to missing marker reconstruction in human motion capture. arXiv:1803.02665, 2018.

[341] T. Kucherenko, D. Hasegawa, G. E. Henter, N. Kaneko, and H. Kjellström. Analyzing input and output representations for speech-driven gesture generation. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, IVA '19, page 97–104, New York, NY, USA, 2019. Association for Computing Machinery.

[342] T. Kucherenko, D. Hasegawa, N. Kaneko, G. E. Henter, and H. Kjellström. On the importance of representations for speech-driven gesture generation. AAMAS '19, page 2072–2074, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems.

[343] T.. Kucherenko, D. Hasegawa, N. Kaneko, G. E. Henter, and Hedvig Kjellström. Moving fast and slow: Analysis of representations and post-processing in speech-driven automatic gesture generation. *Int. J. Hum.-Comput. Int.*, 0(0):1–17, 2021.

[344] P. Kumar, K. Snizhko, and Y. Gefen. Engineering mixed states with weak measurements. arXiv:2006.08190, 2020.

[345] P Kumar, H.-G. Zirnstein, K. Snizhko, Y. Gefen, and B. Rosenow. Optimized quantum steering and exceptional points. arXiv:2101.07284, 2021.

[346] G. Kurizki, P. Bertet, Y. Kubo, K. Mølmer, D. Petrosyan, P. Rabl, and J. Schmiedmayer. Quantum technologies with hybrid systems. *PNAS*, 112(13):3866–3873, 2015.

[347] O. Kyriienko. Quantum inverse iteration algorithm for programmable quantum simulators. *npj Quantum Information*, 6(1), Jan 2020.

[348] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek. Perfect quantum error correcting code. *Phys. Rev. Lett.*, 77(1):198, 1996.

[349] N.s Laflorencie, E. S. Sørensen, M. S. Chang, and I. Affleck. Boundary effects in the critical scaling of entanglement entropy in 1d systems. *Phys. Rev. Lett.*, 96(10), Mar 2006.

[350] L. Lamata, U. Alvarez-Rodriguez, J. D. Martín-Guerrero, M. Sanz, and E. Solano. Quantum autoencoders via quantum adders with genetic algorithms. *Quantum Sci. Technol.*, 4(1):014007, 2018.

[351] F. Lange, Z. Lenarčič, and A. Rosch. Pumping approximately integrable systems. *Nat. Commun.*, 8(1), Jun 2017.

[352] M. V. Larsen, C. Chamberland, K. Noh, J. S. Neergaard-Nielsen, and U. L. Andersen. A fault-tolerant continuous-variable measurement-based quantum computation architecture. arXiv:2101.03014, 2021.

[353] N. Lashkari, D. Stanford, M. Hastings, T. Osborne, and P. Hayden. Towards the fast scrambling conjecture. *J. High Energy Phys.*, 2013(4):22, Apr 2013.

[354] Y. LeCun. *Modeles connexionnistes de l'apprentissage (connectionist learning models)*. PhD thesis, Université P. et M. Curie (Paris 6), June 1987.

[355] W. Lee, M. Kim, H. Jo, Y. Song, and J. Ahn. Coherent and dissipative dynamics of entangled few-body systems of rydberg atoms. *Phys. Rev. A*, 99:043404, Apr 2019.

[356] E. L. Lehmann and G. Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

[357] I. D. Leroux, M. H. Schleier-Smith, and V. Vuletić. Orientation-dependent entanglement lifetime in a squeezed atomic clock. *Phys. Rev. Lett.*, 104:250801, Jun 2010.

[358] M. Lewenstein. Quantum Perceptrons. *J. Mod. Opt.*, 41(12):2491–2501, December 1994.

[359] R. Y. Li, R. Di Felice, R. Rohs, and D. A. Lidar. Quantum annealing versus classical machine learning applied to a simplified computational biology problem. *npj Quantum Inf.*, 4(1):1–10, 2018.

[360] Y. Li, S. Li, D. Caballero, M. Miyasaka, A. Lewis, and B. Hannaford. Improving control precision and motion adaptiveness for surgical robot with recurrent neural network. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3538–3543, 2017.

[361] Z.-D. Li, R. Zhang, X.-F. Yin, L.-Z. Liu, Y. Hu, Y.-Q. Fang, Y.-Y. Fei, X. Jiang, J. Zhang, L. Li, N.-L. Liu, F. Xu, Y.-A. Chen, and J.-W. Pan. Experimental quantum repeater without quantum memory. *Nat. Photonics*, 13(9):644–648, 2019.

[362] S.-K. Liao, W.-Q. Cai, J. Handsteiner, B. Liu, J. Yin, L. Zhang, D. Rauch, M. Fink, J.-G. Ren, W.-Y. Liu, Y. Li, Q. Shen, Y. Cao, F.-Z. Li, J.-F. Wang, Y.-M. Huang, L. Deng, T. Xi, L. Ma, T. Hu, L. Li, N.-L. Liu, F. Koidl, P. Wang, Y.-A. Chen, X.-B. Wang, M. Steindorfer, G. Kirchner, C.-Y. Lu, R. Shu, R. Ursin, T. Scheidl, C.-Z. Peng, J.-Y. Wang, A. Zeilinger, and J.-W. Pan. Satellite-relayed intercontinental quantum network. *Phys. Rev. Lett.*, 120:030501, Jan 2018.

[363] J. L. Lin and J. P. Wolfe. Bose-einstein condensation of paraexcitons in stressed cu_2o. *Phys. Rev. Lett.*, 71:1222–1225, Aug 1993.

[364] G. Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, Jun 1976.

[365] A. Ling, M. Peloso, I. Marcikic, and C. Lamas-Linares, A.and Kurtsiefer. Experimental e91 quantum key distribution. *Integrated Optoelectronic Devices 2008*, pages 69030U–69030U, 2008.

[366] Y. Liu, S. Arunachalam, and K. Temme. A rigorous and robust quantum speed-up in supervised machine learning. arXiv:2010.02174, 2020.

[367] Y.-H. Liu and D. Poulin. Neural belief-propagation decoders for quantum error-correcting codes. *Phys. Rev. Lett.*, 122(20):200501, 2019.

[368] S. Lloyd. Almost any quantum logic gate is universal. *Phys. Rev. Lett.*, 75(2):346, 1995.

[369] M. Lo, H.-K.and Curty and K. Tamaki. Secure quantum key distribution. *Nat. Photonics*, 8(8):595–604, 2014.

[370] A. López-Incera, P. Sekatski, and W. Dür. All macroscopic quantum states are fragile and hard to prepare. *Quantum*, 3:118, January 2019.

[371] M. Lotem, A. Weichselbaum, J. von Delft, and M. Goldstein. Renormalized lindblad driving: A numerically exact nonequilibrium quantum impurity solver. *Phys. Rev. Research*, 2(4), Oct 2020.

[372] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. arXiv:1709.02540, 2017.

[373] B. Lücke, J. Peise, G. Vitagliano, J. Arlt, L. Santos, G. Tóth, and C. Klempt. Detecting multiparticle entanglement of dicke states. *Phys. Rev. Lett.*, 112:155304, Apr 2014.

[374] B. Lücke, M. Scherer, J. Kruse, L. Pezzé, F. Deuretzbacher, P. Hyllus, O. Topic, J. Peise, W. Ertmer, J. Arlt, L. Santos, A. Smerzi, and C. Klempt. Twin matter waves for interferometry beyond the classical limit. *Science*, 334(6057):773–776, 2011.

[375] H. P. Lüschen, S. Scherg, T. Kohlert, M. Schreiber, P. Bordia, X. Li, S. Das Sarma, and I. Bloch. Single-particle mobility edge in a one-dimensional quasiperiodic optical lattice. *Phys. Rev. Lett.*, 120:160404, Apr 2018.

