Quantum Science and Technology

PERSPECTIVE • OPEN ACCESS

From architectures to applications: a review of neural quantum states

To cite this article: Hannah Lange *et al* 2024 *Quantum Sci. Technol.* **9** 040501

Quantum Science and Technology

OPEN ACCESS

RECEIVED
8 March 2024

REVISED
11 July 2024

ACCEPTED FOR PUBLICATION
20 August 2024

PUBLISHED
30 September 2024

PERSPECTIVE

From architectures to applications: a review of neural quantum states

Hannah Lange[1,2,3], **Anka Van de Walle**[2,3], **Atiye Abedinnia**[4] **and Annabelle Bohrdt**[2,4,*]

[1] Max-Planck-Institute for Quantum Optics, Hans-Kopfermann-Str.1, D-85748 Garching, Germany
[2] Munich Center for Quantum Science and Technology, Schellingstr. 4, D-80799 Munich, Germany
[3] Ludwig-Maximilians-University Munich, Theresienstr. 37, D-80333 Munich, Germany
[4] University of Regensburg, Universitätsstr. 31, D-93053 Regensburg, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: annabelle.bohrdt@ur.de

Keywords: neural quantum states, quantum many-body systems, variational monte carlo, neural networks

Abstract

Due to the exponential growth of the Hilbert space dimension with system size, the simulation of quantum many-body systems has remained a persistent challenge until today. Here, we review a relatively new class of variational states for the simulation of such systems, namely neural quantum states (NQS), which overcome the exponential scaling by compressing the state in terms of the network parameters rather than storing all exponentially many coefficients needed for an exact parameterization of the state. We introduce the commonly used NQS architectures and their various applications for the simulation of ground and excited states, finite temperature and open system states as well as NQS approaches to simulate the dynamics of quantum states. Furthermore, we discuss NQS in the context of quantum state tomography.

Contents

Quantum many-body systems are of great interest for many research areas, including physics, biology and chemistry. However, their simulation has remained challenging until today, due to the exponential growth of the Hilbert space with the system size, making it exceedingly difficult to parameterize the wave functions of large systems using exact methods. One common approach to overcome this problem are variational methods, where a certain functional form of the quantum state is assumed, with free parameters to be optimized to obtain the best possible representation of the state under investigation. A well established method based on variational wave functions are tensor networks (TN) [1–8], among them variants that can be contracted efficiently, like matrix product states (MPS) [9–11], and which hence allow an efficient evaluation of observables. MPS are restricted to states that obey the area law of entanglement [12, 13], and are hence particularly well suited for one-dimensional gapped quantum systems, although extensions to higher dimensional systems are possible [1, 7, 14]. Another class of methods for the numerical simulation of quantum systems, quantum Monte Carlo (MC) algorithms [15–17], suffers from the sign problem [18, 19] and slow convergence for large system sizes close to critical points or other challenging statistical physics problems [20].

The ability of sufficiently large neural networks to represent any continuous function [21–24] motivated their use for the simulation of quantum states, and was pioneered by Carleo and Troyer [25] in 2017. To date, these so-called neural quantum states (NQS) have been shown to overcome many problems that are inherent to some conventional methods such as MPS: (*i*) Some works have demonstrated that NQS are capable of representing volume-law entangled states [26–30] and can hence in principle be used for a broad range of quantum systems [26–28, 30–33]. In particular, it has been shown that in some cases mappings between NQS and efficiently contractable TNs can be established, e.g. in [34] the authors find that TNs are a subset of the considered NQS [26]. (*ii*) They can be designed to be particularly well suited for two-dimensional problems. Most prominently, some architectures like convolutional neural networks were specifically designed for two-dimensional data; (*iii*) In many cases, they allow for an efficient evaluation of operators, in some cases even global operators like the momentum [35].

NQS are typically used for two distinct tasks: First, they have appeared in the field of quantum state tomography (QST), where they are used for quantum state reconstruction of states prepared in experiments, allowing the estimation of observables that can not be accessed in experiments [36, 37]. Second, they can be used as simulation tools for quantum systems, with a Hamiltonian driven optimization similar to TNs. In this setting no training data is needed [25]. Furthermore, NQS simulations can not only be applied to represent ground states, but also excited states, finite temperature states, the time evolution of quantum states or open systems.

The goal of this article is to give an overview of the current state of NQS, i.e. existing NQS architectures, their training and their performance in quantum state simulation and tomography in comparison to conventional methods. Previously published overviews on neural quantum states can be found in [20, 38–41] in the more general context of neural network applications in quantum physics, or more specifically on NQS and their optimization in [42–45]. Furthermore, another review on NQS [46] appeared after the publication of the first version of this work.

The outline of this review is as follows: We start with an overview of existing NQS architectures and their application to physical systems as well as commonly used design choices. The second part considers applications of NQS, namely the simulation of ground and excited states, finite temperature states, time evolution and open quantum systems. Furthermore, we review QST and hybrid simulation schemes with NQS.

1. A short introduction to neural quantum states (NQS)

In most cases, NQS are used to represent a quantum state

$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} \psi(\boldsymbol{\sigma})\,|\boldsymbol{\sigma}\rangle, \tag{1}$$

for complex $\psi(\boldsymbol{\sigma})$ in a certain basis choice $|\boldsymbol{\sigma}\rangle$ that is given by the d_l different local configurations, e.g. the spin configurations or Fock space configurations $\boldsymbol{\sigma}$. For a system with N sites, $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_N)$ consists of e.g. $\sigma_i = 0, 1$ for spin systems ($d_l = 2$) or $\sigma_i = 0, 1, 2, \ldots n_{\max}$ for bosonic systems ($d_l = n_{\max} + 1$).

The underlying idea of neural quantum states is to use neural networks in order to represent the wave function coefficients $\psi(\boldsymbol{\sigma})$ of the state under investigation. Hereby, the neural network is used as a variational wave function, mapping configurations $\boldsymbol{\sigma}$ to the respective wave function coefficient $\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$, parameterized by the neural network parameters $\boldsymbol{\theta}$. More precisely, the input of the neural network used for the NQS representation are configurations $\boldsymbol{\sigma}$, and the output is

$$\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \sqrt{p_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}e^{i\phi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}, \tag{2}$$

which is often split into its amplitude $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2$ and its phase $\phi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \mathrm{Im}\left(\log \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})\right)$. To feed an input $\boldsymbol{\sigma}$ into the network, $\boldsymbol{\sigma}$ can be one-hot encoded, i.e. the d_l different local configurations are encoded binary, resulting in a matrix $\boldsymbol{\sigma} \in \mathbb{R}^{N \times d_l}$ for every configuration $\boldsymbol{\sigma}$ of length N. Note that this is not necessary for spin-$1/2$ systems where the values of the spins are normally mapped to a sequence of ± 1 or $0, 1$. Furthermore, the input is often embedded into a space of dimension d_h, i.e. using a trainable or physically inspired projection the input is projected onto the d_h dimensional space that the network is operating on.

The main difficulty of variational approaches is to come up with a good representation $\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ of the true wave function coefficients. Here, the great strength of neural networks, namely their expressive power, comes into play: Neural networks with at least one hidden layer, a sufficient number of parameters and an arbitrary non-linear activation function have the ability to represent continuous functions of any—potentially very complicated—form [21–24]. This makes them promising candidates for a successful representation $\psi_{\boldsymbol{\theta}}$ that is close to the exact wave function ψ. In order to obtain this representation $\psi_{\boldsymbol{\theta}}$, the network parameters $\boldsymbol{\theta}$ are adjusted during the training of the NQS, i.e. starting from some initialization of the neural network parameters $\boldsymbol{\theta}_0$, the network parameters are adjusted such that $\psi_{\boldsymbol{\theta}}$ approximates the true wave function ψ at the end of the training.

NQS have been first proposed by Carleo and Troyer in 2017 [25]. In this first work, NQS have been applied for the ground state and dynamics simulations of spin systems. Figure 1(a) shows two exemplary results from the original work for the antiferromagnetic Heisenberg (AFH) model: In the top figure, the relative ground state errors ϵ_{rel} of the NQS energies E_{NQS} compared to exact energies E_{exact}, i.e.

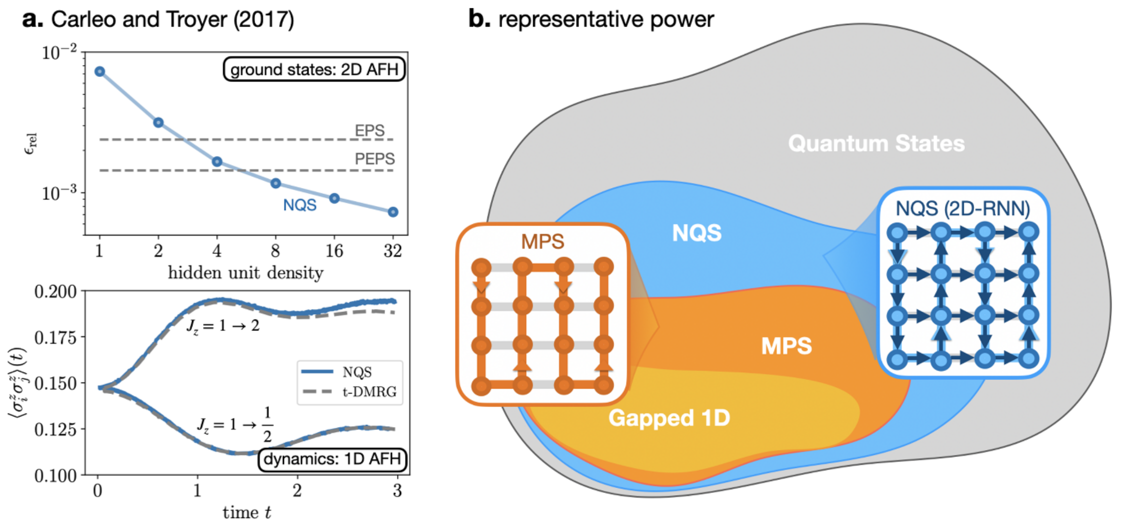

Figure 1. Neural quantum states (NQS): (a) In the seminal work bei Carleo and Troyer [25], NQS are used for ground state representations (top) and dynamics simulations of spin systems (bottom), here for antiferromagnetic Heisenberg (AFH) models on a 10 × 10 lattice (top) and on an 80 site chain. From [25]. Reprinted with permission from AAAS. (b) Representative power of Matrix product states (MPS) and NQS. NQS have been proven to have strictly the same or higher expressive power than practically usable variational tensor networks, see e.g. [26]. This results from the high connectivity between sites of NQS, here exemplary shown for a two-dimensional recurrent neural network (2D-RNN, see blue inset) as considered in [49] on the top right, compared to MPS, for which information is passed through the system in a 1D manner (see orange inset). Figures adapted from [26, 38].

$\epsilon_{\mathrm{rel}} = (E_{\mathrm{NQS}} - E_{\mathrm{exact}})/|E_{\mathrm{exact}}|$, are shown for a 10 × 10 square lattice AFH. Already in this early work, NQS achieve competitive results to state-of-the art 2D methods like entangled plaquette states (EPS) and PEPS when using sufficiently many parameters (hidden units). Correspondingly, the NQS dynamics simulations of a spin chain with 80 sites for two quenches of J_z shows very good agreement with t-DMRG.

Starting from this paper, many works on NQS have appeared in recent years, exploring different NQS architectures, training approaches and their application to various physical systems. Furthermore, many works concern the theoretical representative power of NQS. NQS have been shown to be capable of representing a broad range of quantum states [26–28, 30–33]. To compare their expressivity to more conventional variational approaches like TNs and PEPS, the relationship between them has been studied [26, 47–49]. Some NQS architectures have been proven to have strictly the same or higher expressive power than practically usable variational tensor networks [26], see figure 1. In particular, a range of works have shown that some NQS can encode some volume law states without exponential cost [26–30], although there are volume-law entangled states like the ground state of the Sachdev-Ye-Kitaev model that can not be represented efficiently [50]. Furthermore, [51] develops a combination of TNs and autoregressive NQS, which improves the capabilities compared to both the original TN and NQS. However, in contrast to TNs which are guaranteed to converge after a sufficiently long optimization, the training of NQS involves a non-convex landscape. Hence, it can be challenging to find the actual ground state, even when the NQS ansatz itself is expressive enough to capture it. Advanced training strategies to overcome this issue are discussed in section 3.1.

In contrast to most machine learning applications, the training can be done in a self-contained way without the use of external data. In general, the specific design choices for the NQS can have a significant impact on its performance, which will be the focus of section 2: Besides the choice of architecture, e.g. the way how the real and imaginary parts of the wave function coefficients $\psi_{\boldsymbol{\theta}}$ are modeled. This can be done by splitting $\psi_{\boldsymbol{\theta}}$ into amplitude $p_{\boldsymbol{\theta}}$ and phase $\phi_{\boldsymbol{\theta}}$ parts and using separate networks or separate output nodes / final layers for each part. Another possibility is to use complex network parameters to model the full $\psi_{\boldsymbol{\theta}}$ with a single network. The performance of the wave function does moreover depend on the optimization and the specific task under consideration, which is discussed in section 3.

Similar to Monte Carlo methods, observables of NQS are evaluated by generating samples $\{\boldsymbol{\sigma}\}$ from the NQS amplitudes, which are used for the estimation of the respective expectation values. Specifically, for an operator $\hat{\mathrm{O}}$, the expectation value can be written as

$$\langle\hat{\mathrm{O}}\rangle = \frac{\langle\psi_{\boldsymbol{\theta}}|\hat{\mathrm{O}}|\psi_{\boldsymbol{\theta}}\rangle}{\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle} = \sum_{\boldsymbol{\sigma},\boldsymbol{\sigma}'} \frac{\langle\psi_{\boldsymbol{\theta}}|\boldsymbol{\sigma}\rangle\langle\boldsymbol{\sigma}|\hat{\mathrm{O}}|\boldsymbol{\sigma}'\rangle\langle\boldsymbol{\sigma}'|\psi_{\boldsymbol{\theta}}\rangle}{\sum_{\boldsymbol{\sigma}''}\langle\psi_{\boldsymbol{\theta}}|\boldsymbol{\sigma}''\rangle\langle\boldsymbol{\sigma}''|\psi_{\boldsymbol{\theta}}\rangle} = \sum_{\boldsymbol{\sigma}} P_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \sum_{\boldsymbol{\sigma}'} \frac{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}')}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})} \langle\boldsymbol{\sigma}|\hat{\mathrm{O}}|\boldsymbol{\sigma}'\rangle \approx \langle O^{\mathrm{loc}}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})\rangle_{\boldsymbol{\sigma}} \quad (3)$$

with the probability for each configuration $P_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ and the local estimator $O_{\boldsymbol{\theta}}^{\mathrm{loc}}(\boldsymbol{\sigma})$, defined as

$$P_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \frac{|\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2}{\sum_{\boldsymbol{\sigma}''}|\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}'')|^2} \quad \text{and} \quad O_{\boldsymbol{\theta}}^{\mathrm{loc}}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\sigma}'} \frac{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}')}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})} \langle \boldsymbol{\sigma}|\hat{O}|\boldsymbol{\sigma}'\rangle \tag{4}$$

respectively, as well as the Monte Carlo average $\langle\cdot\rangle_{\boldsymbol{\sigma}}$. For operators involving only a limited number of matrix elements, namely local operators or global operators that do not require the calculation of higher order correlations, $O_{\boldsymbol{\theta}}^{\mathrm{loc}}$ can be evaluated very efficiently [52]. The computational cost of equation (3) results from the generation of samples $\boldsymbol{\sigma}$ from $|\psi_{\boldsymbol{\theta}}|^2$, as well as from the evaluation of the wave function amplitudes for $\boldsymbol{\sigma}$ and its connected samples $\boldsymbol{\sigma}'$. The computational cost of the former strongly depends on $\psi_{\boldsymbol{\theta}}$ being normalized or not, since in the latter case samples can not directly be generated from the wave function and more elaborate approaches like Metropolis sampling are needed. Normalized NQS, using so-called autoregressive architectures, are the topic of section 2.5. Autoregressive NQS can be designed to obey certain symmetries like $U(1)$ symmetries as discussed in section 2.5.1. For non-autoregressive NQS symmetries can be taken into account in the Monte Carlo sampling.