[376] E. Lustig, S. Weimann, Y. Plotnik, Y. Lumer, M. A. Bandres, A. Szameit, and M. Segev. Photonic topological insulator in synthetic dimensions. *Nature*, 567(7748):356–360, 2019.

[377] D. J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[378] T. Maehara and K. Murota. Error-controlling algorithm for simultaneous block-diagonalization and its application to independent component analysis. *JSIAM Letters*, 2:131–134, 2010.

[379] T. Maehara and K. Murota. A numerical algorithm for block-diagonal decomposition of matrix *-algebras with general irreducible components. *Japan J. Indust. Appl. Math.*, 27(2):263–293, 2010.

[380] T. Maehara and K. Murota. Simultaneous singular value decomposition. *Linear Algebra and its Applications*, 435(1):106 – 116, 2011.

[381] S. Mangini, F. Tacchino, D. Gerace, D. Bajoni, and C. Macchiavello. Quantum computing models for artificial neural networks. arXiv:2102.03879, 2021.

[382] Y. Manin. Computable and uncomputable. *Sovetskoye Radio, Moscow*, 128, 1980.

[383] F. Martínez-García, D. Vodola, and M. Müller. Adaptive bayesian phase estimation for quantum error correcting codes. *New J. Phys.*, 21(12):123027, dec 2019.

[384] G. Martius and C. H. Lampert. Extrapolation and learning equations. arXiv:1610.02995, 2016.

[385] N. Maskara, A. Kubica, and T. Jochym-O'Connor. Advantages of versatile neural-network decoding for topological codes. *Phys. Rev. A*, 99(5):052351, 2019.

[386] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan. Quantum computational chemistry. *Rev. Mod. Phys.*, 92:015003, Mar 2020.

[387] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, 5(1), Sep 2019.

[388] K. C. McCormick, J. Keller, S. C. Burd, D. J. Wineland, A. C. Wilson, and D. Leibfried. Quantum-enhanced sensing of a single-ion mechanical oscillator. *Nature*, 572(7767):86–90, 2019.

[389] W. F. McGrew, X. Zhang, R. J. Fasano, S. A. Schäffer, K. Beloy, D. Nicolodi, R. C. Brown, N. Hinkley, G. Milani, M. Schioppo, T. H. Yoon, and A. D. A. D. Ludlow. Atomic clock performance enabling geodesy below the centimetre level. *Nature*, 564(7734):87–90, 2018.

[390] T. E. Mehlstäubler, G. Grosche, C. Lisdat, P. O. Schmidt, and H. Denker. Atomic clocks for geodesy. *Rep. Prog. Phys.*, 81(6):064401, 2018.

[391] A. A. Melnikov, H. P. Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel. Active learning machine learns to create new quantum experiments. *PNAS*, 115(6):1221–1226, 2018.

[392] A. A. Melnikov, P. Sekatski, and N. Sangouard. Setting up experimental bell tests with reinforcement learning. *Phys. Rev. Lett.*, 125:160401, Oct 2020.

[393] M. Metcalf, J. E. Moussa, W. A. de Jong, and M. Sarovar. Engineered thermalization and cooling of quantum many-body systems. *Phys. Rev. Research*, 2:023214, May 2020.

[394] M. Metcalf, E. Stone, K. Klymko, A. F. Kemper, M. Sarovar, and W. A. de Jong. Quantum markov chain monte carlo with digital dissipative dynamics on quantum computers. arXiv:2103.03207, 2021.

[395] A. Micheli, G. K. Brennen, and P. Zoller. A toolbox for lattice-spin models with polar molecules. *Nature Physics*, 2(5):341–347, Apr 2006.

[396] M. Minsky and S. A. Papert. *Perceptrons: An introduction to computational geometry.* MIT press, 2017.

[397] K. Mølmer, Y. Castin, and J. Dalibard. Monte carlo wave-function method in quantum optics. *JOSA B*, 10(3):524–538, 1993.

[398] A. Molnar, N. Schuch, F. Verstraete, and I. Cirac. Approximating gibbs states of local hamiltonians efficiently with projected entangled pair states. *Phys. Rev. B*, 91:045138, Jan 2015.

[399] A. Monràs, G. Sentís, and P. Wittek. Inductive supervised quantum learning. *Phys. Rev. Lett.*, 118:190503, May 2017.

[400] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg. Generation and verification of 27-qubit greenberger-horne-zeilinger states in a superconducting quantum computer. arXiv:2101.08946, 2021.

[401] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg. Whole-device entanglement in a 65-qubit superconducting quantum computer, 2021.

[402] J. H. Moore, C. C. Davis, M. A. Coplan, and S. C. Greer. *Building scientific apparatus.* Cambridge University Press, 2009.

[403] Alex Mott, J. Job, J.-R. Vlimant, D. Lidar, and M. Spiropulu. Solving a higgs optimization problem with quantum annealing for machine learning. *Nature*, 550(7676):375–379, 2017.

[404] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão, and G. K.-L. Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nat. Phys.*, 16(2):205–210, Nov 2019.

[405] R. Movassagh, E. Farhi, J. Goldstone, D. Nagaj, T. Osborne, and P. Shor. Unfrustrated qudit chains and their ground states. *Phys. Rev. A*, 82:012318, Jul 2010.

[406] J. T. Muhonen, J. P. Dehollain, A. Laucht, F. E. Hudson, R. Kalra, T. Sekiguchi, K. M. Itoh, D. N. Jamieson, J. C. McCallum, A. S. Dzurak, and A. Morello. Storing quantum information for 30 seconds in a nanoelectronic device. *Nat. Nanotechnol.*, 9(12):986–991, 2014.

[407] M. Müller. *Many-Body Quantum Simulation with Rydberg Atoms and Ions.* PhD thesis, Leopold-Franzens-Universität Innsbruck, 2011.

[408] M. Müller, S. Diehl, G. Pupillo, and P. Zoller. Engineered open systems and quantum simulations with atoms and ions. In Paul Berman, Ennio Arimondo, and Chun Lin, editors, *Advances in Atomic, Molecular, and Optical Physics*, volume 61 of *Advances In Atomic, Molecular, and Optical Physics*, pages 1–80. Academic Press, 2012.

[409] M. Müller, A. Rivas, E. A. Martínez, D. Nigg, P. Schindler, T. Monz, R. Blatt, and M. A. Martin-Delgado. Iterative phase optimization of elementary quantum error correcting codes. *Phys. Rev. X*, 6:031030, Aug 2016.

[410] M. Müller-Lennert, F. Dupuis, O. Szehr, S. Fehr, and M. Tomamichel. On quantum rényi entropies: A new generalization and some properties. *Journal of Mathematical Physics*, 54(12):122203, 2013.

[411] W. J. Munro, A. M. Stephens, S. J. Devitt, K. A. Harrison, and K. Nemoto. Quantum communication without the necessity of quantum memories. *Nat. Photonics*, 6(11):777–781, 2012.

[412] S. Muralidharan, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang. Ultrafast and fault-tolerant quantum communication across long distances. *Phys. Rev. Lett.*, 112:250501, Jun 2014.

[413] K. Murota, Y. Kanno, M. Kojima, and S. Kojima. A numerical algorithm for block-diagonal decomposition of matrix *-algebras with application to semidefinite programming. *Japan J. Indust. Appl. Math.*, 27(1):125–160, Jun 2010.

[414] G. Murta, S. B. van Dam, J. Ribeiro, R. Hanson, and S. Wehner. Towards a realization of device-independent quantum key distribution. *Quantum Sci. Technol.*, 4(3):035011, jul 2019.

[415] B. Nachtergaele. The spectral gap for some spin chains with discrete symmetry breaking. *Commun. Math. Phys.*, 175(3):565–606, Feb 1996.