2. NQS architectures

Neural network quantum states can be implemented using several techniques, including various neural network architectures and different representations of phase and amplitude parts of the wave function. Each architecture comes with its advantages and specialized training strategies, see also [120]. Additionally, the choice of architecture can also depend on the physical model under investigation.

In this section, we discuss commonly used architectures, their application to physical systems in the literature as well as their advantages and downsides compared to other ansätze. In figure 2 the type of architectures used in the works cited in this review are shown. It can be seen that the field started with representations in terms of restricted Boltzmann machines (RBMs, section 2.2), as in the seminal work by Carleo and Troyer in 2017 [25]. In the first two years after that, mainly works with NQS based on convolutional neural networks (CNNs, section 2.3) and feed forward neural networks (FFNNs, section 2.1) appeared, while in recent years autoregressive networks (section 2.5) and transformer neural networks (section 2.6) have gained attention. Furthermore, after a focus on spin systems in the early stage of development of NQS architectures, the field turns towards the simulation of fermionic systems (section 2.7). At the time of publication of this (revised) work (June 2024), works on many different architectures appear with a similar fraction. To the best of our knowledge, this is mostly due to the fact that it is not clear at first sight and which NQS architecture is most suitable for a given physical problem is a major open question in the field. Nevertheless, most architectures have certain strengths, which we attempt to summarize in table 1 and in the remaining part of this section.

2.1. Feed forward neural networks (FFNNs)

A feed forward neural network (FFNN), often represented by the structure of a multi-layer perceptron (MLP), is the fundamental building block of artificial neural networks. It is composed of distinct layers of neurons, including an input layer that receives the data, one or more hidden layers where computations are performed, and an output layer that delivers the final result, see figure 3. Within each layer l, each neuron is assigned a bias $b_j^{(l)}$, and is linked to neurons in the adjacent layers through connections $w_{i,j}^{(l)}$. These weights and biases are crucial as they are iteratively adjusted during the network's training, primarily using backpropagation and optimization techniques like gradient descent. The activation functions applied to each neuron's output introduce non-linearity, enabling the network to model complex relationships. In a fully connected FFNN, every neuron in a layer is connected to all neurons in the next layer. The value of each neuron, $a_j^{(l)}$, in layer l, can be described mathematically as:

$$a_j^{(l)} = f\left(\sum_{i=1}^{n} w_{ij}^{(l)} \cdot a_i^{(l-1)} + b_j^{(l)}\right), \tag{5}$$

where $w_{ij}^{(l)}$ represents the weight from the i-th neuron in layer $l-1$ to the j-th neuron in layer l, $a_i^{(l-1)}$ is the activation of the i-th neuron in layer $l-1$, and $b_j^{(l)}$ is the bias of the j-th neuron in layer l. The function f denotes the activation function. This straightforward, yet powerful structure makes FFNNs a vital component in the field of neural networks and deep learning, and has lead to a range of applications in the context of NQS:

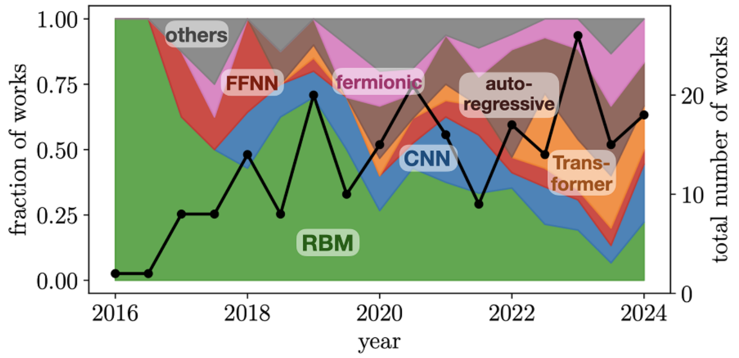

Figure 2. Architectures used in the works cited in this review in a time interval of 6 months from 2016 to June 2024: The fraction of works with implementations of variants of feed forward neural networks (FFNNs, section 2.1), restricted Boltzmann machines (RBMs, section 2.2), convolutional neural networks (CNNs, section 2.3), transformers (section 2.6), autoregressive networks (section 2.5), fermionic networks (section 2.7) or other architectures. The total number of works is shown in black; note that for 2024 we consider only works until June 2024. In some works, several architectures are used, and are taken into account as separate works for each architecture here. Autoregressive transformers are counted twice.

Table 1. Overview of NQS architectures and their applications to ground, excited states, dynamics, finite temperature states, and open systems.

		Exemplary Works and Considered Physical Systems	
Architecture	Features	Ground and Excited States, Finite T States	Open Systems and Dynamics
Feed Forward Neural Networks (section 2.1)	simplicity	(frustrated) spin systems [53, 54], bosons [53, 54, 56–58]	spin systems [55] —
Restricted Boltzmann Machines (section 2.2)	well studied, e.g. in terms of expressivity; interpretability	(frustrated) spin systems [59–64], spin liquids [48, 65], topologically ordered states [31, 66–69], bosons [53, 72–74], fermions [78], molecules [79]	spin systems in 1D [25, 55, 70], ladders [71] and 2D [75–77]
Convolutional Neural Networks (section 2.3)	incorporate lattice symmetries	frustrated spin systems [25, 45, 80–87] on various lattice geometries [90–94]	2D spin systems [75, 88, 89]
Graph Neural Networks (section 2.4)	applicable to any lattice geometry	(frustrated) spin systems [95] and bosons [96] on various lattice geometries	—
Transformer Neural Networks (section 2.6)	self-attention mechanism	(frustrated) spin systems [97–100], Rydberg states [101–103], quantum chemistry [104, 106–108]	spin systems [40, 104, 105]
Autoregressive Neural Networks (section 2.5)	perfect sampling	(frustrated) spin systems [34, 98, 109–112], spin glass [113], topologically ordered (bosonic) states [115, 116], Rydberg states [101, 103, 117], fermions [35, 118]	spin systems [105, 114], Rydberg states [119]

Cai and Liu [54] uses FFNNs to describe ground states of different one-dimensional systems, as well as spinless fermions and the frustrated $J_1 - J_2$ spin-1/2 model in 2D. Choo *et al* [53] explores the possibility to directly target excited states, see section 3.2, and compares the capabilities of FFNNs and restricted Boltzmann machines (section 2.2) to represent excited states of the one-dimensional Heisenberg and Bose–Hubbard models. A Bose–Hubbard model on a ladder with strong magnetic flux is studied using a FFNN in [56]. In [57], a FFNN is trained to represent the ground state of the one- and two-dimensional Bose–Hubbard model. By using the particle number as well as the interaction strength U as additional input parameters to the network, the ground state can be directly obtained without or with little re-training for different Hamiltonian parameters.

Furthermore, FFNNs were applied to simulate quantum systems with continuous degrees of freedom. In [58], a FFNN is used to simulate the ground state of the Calogero-Sutherland model in one dimension and Efimov bound states in three dimensions, where the particle positions in real space are used as input to the network. Another approach to use FFNNs to simulate continuous quantum systems was taken in [121] using

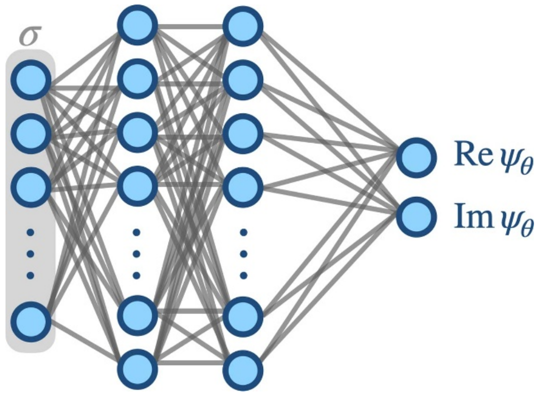

Figure 3. Feed forward neural network (FFNN): The physical state $\boldsymbol{\sigma}$ (blue circles enclosed by the grey rectangle) is taken as the input. Blue circles denote the nodes of the network, grey lines the connecting weight matrices. The output of the neural network is the wave function coefficient $\psi(\boldsymbol{\sigma})$, here split into its amplitude and phase.