[416] H. Nakazato, M. Unoki, and K. Yuasa. Preparation and entanglement purification of qubits through zeno-like measurements. *Phys. Rev. A*, 70:012303, Jul 2004.

[417] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis. Optimizing quantum error correction codes with reinforcement learning. *Quantum*, 3:215, 2019.

[418] G. A. Negoita, J. P. Vary, G. R. Luecke, P. Maris, A. M. Shirokov, I. J. Shin, Y. Kim, E. G. Ng, C. Yang, M. Lockner, and G. M. Prabhu. Deep learning: Extrapolation tool for ab initio nuclear theory. *Phys. Rev. C*, 99:054308, May 2019.

[419] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. In *Doklady A.N. SSSR*, volume 269, pages 543–547, 1983.

[420] A. Neven, D. Gunn, M. Hebenstreit, and B. Kraus. Local transformations of multiple multipartite states. arXiv:2007.06256, 2020.

[421] R. Nichols, L. Mineh, J. Rubio, J. C. F. Matthews, and P. A. Knott. Designing quantum experiments with a genetic algorithm. *Quantum Sci. Technol.*, 4(4):045012, Oct 2019.

[422] M. Nielsen. Conditions for a class of entanglement transformations. *Phys. Rev. Lett.*, 83:436–439, Jul 1999.

[423] M. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[424] M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.

[425] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt. Quantum computations on a topologically encoded qubit. *Science*, 345(6194):302–305, 2014.

[426] M. Nonnenmacher, D. Reeb, and I. Steinwart. Which minimizer does my neural network converge to? arXiv:2011.02408, 2020.

[427] M. Nonnenmacher, D. Reeb, and I. Steinwart. Wide neural networks are interpolating kernel methods: Impact of initialization on generalization, 2020.

[428] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Ye. Liu, L. Frunzio, S. M. Girvin, L. Jiang, M. Mirrahimi, M. H. Devoret, and R. J. Schoelkopf. Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature*, 536(7617):441–445, 2016.

[429] P. Ondruska and I. Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[430] T. P. Orlando, J. E. Mooij, L. Tian, C. H. van der Wal, L. S. Levitov, S. Lloyd, and J. J. Mazo. Superconducting persistent-current qubit. *Phys. Rev. B*, 60:15398–15413, Dec 1999.

[431] R. Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, Oct 2014.

[432] T. Osborne, J. Eisert, and F. Verstraete. Holographic quantum states. *Phys. Rev. Lett.*, 105:260401, Dec 2010.

[433] T. Ozawa and H. M. Price. Topological quantum matter in synthetic dimensions. *Nat. Rev. Phys.*, 1(5):349–357, 2019.

[434] R. O'Donnell and Z. Jiang. Lecture notes for cmu 18-859bb: Quantum computation. `https://www.cs.cmu.edu/~odonnell/quantum15/lecture17.pdf`, Nov 2015. [Online; accessed 23-March-2021].

[435] A. K. Pal, P. Schindler, A. Erhard, A. Rivas, M.-A. Martin-Delgado, R. RBlatt, T. Monz, and M. Müller. Relaxation times do not capture logical qubit dynamics. arXiv:2012.07911, 2020.

[436] F. Pan and P. Zhang. Simulating the sycamore quantum supremacy circuits, 2021.

[437] J.-W. Pan, D. Bouwmeester, M. Daniell, H. Weinfurter, and A. Zeilinger. Experimental test of quantum nonlocality in three-photon greenberger–horne–zeilinger entanglement. *Nature*, 403(6769):515–519, 2000.

[438] J.-W. Pan, D. Bouwmeester, H. Weinfurter, and A. Zeilinger. Experimental entanglement swapping: Entangling photons that never interacted. *Phys. Rev. Lett.*, 80:3891–3894, May 1998.

[439] M. Pant, H. Krovi, D. Englund, and S. Guha. Rate-distance tradeoff and resource costs for all-optical quantum repeaters. *Phys. Rev. A*, 95:012304, Jan 2017.

[440] J. L. Park. The concept of transition in quantum mechanics. *Found. Phys.*, 1(1):23–33, 1970.

[441] F. Pastawski, L. Clemente, and J. I. Cirac. Quantum memories based on engineered dissipation. *Phys. Rev. A*, 83:012304, Jan 2011.

[442] F. Pastawski, B. Yoshida, D. Harlow, and J. Preskill. Holographic quantum error-correcting codes: toy models for the bulk/boundary correspondence. *Journal of High Energy Physics*, 2015(6):149, Jun 2015.

[443] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, T.. Magerlein, E. Solomonik, and R. Wisnieff. Breaking the 49-qubit barrier in the simulation of quantum circuits. *arXiv:1710.05867*, 2017.

[444] E. Pednault, J. A. Gunnels, G Nannicini, L. Horesh, and R Wisnieff. Leveraging secondary storage to simulate deep 54-qubit sycamore circuits. arXiv:1910.09534, 2019.

[445] E. Pedrozo-Peñafiel, S. Colombo, C. Shu, A. F. Adiyatullin, Z. Li, E. Mendez, B. Braverman, A. Kawasaki, D. Akamatsu, Y. Xiao, and V. Vuletić. Entanglement-enhanced optical atomic clock. arXiv:2006.07501, 2020.

[446] A. Pepper, N. Tischler, and G. J. Pryde. Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning. *Phys. Rev. Lett.*, 122(6):060501, 2019.

[447] D. Perekrestenko, P. Grohs, D. Elbrächter, and H. Bölcskei. The universal approximation power of finite-width deep relu networks. arXiv:1806.01528, 2018.

[448] D. Perekrestenko, S. Müller, and H. Bölcskei. Constructive universal high-dimensional distribution generation through deep relu networks. In *International Conference on Machine Learning*, pages 7610–7619. PMLR, 2020.

[449] A. Peres. Reversible logic and quantum computers. *Phys. Rev. A*, 32(6):3266, 1985.

[450] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Inf. Comput.*, 7(5):401–430, 2007.

[451] L. Pezzè and A. Smerzi. Quantum theory of phase estimation. arXiv:1411.5164, 2014.

[452] L. Pezzè, A. Smerzi, M. K. Oberthaler, R. Schmied, and P. Treutlein. Quantum metrology with nonclassical states of atomic ensembles. *Rev. Mod. Phys.*, 90:035005, Sep 2018.

[453] P. Pfeuty. The one-dimensional ising model with a transverse field. *Ann. Phys.*, 57(1):79–90, 1970.

[454] A. C. Pflanzer, O. Romero-Isart, and J. I. Cirac. Optomechanics assisted by a qubit: From dissipative state preparation to many-partite systems. *Phys. Rev. A*, 88:033804, Sep 2013.

[455] H. Pichler and P. Zoller. Photonic circuits with time delays and quantum feedback. *Phys. Rev. Lett.*, 116:093601, Mar 2016.

[456] A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, January 1999.

[457] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. Pereira, M. Razavi, J. S. Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden. Advances in quantum cryptography. arXiv:1906.01645, Jun 2019.

[458] T. B. Pittman, B. C. Jacobs, and J. D. Franson. Demonstration of quantum error correction using linear optics. *Phys. Rev. A*, 71(5):052332, 2005.

[459] I. Pogorelov, T. Feldker, C. D. Marciniak, G. Jacob, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, K. Lakhmanskiy, R. Blatt, P. Schindler, and T. Monz. A compact ion-trap quantum computing demonstrator. arXiv:2101.11390, 2021.

[460] K. Poland, K. Beer, and T. J. Osborne. No free lunch for quantum machine learning. arXiv:2003.14103, 2020.

[461] S. Polla, Y. Herasymenko, and T. E. O'Brien. Quantum digital cooling. arXiv:1909.10538, 2019.

[462] J. B. Pollack. Backpropagation is sensitive to initial conditions. *Complex Syst.*, 4:269–280, 1990.

[463] S. Popescu and D. Rohrlich. Thermodynamics and the measure of entanglement. *Phys. Rev. A*, 56(5):R3319–R3321, Nov 1997.

[464] C. Portmann and R. Renner. Security in quantum cryptography. arXiv:2102.00021, 2021.

[465] I. Potamitis. Deep learning for detection of bird vocalisations. arXiv:1609.08408, 2016.

[466] J. Preskill. Fault-tolerant quantum computation. In *Introduction to quantum computation and information*, pages 213–269. World Scientific, 1998.

[467] J. Preskill. Quantum computing and the entanglement frontier. *arXiv:1203.5813*, 2012.

[468] J. Preskill. Lecture notes for ph219/cs219: quantum information. `http://www.theory.caltech.edu/people/preskill/ph219/chap3_15.pdf`, 2018. [Online; accessed 6-February-2020].

[469] J. Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[470] H. M. Price, O. Zilberberg, T. Ozawa, I. Carusotto, and N. Goldman. Four-dimensional quantum hall effect with ultracold atoms. *Phys. Rev. Lett.*, 115(19):195303, 2015.

[471] Y. F. Pu, N. Jiang, W. Chang, H. X. Yang, C. Li, and L. M. Duan. Experimental realization of a multiplexed quantum memory with 225 individually accessible memory cells. *Nat. Commun.*, 8(1):1–6, 2017.

[472] C. J. Pugh, S. Kaiser, J.-P. Bourgoin, J. Jin, N. Sultana, S. Agne, E. Anisimova, V. Makarov, E. Choi, B. L. Higgins, and T. Jennewein. Airborne demonstration of a quantum key distribution receiver payload. *Quantum Sci. Technol.*, 2(2):024009, Jun 2017.

[473] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Netw.*, 12(1):145 – 151, 1999.

[474] Y. Quek, S. Fort, and H. K. Ng. Adaptive quantum state tomography with neural networks. arXiv:1812.06693, 2018.

[475] F. Quijandría, D. Porras, J. J.e García-Ripoll, and D. Zueco. Circuit qed bright source for chiral entangled light based on dissipation. *Phys. Rev. Lett.*, 111:073602, Aug 2013.

[476] M. Raghunandan. *Quantum technological applications using dissipative many-body dynamics*. PhD thesis, Hannover: Institutionelles Repositorium der Leibniz Universität Hannover, 2020.

[477] M. Raghunandan, F. Wolf, C. Ospelkaus, P. O. Schmidt, and H. Weimer. Initialization of quantum simulators by sympathetic cooling. *Science Advances*, 6(10), 2020.

[478] M. Raghunandan, J. Wrachtrup, and H. Weimer. High-density quantum sensing with dissipative first order transitions. *Phys. Rev. Lett.*, 120:150501, Apr 2018.

[479] C. R. Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.*, 37:81–93, 1945.

[480] E. Räsänen, A. Castro, J. Werschnik, A. Rubio, and E. K. U. Gross. Optimal control of quantum rings by terahertz laser pulses. *Phys. Rev. Lett.*, 98:157404, Apr 2007.

[481] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001.

[482] R. Raussendorf, D. E. Browne, and H. J. Briegel. Measurement-based quantum computation on cluster states. *Phys. Rev. A*, 68:022312, Aug 2003.

[483] R. Raussendorf, J. Harrington, and K. Goyal. Topological fault-tolerance in cluster state quantum computation. *New J. Phys.*, 9(6):199–199, jun 2007.

[484] R. Raussendorf, C. Okay, D.-S. Wang, D. T. Stephen, and H. P. Nautrup. Computationally universal phase of quantum matter. *Phys. Rev. Lett.*, 122(9), Mar 2019.

[485] W. Reagor, M.and Pfaff, C. Axline, R. W. Heeres, N. Ofek, E. Sliwa, Ka.and Holland, C. Wang, J. Blumoff, K. Chou, M. J. Hatridge, M. H. Frunzio, L.and Devoret, L. Jiang, and R. J. Schoelkopf. Quantum memory with millisecond coherence in circuit qed. *Phys. Rev. B*, 94:014506, Jul 2016.

[486] P. Rebentrost, T. R. Bromley, C. Weedbrook, and S. Lloyd. Quantum hopfield neural network. *Phys. Rev. A*, 98:042308, Oct 2018.

[487] M. D. Reed, L. DiCarlo, S. E. Nigg, L. Sun, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf. Realization of three-qubit quantum error correction with superconducting circuits. *Nature*, 482(7385):382–385, 2012.

[488] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer. Elucidating reaction mechanisms on quantum computers. *Proc. Natl. Acad. Sci. U. S. A.*, 114(29):7555–7560, 2017.

[489] M. Reimpell and R. F. Werner. Iterative optimization of quantum error correcting codes. *Phys. Rev. Lett.*, 94(8), Mar 2005.

[490] F. Reiter, F. Lange, S. Jain, M. Grau, J. P. Home, and Z. Zala Lenarčič. Engineering generalized gibbs ensembles with trapped ions. arXiv:1910.01593, 2020.

[491] F. Reiter, D. Reeb, and A. S. Sørensen. Scalable dissipative preparation of many-body entanglement. *Phys. Rev. Lett.*, 117:040501, Jul 2016.

[492] J.-G. Ren, P. Xu, H.-L. Yong, L. Zhang, S.-K. Liao, J. Yin, W.-Y. Liu, W.-Q. Cai, M. Yang, L. Li, K.-X. Yang, X. Han, Y.-Q. Yao, J. Li, H.-Y. Wu, S. Wan, L. Liu, D.-Q. Liu, Y.-W. Kuang, Z.-P. He, P. Shang, C. Guo, R.-H. Zheng, K. Tian, Z.-C. Zhu, N.-L. Liu, C.-Y. Lu, R. Shu, Y.-A. Chen, C.-Z. Peng, J.-Y. Wang, and J.-W. Pan. Ground-to-satellite quantum teleportation. *Nature*, 549(7670):70–73, Aug 2017.

[493] M.-O. Renou, D. Trillo, M. Weilenmann, L. P. Thinh, A. Tavakoli, N. Gisin, A. Acín, and M. Navascues. Quantum physics needs complex numbers. arXiv:2101.10873, 2021.

[494] A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[495] D. Riste, S. Poletto, M.-Z. Huang, A. Bruno, V. Vesterinen, O.-P. Saira, and L. DiCarlo. Detecting bit-flip errors in a logical qubit using stabilizer measurements. *Nat. Commun.*, 6(1):1–6, 2015.

[496] A. Rivas and S. F. Huelga. *Open quantum systems*. Springer, 2012.

[497] C. Roberts, A. Milsted, M. Ganahl, A. Zalcman, B. Fontaine, Y. Zou, J. Hidary, G. Vidal, and S. Leichenauer. Tensornetwork: A library for physics and machine learning. arXiv:1905.01330, 2019.

[498] J. Romero, J. P. Olson, and A. Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol.*, 2(4):045001, 2017.

[499] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408, 1958.

[500] W. Rosenfeld, D. Burchardt, R. Garthoff, K. Redeker, N. Ortegel, M. Rau, and H. Weinfurter. Event-ready bell test using entangled atoms simultaneously closing detection and locality loopholes. *Phys. Rev. Lett.*, 119:010402, Jul 2017.

[501] C. Roser. *Faster, better, cheaper in the history of manufacturing: from the stone age to lean manufacturing and beyond.* CRC Press, 2016.

[502] P. Rotondo, M. Marcuzzi, J. P. Garrahan, I. Lesanovsky, and M. Müller. Open quantum generalisation of hopfield neural networks. *J. Phys. A Math.*, 51(11):115301, 2018.