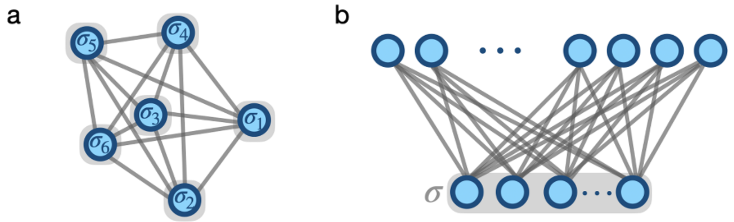

Figure 4. (a) Hopfield network: All-to-all connections between the physical sites are learned by the network. (b) Restricted Boltzmann machine (RBM): An additional set of hidden, unphysical nodes is introduced. Furthermore, the Boltzmann machine is restricted to only inter-layer connections which connect the physical (visible) layer and the hidden layer.

Radial Basis Function (RBF) networks. RBF networks consist, as FFNNs, of an input layer, one or more hidden layers with RBFs as activation functions, and an output layer. RBFs are of the general form

$$f(x) = \phi\left(\|\boldsymbol{x} - \boldsymbol{c}\|\right), \tag{6}$$

where $\boldsymbol{x}$ is a point in the input space, $\boldsymbol{c}$ is the center of the radial basis function, $\|\cdot\|$ denotes a distance measure, such as the Euclidean distance, and ϕ is a radial function, such as the Gaussian function. The neurons' activation is thus determined by the distance of the input $\boldsymbol{x}$ from certain points in the input space $\boldsymbol{c}$, known as centers, which are trainable network parameters. The RBF activation functions strongly depend on the distance to a center point, which allows them to capture variations in data that are radially symmetric. In [121], the input to the RBF corresponds to the quantum numbers of e.g. an undisturbed quantum harmonic oscillator, and the network parameters are optimized to represent the ground state of a quantum harmonic oscillator with an additional applied field.

2.2. Restricted Boltzmann machines (RBMs)

Restricted Boltzmann machines are energy based models, i.e. they are governed by an energy function $E_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ for configurations $\boldsymbol{\sigma}$. Using statistical physics, the respective probability distribution of these models is directly related,

$$p_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right) = \frac{1}{Z_{\boldsymbol{\theta}}}\exp\left(-E_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right)\right), \tag{7}$$

with the normalization constant $Z_{\boldsymbol{\theta}}$. A first example for energy based models were Hopfield networks [122] shown in figure 4(a), which consist of all-to-all connected nodes with connections W_{ij} and the biases b_i, similar to an Ising model with long-range interactions and local magnetic field.

When being used to model physical systems, the number of nodes in a Hopfield model corresponds to the number of physical sites in the system under consideration (*visible nodes* $\boldsymbol{\sigma}$). In contrast, Boltzmann machines (BMs) increase the expressiveness by introducing additional, unphysical nodes (*hidden nodes* $\boldsymbol{h}$)

and the respective connections that increase the expressiveness of the network. With their all-to-all connections between all visible and hidden nodes, BMs are very expressive but can be hard to train. Hence, they are mostly used in their restricted version, see figure 4(b), were only visible-to-hidden node connections W_{ij} and no hidden-to-hidden or visible-to-visible node connections are considered. Analogously to statistical physics, the energy of a restricted Boltzmann machine (RBM) is given by

$$E_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma},\boldsymbol{h}\right) = -\sum_{ij} W_{ij}h_i\sigma_j - \sum_j b_j\sigma_j - \sum_i c_i h_i, \tag{8}$$

with the biases in the visible and hidden layers, b_j and c_i, respectively. Using equation (7) and summing over the hidden nodes $\mathbf{h}$, the corresponding probability for a given input configuration of physical sites σ is given by [123]

$$p_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right) = \sum_{\boldsymbol{h}} p_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma},\boldsymbol{h}\right) = \sum_{\boldsymbol{h}} \frac{\exp\left(-E_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma},\boldsymbol{h}\right)\right)}{Z_{\boldsymbol{\theta}}}. \tag{9}$$

An overview on the application of RBMs in physics can be found in [123].

To use RBMs for representing quantum states, apart from the amplitude $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2$ a phase of the RBM has to be defined. This can be done in several ways, e.g. by making the network parameters $\boldsymbol{\theta}$ complex [25] or by modeling the phase with a separate RBM [124].

The expressivity of the RBM ansatz is often studied in the framework of tensor networks [47, 48, 67, 125], using the entanglement that can be captured with an ansatz as an indicator for the representability. For RBMs, the connectivity between visible and hidden layers, that indirectly couples all sites of the physical system, allows for the entanglement entropy to scale with a subregion's volume, in contrast to its area [27]. In particular, this can make RBMs more efficient in capturing volume-law entangled states compared to e.g. MPS or PEPS. However, the efficiency of shallow RBMs to represent general quantum states has limitations, but they can be overcome with deep RBMs [24, 28, 126, 127]. This was further confirmed empirically e.g. in [128, 129], where random matrix product states were learned with shallow and deep RBMs using supervised approaches. Furthermore, RBMs can, due to their connectivity, straightforwardly be applied to higher dimensions, e.g. 3D systems [130].

RBMs have been used for modeling a large number of physical systems, among them frustrated spin systems [59, 60], spin liquids [65], topologically ordered states [31, 66, 68], the Toric code [66, 69], Bose–Hubbard models [53, 72–74], strongly interacting fermionic systems [78], boson—fermion coupled systems such as electron—phonon coupled systems [131], and molecules [79]. In these works, often variants of RBMs are used, e.g. the correlator RBM where correlations are introduced into the RBM energy functional based on physical insights [69]. Another modification, the convolutional RBM (CRBM) (see section 2.3), makes use of the fact that physical models are typically translationally invariant and feature local interactions. This is taken into account by introducing an additional convolutional layer between the visible and the hidden layers and is employed e.g. in [132, 133] for the simulation of Ising and Kiteav models as well as the Hubbard model. Furthermore, the implementation of symmetries was shown to improve the results [62]. In [63, 64] further symmetries such as non-abelian or anyonic symmetries are considered. RBMs have also been used for the simulation of real-time dynamics [25, 70, 71, 75–77], see also section 3.3.

2.3. Convolutional neural networks (CNNs) and group CNNs

Convolutional neural networks (CNNs) are used in processing data with a grid-like topology, most commonly two-dimensional data like images. The building blocks of CNNs are shown in figure 5: First, convolutional layers employ filters or kernels to scan the input, which can detect local patterns and capture spatial relationships. Basically, each filter in the network uses the same weights for different parts of the input, making CNNs translationally invariant. This approach significantly reduces the number of parameters compared to fully connected networks. Second, pooling layers downsample the spatial dimensions, reducing computational complexity while preserving important features. The final layer of a CNN typically consists of one or more fully connected layers.

Each convolution layer consists of feature maps a and kernels k. The kernel k is slid over the input image (or feature maps at later layers), and for each translation $(x-\tilde{x}, y-\tilde{y})$, the the kernel values $k_i(x-\tilde{x}, y-\tilde{y})$ are multiplied with the translated input feature values $a_i(\tilde{x},\tilde{y})$. In total, the convolution of the two-dimensional input data/feature map a is

$$\tilde{a}\left(x,y\right) = \left[a*k\right]\left(x,y\right) = \sum_{\tilde{x},\tilde{y}}\sum_i a_i\left(\tilde{x},\tilde{y}\right)k_i\left(x-\tilde{x},y-\tilde{y}\right). \tag{10}$$

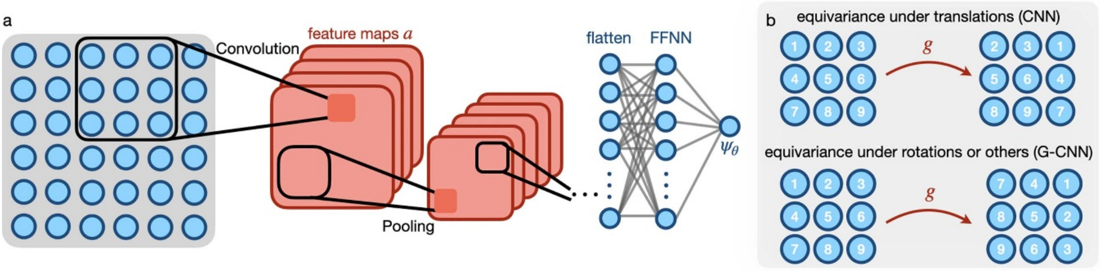

Figure 5. (a) Convolutional neural network (CNN) with convolutions as defined in equation (10), pooling layers that downsample the spatial dimension and fully connected layers as final layers. (b) Comparison of equivariance w.r.t. the translational group present in conventional CNNs (top) and w.r.t. other groups, e.g. C_4 rotations (bottom) that can be implemented in G-CNNs.

The result of this operation is a new feature map $\tilde{a}$, which is the input of the subsequent layers. The index i refers to the value of the $i-th$ channel of the input feature. For an RGB image, for example, there are three channels (red, green, blue), and $a_i(\tilde{x},\tilde{y})$ is the intensity of one of these colors at pixel position $(\tilde{x},\tilde{y})$. $k_i(x,y)$ is the the value of the $i-th$ channel of the kernel at position (x,y) within the kernel.

In the context of NQS, CNNs are regarded as a viable approach to deal with the properties of two-dimensional systems. The application of CNNs to solve the highly frustrated J_1-J_2 antiferromagnetic Heisenberg model was first introduced in [80]. In these systems the sign problem remains a significant challenge for quantum Monte Carlo approaches. Liang *et al* [80] demonstrates how CNNs effectively tackle the challenging problem of finding the ground state of such models. Furthermore, [30] shows that deep CNNs can encode volume-law entangled states efficiently, requiring only $O(\sqrt{N})$ parameters to represent a 2D system with N particles, instead of $O(N)$ for RBMs or $O(N^2)$ for fully connected networks. In a typical neural network, adding more layers, i.e. making the network deeper, theoretically allows the network to learn more complex features and improve its performance on tasks. However, in practice, when the network gets too deep, one faces problems such as vanishing gradients, where the gradients (which are used to update the weights in the network during training) become very small and make learning very slow or even stop it entirely. This problem is addressed e.g. in [87] using skip connections that allow to bypass some layers and layer normalization, where inputs for all neurons within the same layer are normalized for each sample. Furthermore, a variant of stochastic reconfiguration (SR) tailored for large parameter numbers is used in this work. Besides these ground state calculations, CNNs have also been applied for dynamics simulations [75, 88, 89], see also section 3.3.

In [90], a novel approach is introduced for adapting CNNs to other common lattice structures such as triangular lattices, which are somewhat analogous to sheared square lattices, allowing the application of regular CNN filters. In the same work, the authors consider honeycomb and Kagome lattices, where the key techniques involve augmenting these lattices with strategically placed virtual vertices, effectively transforming them into grid-like structures akin to triangular lattices. This allows for the application of standard CNN convolutional kernels while preserving the unique properties of the original lattices. The method enhances information processing and exchange, expanding the receptive field and enabling the analysis of varied local structures and staggered arrangements unique to these lattices.

Although CNNs exhibit translational invariance, they lack the ability to learn additional types of symmetries, such as rotation or mirror symmetries. Typically, data augmentation is employed to train the model for these specific symmetries [86]. In [81] the wave function was symmetrized in order to incorporate the rotational symmetries, see also section 2.8. A more intrinsic solution is the development of group convolutional neural networks (G-CNNs). These networks extend the capabilities of standard CNNs by using group theory, allowing them to automatically incorporate various symmetries, see figure 5(b). The key component of G-CNNs is the group convolution operation. An equivariant convolution ensures that if the input is transformed (e.g. rotated), the output feature maps will be transformed in the same way. The group-equivariant convolution of the input/feature map a with a kernel k under the group G evaluated at a group element g corresponds to

$$[a *_G k](g) = \sum_{h\in G} a(h)\,k\left(g^{-1}h\right). \tag{11}$$

This convolution operation is designed to be equivariant to the transformations in the group G. G can be a group of (discrete) rotations, translations, or other transformations. Roth and MacDonald [91] considers

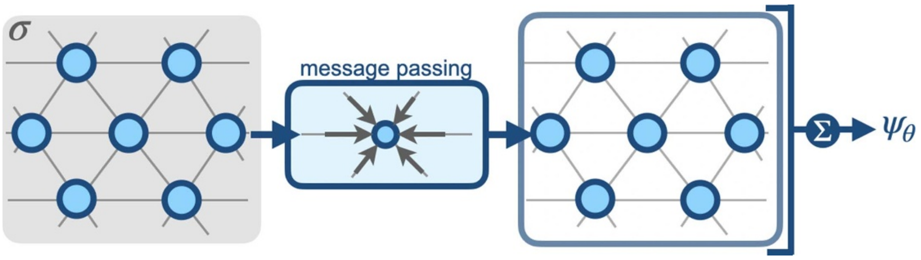

Figure 6. Graph neural network: In each permutation equivariant layer, the weights are updated through typically pairwise message passing, i.e. the value of a given node is updated according to the values of its immediate neighbors. Graph neural networks can easily be applied to any lattice geometry, here we show a hexagonal lattice as an example.

the full wallpaper group, consisting of translation, rotation and mirror symmetry. The first convolution takes the input $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_N)$, where i are the positions in the lattice, and transforms it as

$$a^1(g) = f\left(\sum_{\boldsymbol{i}} k\left(g^{-1}\boldsymbol{i}\right)\sigma_{\boldsymbol{i}}\right), \tag{12}$$

where f is a point-wise non-linear activation function, to obtain the value of the feature map $a(g)$ for the group element g. This first (embedding) layer thus generates equivariant feature maps, which are indexed with group elements, from the input. After repeating the application of group convolution and non-linearity for l layers, the wave function coefficient $\psi(\boldsymbol{\sigma})$ is determined as

$$\psi(\boldsymbol{\sigma}) = \sum_g \chi_{g^{-1}} \exp\left(a_g^l\right), \tag{13}$$

where χ_g is the character of the symmetry operation g.

Roth and MacDonald [91] uses G-CNNs to determine the ground state energy of the $J_1 - J_2$ Heisenberg model on a square and triangular lattice. Roth *et al* [92] underscores the capability of very deep G-CNNs in achieving high-accuracy results for the same lattices, and furthermore directly calculates low-lying excited states by changing the characters of the symmetry operations χ_g accordingly. For a Heisenberg model on a Kagome lattice, one of the most studied models in frustrated magnetism since it is a promising candidate to host exotic spin liquid states, [93] presents a new ground state: the spinon pair density wave (PDW), which does not break the time-reversal and lattice symmetries. G-CNNs are used to study the ground state of the $J - J_d$ Heisenberg model on the Maple-Leaf lattice, which results in dimer state paramagnetic and canted magnetic order phases for different values of J_d/J, [94].

2.4. Graph Neural Networks

Graph neural networks directly take the geometry of the underlying problem into account [134]. In the case of NQS, this means that the lattice geometry of the Hamiltonian under consideration is used as the graph structure. Throughout the graph neural network, this graph structure is kept, see figure 6. In [95], a sublattice encoding, denoting the position of the site within the unit cell, is used as additional input for each site on the lattice. Subsequently, in each permutation equivariant layer of the graph neural network, the values of the nodes are updated through typically pairwise message passing. This means that the value of a given node is updated according to the values of its immediate neighbors, thus directly taking the graph structure into account. The specific details of this updating procedure are design choices, leading e.g. to graph convolutional neural networks [135] or gated graph sequence neural networks [136], where a gated recurrent unit is used.

One advantage of the graph structure and message passing layers is that a transfer to different system sizes is straightforwardly possible, as shown in [95]. Yang *et al* [96] considers the ground state of the hard-core bosonic $t - V$ model on different lattice geometries, such as the Kagome and triangular lattices. Since this constitutes a stoquastic Hamiltonian, no sign structure has to be learned. In [95], the ground state of the $J_1 - J_2$ Heisenberg model on square, triangular, honeycomb and Kagome lattices is studied, in which case a non-trivial sign structure exists. The performance for using complex network weights as well as separate networks for amplitude and phase are compared.