[503] Pietro Rotondo, Marco Cosentino Lagomarsino, and Giovanni Viola. Dicke simulators with emergent collective quantum computational abilities. *Phys. Rev. Lett.*, 114:143601, Apr 2015.

[504] A. Roura, C. Schubert, D. Schlippert, and E. M. Rasel. Measuring gravitational time dilation with delocalized quantum superpositions. arXiv:2010.11156, 2020.

[505] S. Ruder. An overview of gradient descent optimization algorithms. arXiv:1609.04747, 2016.

[506] M. S. Ruiz. *Tensor networks in condensed matter.* PhD thesis, Technische Universität München, 2011.

[507] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[508] L. Sá, P. Ribeiro, and T. Prosen. Integrable non-unitary open quantum circuits. arXiv:2011.06565, 2020.

[509] S. Saito, X. Zhu, R. Amsüss, Y. Matsuzaki, K. Kakuyanagi, T. Shimo-Oka, N. Mizuochi, K. Nemoto, W. J. Munro, and K. Semba. Towards realizing a quantum memory for a super-conducting qubit: Storage and retrieval of quantum states. *Phys. Rev. Lett.*, 111:107008, Sep 2013.

[510] Leonardo Salvi, Nicola Poli, Vladan Vuletić, and Guglielmo M. Tino. Squeezing on momentum states for atom interferometry. *Phys. Rev. Lett.*, 120:033601, Jan 2018.

[511] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin. Quantum repeaters based on atomic ensembles and linear optics. *Rev. Mod. Phys.*, 83:33–80, Mar 2011.

[512] M. Sanz, D. Perez-Garcia, M. Wolf, and I. Cirac. A quantum version of wielandt's inequality. *IEEE Trans. Inf. Theory*, 56(9):4668–4673, Sept 2010.

[513] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka, and et al. Field test of quantum key distribution in the tokyo qkd network. *Opt. Express*, 19(11):10387, May 2011.

[514] D. Sauerwein, N. Wallach, G. Gour, and B. Kraus. Transformations among pure multipartite entangled states via local operations are almost never possible. *Phys. Rev. X*, 8:031020, Jul 2018.

[515] P. R. Saulson. Thermal noise in mechanical experiments. *Phys. Rev. D*, 42:2437–2445, Oct 1990.

[516] D. Scheiermann. Noise robustness of quantum neural networks. Bachelor's thesis, Leibniz Universität Hannover, Nov 2019.

[517] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, and R. Blatt. Experimental repetitive quantum error correction. *Science*, 332(6033):1059–1061, 2011.

[518] P. Schindler, M. Müller, D. Nigg, J. T. Barreiro, E. A. Martinez, M. Hennrich, T. Monz, S. Diehl, P. Zoller, and R. Blatt. Quantum simulation of dynamical maps with trapped ions. *Nat. Phys.*, 9(6):361–367, May 2013.

[519] R. Schirhagl, K. Chang, M. Loretz, and C. L. Degen. Nitrogen-vacancy centers in diamond: Nanoscale sensors for physics and biology. *Annu. Rev. Phys. Chem.*, 65(1):83–105, 2014. PMID: 24274702.

[520] D. Schlingemann and R. F. Werner. Quantum error-correcting codes associated with graphs. *Phys. Rev. A*, 65:012308, Dec 2001.

[521] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Netw.*, 61:85–117, Jan 2015.

[522] P. O. Schmidt, T. Rosenband, C. Langer, W. M. Itano, J. C. Bergquist, and D. J. Wineland. Spectroscopy using quantum logic. *Science*, 309(5735):749–752, 2005.

[523] R. Schmied, J.-D. Bancal, B. Allard, M. Fadel, V. Scarani, P. Treutlein, and N. Sangouard. Bell correlations in a bose-einstein condensate. *Science*, 352(6284):441–444, Apr 2016.

[524] U. Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77:259–315, Apr 2005.

[525] M. Schreiber, S. S. Hodgman, P. Bordia, H. P. Luschen, M. H. Fischer, R. Vosk, E. Altman, U. Schneider, and I. Bloch. Observation of many-body localization of interacting fermions in a quasirandom optical lattice. *Science*, 349(6250):842–845, Jul 2015.

[526] N. Schuch, D. Perez-Garcia, and I. Cirac. Classifying quantum phases using matrix product states and projected entangled pair states. *Phys. Rev. B*, 84(16), Oct 2011.

[527] M. Schuld. *Supervised learning with quantum computers*. Springer, 2018.

[528] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*, 99:032331, Mar 2019.

[529] M. Schuld, A. Bocharov, K. Svore, and N. Wiebe. Circuit-centric quantum classifiers. arXiv:1804.00633, 2018.

[530] M. Schuld, I. Sinayskiy, and F. Petruccione. The quest for a Quantum Neural Network. *Quantum Inf. Process.*, 13(11):2567–2586, 2014.

[531] M. Schuld, I. Sinayskiy, and F. Petruccione. An introduction to quantum machine learning. *Contemp. Phys.*, 56(2):172–185, 2015.

[532] M. Schulte, C. Lisdat, P. O. Schmidt, U. Sterr, and K. Hammerer. Prospects and challenges for squeezing-enhanced optical atomic clocks. *Nat. Commun.*, 11(1):1–10, 2020.

[533] R. Schwonnek, K. T. Goh, I. W. . Primaatmaja, E. Y.-Z. Tan, R. Wolf, V. Scarani, and C. C.-W. Lim. Robust device-independent quantum key distribution. arXiv:2005.02691, 2020.

[534] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. P. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[535] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. P. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis. Protein structure prediction using multiple deep neural networks in the 13th critical assessment of protein structure prediction (casp13). *Proteins*, 87(12):1141–1148, 2019.

[536] G. Sentís, E. Bagan, J. Calsamiglia, and R. Muñoz Tapia. Programmable discrimination with an error margin. *Phys. Rev. A*, 88:052304, Nov 2013.

[537] G. Sentís, A. Monràs, R. Muñoz-Tapia, J. Calsamiglia, and E. Bagan. Unsupervised classification of quantum data. arXiv:1903.01391, 2019.

[538] A. Setter, M. Toroš, J. F. Ralph, and H. Ulbricht. Real-time kalman filter: Cooling of an optically levitated nanoparticle. *Phys. Rev. A*, 97:033822, Mar 2018.

[539] J. Shang, Y. Wang, M. Chen, J. Dai, X. Zhou, J. Kuttner, G. Hilt, X. Shao, M. Gottfried, and K. Wu. Assembling molecular sierpiński triangle fractals. *Nat. Chem.*, 7(5):389–393, 2015.

[540] A. Shankar, L. Salvi, M. L. Chiofalo, N. Poli, and M. J. Holland. Squeezed state metrology with bragg interferometers operating in a cavity. *Quantum Sci. Technol.*, 4(4):045010, Oct 2019.

[541] S. Shankar, M. Hatridge, Z. Leghtas, K. M. Sliwa, A. Narla, U. Vool, S. M. Girvin, L. Frunzio, M. Mirrahimi, and M. H. Devoret. Autonomously stabilized entanglement between two superconducting quantum bits. *Nature*, 504(7480):419–422, 2013.

[542] E. Shchukin, F. Schmidt, and P. van Loock. Waiting time in quantum repeaters with probabilistic entanglement swapping. *Phys. Rev. A*, 100:032322, Sep 2019.

[543] D. Sholl and J. A Steckel. *Density functional theory: a practical introduction.* John Wiley & Sons, 2011.

[544] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[545] P. W. Shor. Fault-tolerant quantum computation. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 56–65. IEEE, 1996.

[546] N. I. Shvetsov-Shilovski. Semiclassical two-step model for ionization by a strong laser pulse: Further developments and applications. arXiv:2102.00503, 2021.

[547] H. T. Siegelmann. Computation beyond the turing limit. *Science*, 268(5210):545–548, 1995.

[548] H. T. Siegelmann and E. D. Sontag. Turing computability with neural nets. *Appl. Math. Lett.*, 4(6):77–80, 1991.

[549] H. T. Siegelmann and E. D. Sontag. On the computational power of neural nets. *J. Comput. Syst. Sci.*, 50(1):132–150, 1995.

[550] I. Siemon, A. Holevo, and R. F. Werner. Unbounded generators of dynamical semigroups. *Open Syst. Inf. Dyn.*, 24(04):1740015, Dec 2017.

[551] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[552] C. Simon, M. Afzelius, J. Appel, A. Boyer de la Giroday, S. J. Dewhurst, N. Gisin, C. Y. Hu, F. Jelezko, S. Kröll, J. H. Müller, and et al. Quantum memories. *Euro. Phys. J. D*, 58(1):1–22, Apr 2010.

[553] M. Skotiniotis, W. Dür, and P. Sekatski. Macroscopic superpositions require tremendous measurement devices. *Quantum*, 1:34, November 2017.

[554] M. Skotiniotis, F. Fröwis, W. Dür, and B. Kraus. Algebraic metrology: Nonoptimal but pretty good states and bounds. *Phys. Rev. A*, 92:022323, Aug 2015.

[555] O. Słowik, M. Hebenstreit, B. Kraus, and A. Sawicki. A link between symmetries of critical states and the structure of SLOCC classes in multipartite systems. *Quantum*, 4:300, July 2020.

[556] S. Slussarenko and G. J. Pryde. Photonic quantum information processing: A concise review. *Appl. Phys. Rev.*, 6(4):041303, 2019.

[557] J. Solanpää, J. A. Budagosky, N. I. Shvetsov-Shilovski, A. Castro, A. Rubio, and E. Räsänen. Optimal control of high-harmonic generation by intense few-cycle pulses. *Phys. Rev. A*, 90:053402, Nov 2014.

[558] A. Steane. Multiple-particle interference and quantum error correction. *Proc. R. Soc. Lond. A*, 452(1954):2551–2577, Nov 1996.

[559] G. R. Steinbrecher, J. P. Olson, D. Englund, and J. Carolan. Quantum optical neural networks. arXiv:1808.10047, 2018.

[560] E. M. Stoudenmire and D. J. Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.

[561] R. Stricker, D. Vodola, A. Erhard, L. Postler, M. Meth, M. Ringbauer, Ph. Schindler, T. Monz, M. Müller, and R. Blatt. Experimental deterministic correction of qubit loss. *Nature*, 585(7824):207–210, 2020.

[562] S. J. Summers and R. Werner. Maximal violation of bell's inequalities is generic in quantum field theory. *Commun. Math. Phys.*, 110(2):247–259, 1987.

[563] S. J. Summers and R. F. Werner. The vacuum violates bell's inequalities. *Phys. Lett. A*, 110(5):257–259, 1985.

[564] J. Surace, M. Piani, and L. Tagliacozzo. Simulating the out-of-equilibrium dynamics of local observables by trading entanglement for mixture. *Phys. Rev. B*, 99:235115, Jun 2019.

[565] S. Szalay, M. Pfeffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and Ö. Legeza. Tensor product methods and entanglement optimization for ab initio quantum chemistry. *Int. J. Quantum Chem.*, 115(19):1342–1391, 2015.

[566] S. S. Szigeti, O. Hosten, and S. A. Haine. Will quantum-enhanced atom interferometry ever be useful? prospects for improving cold-atom sensors with quantum entanglement. arXiv:2010.09168, 2020.

[567] S. S. Szigeti, S. P. Nolan, J. D. Close, and S. A. Haine. High-precision quantum-enhanced gravimetry with a bose-einstein condensate. *Phys. Rev. Lett.*, 125:100402, Sep 2020.

[568] L. Tagliacozzo, T. R. de Oliveira, S. Iblisdir, and J. I. Latorre. Scaling of entanglement support for matrix product states. *Phys. Rev. B*, 78:024410, Jul 2008.

[569] T. H. Taminiau, J. Cramer, T. van der Sar, V. V. Dobrovitski, and R. Hanson. Universal control and error correction in multi-qubit spin registers in diamond. *Nat. Nanotechnol.*, 9(3):171, 2014.

[570] K. Temmink. On tensor network methods for one-dimensional open quantum systems. Master's thesis, University of Amsterdam, 2018.

[571] TensorFlow developers. Built-in tensorflow activation functions. `https://www.tensorflow.org/api_docs/python/tf/keras/activations`, 2021. [Online; accessed 7-February-2021].

[572] B. M. Terhal, I. L. Chuang, D. P. DiVincenzo, M. Grassl, and J. A. Smolin. Simulating quantum operations with mixed environments. *Phys. Rev. A*, 60:881–885, Aug 1999.

[573] B. M. Terhal and D. P. DiVincenzo. Problem of equilibration and the computation of correlation functions on a quantum computer. *Phys. Rev. A*, 61:022301, Jan 2000.

[574] F. Ticozzi and L. Viola. Stabilizing entangled states with quasi-local quantum dynamical semigroups. *Philos. Trans. R. Soc. A*, 370(1979):5259–5269, Nov 2012.

[575] M. Tomza, K. Jachymski, R. Gerritsma, A. Negretti, T. Calarco, Z. Idziaszek, and P. S. Julienne. Cold hybrid ion-atom systems. *Rev. Mod. Phys.*, 91:035001, Jul 2019.

[576] G. Torlai and R. Melko. Neural decoder for topological codes. *Phys. Rev. Lett.*, 119(3):030501, 2017.

[577] E. Torrontegui and J. J. García-Ripoll. Unitary quantum perceptron as efficient universal approximator. *EPL*, 125(3):30004, mar 2019.

[578] G. Tóth. Detection of multipartite entanglement in the vicinity of symmetric dicke states. *J. Opt. Soc. Am. B*, 24(2):275–282, Feb 2007.

[579] G. Tóth. Multipartite entanglement and high-precision metrology. *Phys. Rev. A*, 85:022322, Feb 2012.

[580] G. Tóth, T. Moroder, and O. Gühne. Evaluating convex roof entanglement measures. *Phys. Rev. Lett.*, 114:160501, Apr 2015.

[581] G. Toulouse. Theory of the frustration effect: I. *Commun. Phys.*, 2:115–119, 1977.

[582] A. Trask, F. Hill, S. Reed, J. Rae, Ch. Dyer, and P. Blunsom. Neural arithmetic logic units. *arXiv:1808.00508*, 2018.

[583] M. Tse et al. (LIGO). Quantum-enhanced advanced ligo detectors in the era of gravitational-wave astronomy. *Phys. Rev. Lett.*, 123:231107, Dec 2019.

[584] A. Uhlmann. The "transition probability" in the state space of a *-algebra. *Rep. Math. Phys.*, 9(2):273–279, 1976.

[585] R. Ursin, F. Tiefenbacher, T. Schmitt-Manderbach, H. Weier, T. Scheidl, M. Lindenthal, B. Blauensteiner, T. Jennewein, J. Perdigues, P. Trojek, B. Ömer, M. Fürst, M. Meyenburg, J. Rarity, Z. Sodnik, C. Barbieri, H. Weinfurter, and A. Zeilinger. Entanglement-based quantum communication over 144 km. *Nat. Phys.*, 3(7):481–486, 2007.

[586] J. Vannimenus and G. Toulouse. Theory of the frustration effect. ii. ising spins on a square lattice. *J. Phys. C: Solid State Phys.*, 10(18):L537, 1977.