Permutation equivariant message passing has also been used in the context of neural network backflow transformations to simulate interacting fermions in continuous space in [137, 138]. Luo *et al* [139] uses a graph neural network to represent a generalized pair amplitude in the context of a BCS type wave function.

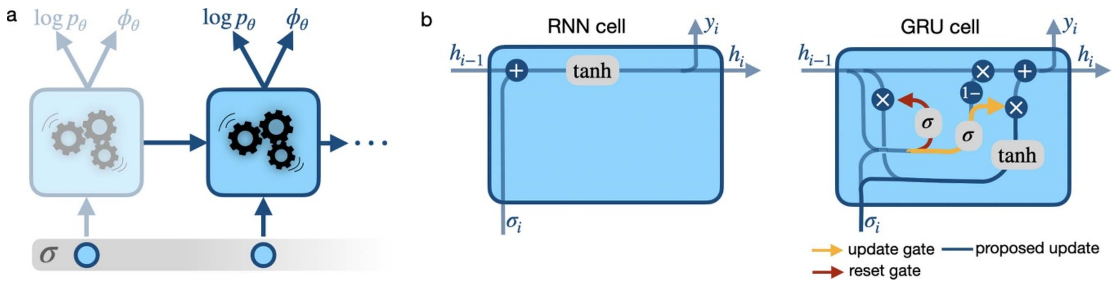

Figure 7. (a) Recurrent neural network (RNN). At each site denoted by the blue circles, the local configuration is passed through the RNN cell. Furthermore, information from previous sites is passed to the next cell (horizontal arrows). (b) Left: A plain vanilla RNN cell, where the local configuration σ_i and the hidden state h_i are simply added and passed through a tanh-activation function. Right: Gated recurrent unit (GRU), enabling the RNN to capture longer-ranged correlations throughout the system. In a GRU, the proposed update (dark blue), reset and update gates (red and orange) decide how much of the old state is kept.

2.5. Autoregressive Networks

Autoregressive architectures are characterized by their normalized amplitudes $p_{\boldsymbol{\theta}} = |\psi_{\boldsymbol{\theta}}|^2$. The use of autoregressive networks for NQS was first proposed by Sharir *et al* [34]: At a local configuration σ_i, the authors propose to mask out the local configurations $j \geqslant i$, and only consider sites $\boldsymbol{\sigma}_{<i} = (\boldsymbol{\sigma}_1, \ldots \boldsymbol{\sigma}_{i-1})$, such that the network represents $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}_i|\boldsymbol{\sigma}_{<i})$. The total probability is given by

$$p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \prod_i^N p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}_i|\boldsymbol{\sigma}_{<i})\,. \tag{14}$$

This allows to normalize $p_{\boldsymbol{\theta}}$ by normalizing each conditional $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}_i|\boldsymbol{\sigma}_{<i})$, and hence to sample directly from the amplitudes instead of more elaborate sampling procedures like Markov chain sampling needed for non-autoregressive architectures. Since the generation of many, uncorrelated samples is crucial for the training, this can yield a speed up and an improvement of the optimization. However, [140] suggests that the autoregressive sampling can in some cases reduce the expressivity of the neural network wave function. Furthermore, the application of SR for optimizing autoregressive NQS can cause problems, as discussed in section 3.1. For further reading on the potential of autoregressive networks in the context of quantum physics and NQS we refer the reader to [41].

Some architectures like CNNs and transformers (see section 2.6) can be made autoregressive by masking out future inputs $\sigma_i, \ldots \sigma_N$ for the i-th input vector [34, 98, 101, 103, 109]. In the following, we discuss a network architecture to which the autoregressive property is inherent due to their recurrent structure: recurrent neural networks.

2.5.1. Recurrent neural networks (RNNs)

Recurrent neural networks (RNNs) consist of several RNN cells, and information is passed from one cell to the next, in a recurrent manner, through the network, as schematically shown in figure 7(a).

The first applications of RNNs to represent quantum states have considered one-dimensional spin systems [110, 111]. In these cases, the RNN is constructed by N cells and the information is passed from the first cell corresponding to the first spin of the 1D chain to the last cell in a recurrent fashion. At each lattice site i, the cell receives a local spin configuration $\boldsymbol{\sigma}_i$ and the so-called *hidden* state $\boldsymbol{h}_{i-1}$ that passes information from previous lattice sites through the network. The cell then outputs the updated hidden state $\boldsymbol{h}_i$ as well as an output $\boldsymbol{y}_i$ that can be used to calculate the local conditional probability and a local phase of the state representation. Normally, each cell is represented by the same weights (weight sharing), but in some contexts the cells can also be chosen to have different weights [113]. In the former case, the RNN architecture is tailored to model bulk properties and hence becomes particularly effective for large systems [111]. Furthermore, it is possible to iteratively retrain on larger and larger systems, which can improve the performance for large systems [111].

The local amplitude at each RNN cell is given by a conditional probability determined by the previous spin configurations $\boldsymbol{\sigma}_{<i}$, i.e. $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}_i|\boldsymbol{\sigma}_{<i})$. The activation function of the RNN's output layer can be chosen such that the local amplitude is normalized and hence also the total amplitude equation (14), making the RNN autoregressive.

To model long-range correlations, it is crucial that the information is passed through the cells in an efficient way. This is usually done by replacing the plain vanilla RNN cells with gated recurrent units (GRU) [141], see figure 7(b), enabling a long-term memory of the RNN [142]. In one-dimensional settings, this modification yields successful representations of spin systems like Heisenberg and transverse field Ising model [110, 111]. For two-dimensional systems, the hidden states can be passed in a 1D snake through the system, similar to MPS calculations, as e.g. in [111]. However, it is also possible to pass the information in a 2D fashion through the system, as proposed in [143] and further improved by introducing a tensorized version of a GRU in [113]. Furthermore, the authors show that an imposed $U(1)$ magnetization conservation and spatial symmetries as well as direct implementation of the Marshall sign rule for spin-1/2 systems improve the results, in agreement with other works [110–112]. This $U(1)$ symmetry is usually imposed by setting $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}_i|\boldsymbol{\sigma}_{<i}) = 0$ if the system with the new sampled $\boldsymbol{\sigma}_i$ violates the corresponding conservation law. For example, for the $U(1)$ particle number conservation of hardcore bosons with local states $\boldsymbol{\sigma}_i = 0(1)$ corresponding to empty (occupied) sites, $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}_i = 1|\boldsymbol{\sigma}_{<i}) = 0$ with if $\boldsymbol{\sigma}_{<i}$ is already in the correct particle number sector. With these modifications, RNNs have been applied in many contexts, including the Heisenberg model on square and triangular lattices [144], prototypical states in quantum information [145], states with topological order [115, 116] and fermionic systems using Jordan-Wigner strings [35]. In the context of real-time dynamics simulations, see also section 3.3, RNNs have been used e.g. in [105, 114, 119].

In order to investigate the expressivity of the RNN ansatz, the authors of [49] present a mapping from tensor networks to 1D MPS-RNNs and 2D tensorized MPS-RNNs, i.e. RNNs with linear or multilinear update rules for the hidden states and quadratic output layers. For linear update rules and one-dimensional settings, MPSs can be mapped to the 1D MPS-RNN with the same number of variational parameters, but not vice versa, making the latter potentially more expressive than MPS. The 2D version of the MPS-RNN receives hidden states from two directions, inspired from projected entangled pair states (PEPS) [146], and features multilinear updates. This architecture is shown to encode an area law of entanglement entropy, but unlike PEPS, it supports perfect sampling and hence efficient evaluations of the wave function. In particular, receiving hidden states from two directions makes the RNN more efficient in state compression compared to the class of TNs which support wave function evaluation in polynomial time [30].