[587] R. A. Vargas-Hernández and R. V. Krems. *Physical Extrapolation of Quantum Observables by Generalization with Gaussian Processes*, pages 171–194. Springer International Publishing, Cham, 2020.

[588] R. A. Vargas-Hernández, J. Sous, M. Berciu, and R. V. Krems. Extrapolating quantum observables with machine learning: Inferring multiple phase transitions from properties of a single phase. *Phys. Rev. Lett.*, 121:255702, Dec 2018.

[589] S. Varsamopoulos, K. Bertels, and C. G. Almudever. Comparing neural network based decoders for the surface code. *IEEE Trans. Comput.*, 69(2):300–311, 2019.

[590] S. Varsamopoulos, B. Criger, and K. Bertels. Decoding small surface codes with feedforward neural networks. *Quantum Sci. Technol.*, 3(1):015004, 2017.

[591] V. Vedral. The role of relative entropy in quantum information theory. *Rev. Mod. Phys.*, 74:197–234, Mar 2002.

[592] V. Vedral, M. Plenio, M. Rippin, and P. Knight. Quantifying entanglement. *Physical Review Letters*, 78(12):2275–2279, Mar 1997.

[593] V. Veitch, H. Mousavian, D. Gottesman, and J. Emerson. The resource theory of stabilizer quantum computation. *New J. Phys.*, 16(1):013009, 2014.

[594] G. Verdon, J. Pye, and M. Broughton. A universal training algorithm for quantum deep learning. *arXiv:1806.09729*, 2018.

[595] F. Verstraete and J. I. Cirac. Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions. *arXiv:0407066*, July 2004.

[596] F. Verstraete and J. I. Cirac. Continuous matrix product states for quantum fields. *Phys. Rev. Lett.*, 104:190405, May 2010.

[597] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac. Matrix product density operators: Simulation of finite-temperature and dissipative systems. *Phys. Rev. Lett.*, 93:207204, Nov 2004.

[598] F. Verstraete, M. Wolf, and I. Cirac. Quantum computation and quantum-state engineering driven by dissipation. *Nat. Phys.*, 5(9):633, 2009.

[599] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac. Criticality, the area law, and the computational power of projected entangled pair states. *Phys. Rev. Lett.*, 96:220601, Jun 2006.

[600] G. Vidal. Entanglement monotones. *Journal of Modern Optics*, 47(2-3):355–376, Feb 2000.

[601] G. Vidal. Class of quantum many-body states that can be efficiently simulated. *Physical Review Letters*, 101(11):110501, September 2008.

[602] G. Vidal and R. F. Werner. Computable measure of entanglement. *Phys. Rev. A*, 65:032314, Feb 2002.

[603] K. Viebahn, M. Sbroscia, E. Carter, J.-C. Yu, and U. Schneider. Matter-wave diffraction from a quasicrystalline optical lattice. *Phys. Rev. Lett.*, 122(11):110404, 2019.

[604] K. G. H. Viebahn. *Quasicrystalline optical lattices for ultracold atoms*. PhD thesis, University of Cambridge, 2018.

[605] G. Waldherr, Y. Wang, S. Zaiser, M. Jamali, T. Schulte-Herbrüggen, H. Abe, T. Ohshima, J. Isoya, J. F. Du, P. Neumann, and J. Wrachtrup. Quantum error correction in a solid-state hybrid spin register. *Nature*, 506(7487):204–207, 2014.

[606] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. S. Kim. Quantum generalisation of feedforward neural networks. *npj Quantum Inf.*, 3(1):36, September 2017.

[607] J. Wang, C. Roberts, G. Vidal, and S. Leichenauer. Anomaly detection with tensor networks. arXiv:2006.02516, 2020.

[608] Q. Wang, Z. Zhang, K. Chen, J. Guan, W. Fang, and M. Ying. Quantum algorithm for fidelity estimation. arXiv:2103.09076, 2021.

[609] W. Wang, Y. Wu, Y. Ma, W. Cai, L. Hu, X. Mu, Y. Xu, Z.-J. Chen, H. Wang, Y. P. Song, H. Yuan, C.-L. Zou, Duan L.-M., and L. Sun. Heisenberg-limited single-mode quantum metrology in a superconducting circuit. *Nat. Commun.*, 10(1):1–6, 2019.

[610] X.-B. Wang, J. Q. You, and F. Nori. Quantum entanglement via two-qubit quantum zeno dynamics. *Phys. Rev. A*, 77:062339, Jun 2008.

[611] G. H. Wannier. Antiferromagnetism. the triangular ising net. *Phys. Rev.*, 79:357–364, Jul 1950.

[612] S. Wehner, D. Elkouss, and R. Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412), 2018.

[613] K. X. Wei, I. Lauer, S. Srinivasan, N. Sundaresan, D. T. McClure, D. Toyli, D. C. McKay, J. M. Gambetta, and S. Sheldon. Verifying multipartite entangled greenberger-horne-zeilinger states via multiple quantum coherences. *Phys. Rev. A*, 101:032343, Mar 2020.

[614] G. Weihs, T. Jennewein, C. Simon, H. Weinfurter, and A. Zeilinger. Violation of bell's inequality under strict einstein locality conditions. *Phys. Rev. Lett.*, 81:5039–5043, Dec 1998.

[615] H. Weimer, A. Kshetrimayum, and R. Orús. Simulation methods for open quantum many-body systems. arXiv:1907.07079, 2020.

[616] P. J Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Netw.*, 1(4):339–356, 1988.

[617] A. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero. Positive tensor network approach for simulating open quantum many-body systems. *Phys. Rev. Lett.*, 116:237201, Jun 2016.

[618] R. F. Werner. An application of bell's inequalities to a quantum state extension problem. *Lett. Math. Phys.*, 17(4):359–363, 1989.

[619] R. F. Werner. Quantum states with einstein-podolsky-rosen correlations admitting a hidden-variable model. *Phys. Rev. A*, 40:4277–4281, Oct 1989.

[620] R. F. Werner. All teleportation and dense coding schemes. *J. Phys. A: Math. Gen.*, 34(35):7081–7094, aug 2001.

[621] R. F. Werner. *Quantum Information Theory – an Invitation*, pages 14–57. Springer Berlin Heidelberg, 2001.

[622] R. F. Werner and M. M. Wolf. All-multipartite bell-correlation inequalities for two dichotomic observables per site. *Phys. Rev. A*, 64:032112, Aug 2001.

[623] J. Werschnik and E. K. U. Gross. Quantum optimal control theory. *J. Phys. B: At., Mol. Opt. Phys*, 40(18):R175–R211, sep 2007.

[624] C. White, M. Zaletel, R. Mong, and G. Refael. Quantum dynamics of thermalizing systems. *Phys. Rev. B*, 97:035127, Jan 2018.

[625] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992.

[626] C. Whittle, E. D. Hall, S. Dwyer, N. Mavalvala, and V. Sudhir et al. (LIGO). Approaching the motional ground state of a 10 kg object. arXiv:2102.12665, 2021.

[627] U.-J. Wiese. Ultracold quantum gases and lattice systems: quantum simulation of lattice gauge theories. *Ann. Phys.*, 525(10-11):777–796, 2013.

[628] S. Wiesner. Conjugate coding. *ACM SIGACT News*, 15(1):78–88, 1983.

[629] Wikipedia contributors. Quantum machine learning — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Quantum_machine_learning&oldid=883275117`, 2019. [Online; accessed 25-February-2019].

[630] Wikipedia contributors. Activation function. `https://en.wikipedia.org/wiki/Activation_function`, 2021. [Online; accessed 7-February-2021].

[631] Wikipedia contributors. D-wave systems. `https://en.wikipedia.org/wiki/D-Wave_Systems`, 2021. [Online; accessed 18-February-2021].