2.6. Transformers

A transformer model relies entirely on an attention mechanism which draws global dependencies between input and output [147]. This self-attention layer in the transformer setup generates all-to-all interactions between the sites in the system. These trainable connections can potentially represent strong connections or correlations, regardless of their position [97, 101]. The transformer first embeds the different given input elements into a unified feature space. This embedding corresponds to a linear projection, with trainable parameters, of the input elements with a dimension d_i to elements with an embedded dimension d_h. The position of the inputs in the sequences are not explicitly modeled in the transformer, but efficiently transformed into abstract representations using positional encoding vectors that are added to the embedded input vectors [97]. Each embedded input element $\sigma_i^{(e)}$ is projected on a query vector (q_i), key vector (k_i) and a value vector (v_i) of the same dimension d_h as the embedded input, given by:

$$q_i = \sum_{l=1}^{d_h} W^q_{i,l}\sigma^{(e)}_{i,l} \qquad k_i = \sum_{l=1}^{d_h} W^k_{i,l}\sigma^{(e)}_{i,l} \qquad v_i = \sum_{l=1}^{d_h} W^v_{i,l}\sigma^{(e)}_{i,l}, \tag{15}$$

with the matrices W^q, W^k and W^v to be the trainable weight matrices of dimension $d_h \times d_h$. The query, key and value matrices are then given by $Q = (q_1, \ldots, q_N)$, $K = (k_1, \ldots, k_N)$ and $V = (v_1, \ldots, v_N)$. In multi-headed attention, each query, key and value vector is mapped to h vectors with trainable weight matrices, with h the number of attention heads. This is indicated in figure 8(middle). A distinction has to be made between two different classes of transformers: the encoder and the decoder. Both architectures consist of the multi-head attention mechanism and a subsequent FFNN. The encoder maps an input sequence of symbolic representations to a context vector, which is used to condition the decoder. The decoder then combines this context vector with the input snapshot to generate the final output probability. Commonly, only a decoder is used for the NQS ansatz. Unlike encoders that map the input to a context vector, the decoder model is usually made autoregressive by adding a mask M to the self-attention layer, which allows connections to all previous elements in the sequence but not to subsequent elements [147] and enables efficient exact sampling from the model [98, 101, 103, 105]. Then, a softmax activation function is applied to the masked dot product of the vectors Q and K. The complete attention formalism in the decoder can be summarized by

$$\mathrm{Attention}(Q, K, V) = softmax\left(\frac{QK}{\sqrt{d_h/h}} + M\right)V, \tag{16}$$

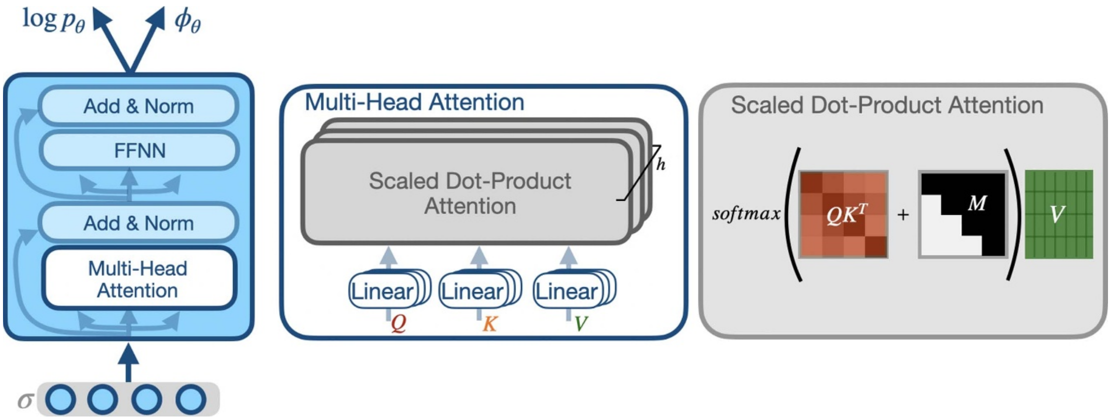

Figure 8. Left: The transformer neural network wave function, adapted from [147]. The input is passed through the multi-head attention mechanism, addition and normalizing layers (Add and Norm) and a feed forward neural network (FFNN). Middle: The multi-head attention itself consists of h scaled dot-product attention layers. Right: The scaled dot-product attention according to equation (16), based on the attention map QK^T, which can in principle represent all-to-all correlations throughout the system. Sites of the system that have not been considered yet are masked out by adding the mask M, before passing it through a softmax activation layer and mutiplying with the values V.

as shown in figure 8(right). In [148] different variants of the attention mechanism are compared.

Transformer quantum states can learn ground state properties of various physical systems, such as the 1D transverse field Ising model and the 1D Heisenberg $J_1 - J_2$ and XYZ model [98]. Comparable results to DMRG calculations have been obtained with a transformer decoder for a 1D frustrated spin model, with a relatively low number of parameters [97]. In [105], transformers have shown to be able to simulate the real-time dynamics and steady state in 1D and 2D transverse field Ising and Heisenberg models [105]. In [104, 106–108] transformers are used in the context of quantum chemistry calculations.

Small modifications of the transformer model, leading to the so-called vision transformers (ViT), inspired the use of patched transformers. This model splits the system into patches, and can be used to calculate the ground state properties of frustrated spin models [97]. A large patch size enhances the efficiency of the transformer, but on the other hand the network output dimension increases exponentially with the patch size, as it encodes the probability distribution over all possible patch states. To overcome this, large patched transformers are introduced in [101]. In this ansatz, the output of the patched transformer is passed to a patched RNN as the initial hidden state, which breaks the large inputs into smaller sub-patches, reducing the output dimension. This model has been shown to accurately capture ground state properties and phase transitions of large Rydberg systems, which can compete with quantum Monte Carlo results [101]. In [149], a combined architecture of a transformer network and a FFNN is used. Hereby, first the transformer maps the physical spins to a high-dimensional feature space. The authors argue that in this feature space the determination of the ground-state properties is simplified, requiring only a single FFNN layer with complex-valued parameters to parameterize the wave function in the second part of the network. The combined architecture is used for the ground state search of the Shastry-Sutherland Model, featuring a spin liquid phase besides phases with plaquette and antiferromagnetic order.

In [98], transformer quantum states have been used to learn ground state properties of a single system, as well as to generalize to different, unseen systems. For the latter, not only the physical degrees of freedom, but also the parameters of the Hamiltonian of the system are used as input. These parameters can be formulated as new elements that have to be passed to the embedding layer. After training the transformer quantum state for a Hamiltonian with different parameters, the transformer is able to generate the ground state for unseen Hamiltonian parameters without any additional training. Although these are with slightly larger error, more accurate results can be achieved with less training then without any *a priori* training. In [102] an encoder-decoder transformer is designed to learn the distribution of measurement outcomes with the Hamiltonian parameters of the system as the input. The transformer is trained on data from different interacting Rydberg arrays and is able to generalize to systems outside of the training set.

In [99, 100], a transformer with a so-called factored attention is used. In contrast to the conventional attention mechanism (16), where the attention weights QK^T and the values V are calculated from both embedded inputs and the positional encoding, factored attention uses QK^T that depend only on positions,

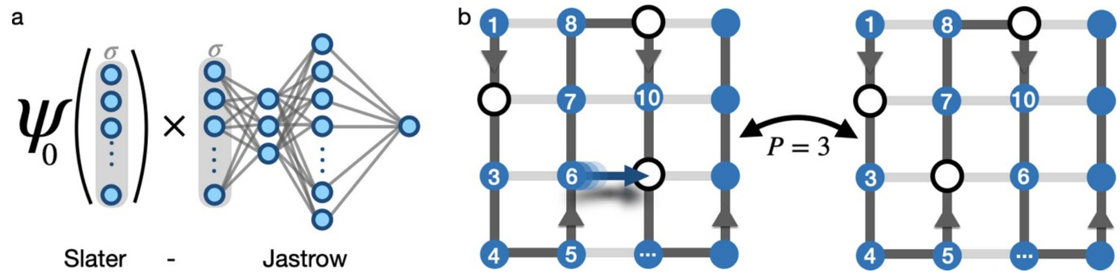

Figure 9. NQS for fermionic systems: (a) Fermionic architectures are typically inspired from Slater-Jastrow ansätze with an antisymmetric Slater determinant and a Jastrow factor that captures the correlations throughout the system. (b) Bosonic architectures can be made fermionic by applying Jordan-Wigner strings. Effectively, this boils down to counting the number of permutations P of configurations that are connected by matrix elements when calculating observables and multiplying the respective terms in equation (19) by $(-1)^P$.

and V that depend only on the embeddings. Rende *et al* [99] shows that training a model with a single self-attention layer with factored attention can be mapped to solving the inverse Potts problem using the pseudo-likelihood method. This method, in combination with the patched transformer, leads to high quality results for the ground-state energy of the $J_1 - J_2$ Heisenberg model [99].