[632] Wikipedia contributors. Id quantique. `https://en.wikipedia.org/wiki/ID_Quantique`, 2021. [Online; accessed 22-March-2021].

[633] Wikipedia contributors. List of datasets for machine-learning research. `https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research`, 2021. [Online; accessed 7-February-2021].

[634] Wikipedia contributors. Magiq technologies, inc. `https://en.wikipedia.org/wiki/MagiQ_Technologies,_Inc.`, 2021. [Online; accessed 22-March-2021].

[635] Wikipedia contributors. Quantum key distribution—commercial. `https://en.wikipedia.org/wiki/Quantum_key_distribution#Commercial`, 2021. [Online; accessed 22-March-2021].

[636] R. J. Williams and D. Zipser. Experimental analysis of the real-time recurrent learning algorithm. *Conn. Sci.*, 1(1):87–111, 1989.

[637] T. M. Wintermantel, Y. Wang, G. Lochead, S. Shevate, G. K. Brennen, and S. Whitlock. Unitary and nonunitary quantum cellular automata with rydberg arrays. *Phys. Rev. Lett.*, 124:070503, Feb 2020.

[638] M. Wolf, Frank Verstraete, M. Hastings, and I. Cirac. Area laws in quantum systems: Mutual information and correlations. *Phys. Rev. Lett.*, 100:070502, Feb 2008.

[639] M. M. Wolf. Quantum channels & operations guided tour. `https://www-m5.ma.tum.de/foswiki/pub/M5/Allgemeines/MichaelWolf/QChannelLecture.pdf`, 2012. [Online; accessed 25-February-2019].

[640] M. M. Wolf. Mathematical foundations of supervised learning. `https://www-m5.ma.tum.de/foswiki/pub/M5/Allgemeines/MA4801_2018S/ML_notes_main.pdf`, 2018. [Online; accessed 25-February-2019].

[641] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[642] Y. Wu, R. Krishnakumar, J. Martínez-Rincón, B. K. Malia, O. Hosten, and M. A. Kasevich. Retrieval of cavity-generated atomic spin squeezing after free-space release. *Phys. Rev. A*, 102:012224, Jul 2020.

[643] Z. L. Xiang, S. Ashhab, J. Q. You, and F. Nori. Hybrid quantum circuits: Superconducting circuits interacting with other quantum systems. *Rev. Mod. Phys.*, 85:623–653, Apr 2013.

[644] K. Xu, M. Zhang, J. L. Li, S. S. Du, K. Kawarabayashi, and S. Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. arXiv:2009.11848, 2021.

[645] K. Yamamoto, Y. Ashida, and N. Kawakami. Rectification in nonequilibrium steady states of open many-body systems. *Phys. Rev. Research*, 2:043343, Dec 2020.

[646] S. Yang, L. Lehman, D. Poilblanc, K. Van Acoleyen, F. Verstraete, J. I. Cirac, and N. Schuch. Edge Theories in Projected Entangled Pair State Models. *Phys. Rev. Lett.*, 112(3):036402, January 2014.

[647] Z.-S. Yuan, Y.-A. Chen, B. Zhao, S. Chen, J. Schmiedmayer, and J.-W. Pan. Experimental demonstration of a bdcz quantum repeater node. *Nature*, 454(7208):1098–1101, 2008.

[648] A. M. Zagoskin. *Quantum engineering: theory and design of quantum coherent structures.* Cambridge University Press, 2011.

[649] B. Zeng, X. Chen, D.-L. Zhou, and X.-G. Wen. Quantum Information Meets Quantum Matter – From Quantum Entanglement to Topological Phase in Many-Body Systems. *arXiv:1508.02595*, August 2015.

[650] X.-M. Zhang, W. Kong, M. U. Farooq, M.-H. Yung, G. Guo, and X. Wang. Detection-based error mitigation using quantum autoencoders. arXiv:2005.04341, 2020.

[651] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020.

[652] T. Zhou and X. Chen. Operator dynamics in a brownian quantum circuit. *Phys. Rev. E*, 99(5):052212, 2019.

[653] M. Zihlmann, D. Perekrestenko, and M. Tschannen. Convolutional recurrent neural networks for electrocardiogram classification. In *2017 Computing in Cardiology (CinC)*, pages 1–4, 2017.

[654] Y. Zou, A. Milsted, and G. Vidal. Conformal fields and operator product expansion in critical quantum spin chains. *Phys. Rev. Lett.*, 124(4), Jan 2020.

[655] M. Zwerger, H. J. Briegel, and W. Dür. Universal and optimal error thresholds for measurement-based entanglement purification. *Phys. Rev. Lett.*, 110:260503, Jun 2013.

[656] M. Zwerger, W. Dür, and H. J. Briegel. Measurement-based quantum repeaters. *Phys. Rev. A*, 85:062326, Jun 2012.

[657] M. Zwolak and G. Vidal. Mixed-state dynamics in one-dimensional quantum lattice systems: a time-dependent superoperator renormalization algorithm. *Phys. Rev. Lett.*, 93:207205, Nov 2004.

Curriculum Vitae

Personal details

Name	Dmytro Olehovych Bondarenko
Date of Birth	January 1993
Place of Birth	Basalychivka, Ukraine
Parents	Oleh and Mariia Bondarenko
Civil Status	married

Education

since 2017	Doctoral candidate, Institute of Theoretical Physics, Leibniz University Hanover, Germany
March 2017	MSc Theoretical and mathematical physics, Ludwig-Maximilians-Universität München and Technische Universität München, Germany
Thesis theme:	Tree tensor network approximations to conformal field theories
Supervisor:	Prof. Dr. Robert König
June 2016 *specialization*:	MSc Physics, Quantum field theory, Taras Shevchenko National University of Kiev, Ukraine
June 2014 *specialization*:	BSc physics, Quantum field theory, Taras Shevchenko National University of Kiev, Ukraine
Thesis theme:	Investigation of effects of Coulomb interaction on stiffness of Dirac materials
Supervisor:	Dr. Oleksandr B. Kashuba
May 2010	High school diploma, Kyivo-Pecherskiy lycee №171 "Leader", Ukraine

Supervised theses

November 2019	Daniel Scheiermann co-supervised with Prof. T. Osborne and R. Wolf *Bachelor thesis*: Noise robustness of quantum neural networks
July 2020	Jannik Eggert co-supervised with Prof. T. Osborne *Bachelor thesis*: Quantum neural networks for cooling down thermal states

Teaching

summer semester 2019	Advanced quantum theory with Prof. R. Werner
summer semester 2018	Theoretical quantum optics with Prof. K. Hammerer
winter semester 2017-2018	Quantum field theory with Prof. H. Frahm
2014	Introduction to non-linear dynamics student-run course at BITP

List of Publications

- F. Anders, A. Idel, P. Feldmann, D. Bondarenko, S. Loriani, K. Lange, J. Peise, M. Gersemann, B. Meyer, S. Abend, N. Gaaloul, C. Schubert, D. Schlippert, L. Santos, E. Rasel, and C. Klempt,
  Momentum entanglement for atom interferometry,
  arXiv:2010.15796 (2020)

- D. Bondarenko and P. Feldmann,
  Quantum autoencoders to denoise quantum data,
  Phys. Rev. Lett. **124**, 130502 (2020)

- K. Beer, D. Bondarenko, T. Farrelly, T. Osborne, R. Salzmann, and R. Wolf,
  Training deep quantum neural networks,
  Nat. Commun. 11,808 (2020)

- K. Beer, D. Bondarenko, A. Hahn, M. Kalabakov, N. Knust, L. Niermann, T. Osborne, C. Schridde, S. Seckmeyer, D. Stiegemann, R. Wolf,
  From categories to anyons: a travelogue,
  arXiv:1811.06670 (2018)