2.7. NQS for fermionic systems

NQS for the representation of fermionic quantum states can be divided into distinctly different ansätze: (*i*) NQS ansätze that inherently incorporate the fermionic statistics and (*ii*) bosonic NQS that are antisymmetrized by a Jordan-Wigner transformation, see figures 9(a) and (b) respectively.

2.7.1. Fermionic architectures

The antisymmetry of NQS ansätze with fermionic statistics can be achieved in various ways. The most commonly used ansatz for fermionic variational wave functions is a Slater-Jastrow-inspired ansatz, where the wave function is constructed from an antisymmetric part ψ_0, typically a Slater determinant, and a Jastrow factor $\mathcal{J}$ capturing the correlations, i.e.

$$|\psi_{\boldsymbol{\theta},\boldsymbol{\nu}}\rangle = \sum_{\boldsymbol{\sigma}} \psi_{0,\boldsymbol{\theta}}(\boldsymbol{\sigma})\,\mathcal{J}_{\boldsymbol{\nu}}(\boldsymbol{\sigma})\,|\boldsymbol{\sigma}\rangle, \tag{17}$$

where in principle both ψ_0 and $\mathcal{J}$ can be parameterized by neural networks with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\nu}$. This is shown schematically in figure 9(a).

In the setting of first quantization, architectures like FermiNet [150] and PauliNet [151, 152] that use Slater determinants ψ_0 reach high accuracies in *ab initio* molecule simulations. However, the evaluation of Slater determinants is costly in first quantization, which is overcome e.g. in [153] by an antisymmetric construction of ψ_0 by deep neural networks. However, these approaches often come at the price of a reduced accuracy [154].

For quantum many-body systems, mostly second quantization is used, despite some exceptions e.g. for repulsively interacting, spin-polarized fermions [155], where the authors chose to model $\psi_0(\boldsymbol{\sigma})\mathcal{J}(\boldsymbol{\sigma})$ in equation (17) by a single neural network. In the works using second quantization, machine learning approaches are used to enhance the expressivity of the Slater-Jastrow ansatz. This can be done by employing NNs to parameterize the Jastrow factor $\mathcal{J}$.

One of the first examples is the RBM+PP architecture in [78], with a slightly different ansatz than equation (17), i.e.

$$|\psi_{\boldsymbol{\theta}}\rangle = \sum_{\boldsymbol{\sigma}} \psi_{\mathrm{ref}}(\boldsymbol{\sigma})\,\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\sigma})\,|\boldsymbol{\sigma}\rangle, \tag{18}$$

where correlations on top of a reference state ψ_{ref} are modeled by a generalized version of an RBM, with additional artificial neurons to mediate entanglement, that is represented by $\mathcal{F}_{\boldsymbol{\theta}}$. In this work, the authors take ψ_{ref} to be a pair-product state (PP) that already incorporates some of the entanglement, and test the architecture for the Fermi-Hubbard model.

Currently, usually two methods are used [156]: (*i*) The hidden fermion determinant state, where neural networks are used to replace the standard Slater determinant with a larger determinant which includes single particle orbitals from additional projected hidden fermions [157, 158]. (*ii*) Neural backflow transformations, which add correlation by making the single particle orbitals of the Slater determinant configuration dependent, with the respective transformation learned by a neural network [138, 150, 151, 159–161]. In [156] the authors show that both (*i*) and (*ii*) can be written as Jastrow-like corrections to the single-particle orbitals. Furthermore, [162, 163] show that using Bloch single-particle wave functions in the Slater determinant allows to simulate large fermionic systems such as moiré materials.

Furthermore, it is worth mentioning that in simulations of lattice models with a Slater-Jastrow variational wave function, the autocorrelation time increases drastically for large system sizes, motivating the development of a fully autoregressive Slater-Jastrow ansatz by combining a Slater determinant with an autoregressive deep neural network as a Jastrow factor [164].

2.7.2. Bosonic architectures with Jordan wigner strings

The other way to simulate fermionic systems using NQS are Jordan-Wigner (JW) transformations, which are used to map the bosonic NQS to a fermionic wave function. Hence, per se bosonic architectures can be used and no special fermionic architecture is needed. The JW transformation is given by

$$\hat{c}_j^{(\dagger)} = \hat{\sigma}^{-(+)} \exp\left(i\pi \sum_{k<j} \hat{\sigma}_k^+ \hat{\sigma}_k^- \right), \tag{19}$$

where indices refer to a one-dimensional labeling of the fermions, the $\hat{\sigma}^{\pm}$ are the spin raising and lowering operators. The resulting annihilation (creation) operators $\hat{c}^{(\dagger)}$ fulfill the fermionic commutation relations. More precisely, for two fermions at site i and j, exchanging these fermions yields a minus sign arising from the argument of the exponential in equation (19). This rule does not have to be implemented in the NQS architecture itself, but only on the level of the calculation of expectation values. For an operator $\hat{O}$ with $\langle \hat{O} \rangle = \langle O_{\boldsymbol{\theta}}^{\mathrm{loc}}(\boldsymbol{\sigma}) \rangle_{\boldsymbol{\sigma}}$ (see equation (3)) with

$$O_{\boldsymbol{\theta}}^{\mathrm{loc}}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\sigma}'} \frac{\langle \boldsymbol{\sigma} | \hat{O} | \boldsymbol{\sigma}' \rangle \langle \boldsymbol{\sigma}' | \psi_{\boldsymbol{\theta}} \rangle}{\langle \boldsymbol{\sigma} | \psi_{\boldsymbol{\theta}} \rangle} \eta_{\boldsymbol{\sigma}\boldsymbol{\sigma}'}, \tag{20}$$

each matrix element is multiplied by a factor $\eta_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} = (-1)^{P_{\boldsymbol{\sigma}\boldsymbol{\sigma}'}}$ if $\boldsymbol{\sigma}'$ is connected to $\boldsymbol{\sigma}$ by $P_{\boldsymbol{\sigma}\boldsymbol{\sigma}'}$ two-particle permutations, see figure 9(b). This method was applied to simulate molecules [165], for Fermi-Hubbard and $t-J$ models [35, 166] and solid state systems [167].

Despite its successful application, JW strings come with the disadvantage that the operators in equation (19) are highly non-local, which can cause problems for some architectures. Whether antisymmetrizing bosonic networks is as efficient as using inherently fermionic architectures is still under debate [29].

2.8. Other design choices

Besides the architecture, other design choices can influence the performance of the NQS:

One choice is the way how amplitude and phase of the NQS are calculated. Hereby, amplitude and phase can be learned by two different, real-valued networks, one real-valued network with two separate output nodes or final layers for phase and amplitude or by one network with complex weights. In [168], a complex-valued RBM and a RBM in which two separate real-valued networks approximate amplitude and phase, are compared for the ground state of the $J_1 - J_2$ model. In a systematic study on small clusters, they show that the complex RBM outperforms the latter.

A second design choice is how to incorporate symmetries in the NQS training in order to restrict the optimization space to states in the target symmetry sector, hence improving the performance [45, 62, 110, 112]. Firstly, global $U(1)$ symmetries can be imposed, see e.g. [110, 112, 118], by restricting to configurations $\boldsymbol{\sigma}$ that obey the respective symmetry, e.g. magnetization or particle number conservation. For autoregressive architectures, this is done by restricting the conditional probabilities to the targeted symmetry, see e.g. section 2.5.1. For non-autoregressive architectures, the Mone Carlo updates can be chosen such that all generated configurations stay in the same symmetry sector. Second, spatial symmetries can be imposed. There are different symmetrizations used in the literature[5], which require to generate new samples $\boldsymbol{\sigma}^S$ that are connected by symmetry transformations $\mathcal{T}$ to the original samples $\boldsymbol{\sigma}$:

[5] We follow [45] here.

(i) bare-symmetry:

$$\psi^S_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \exp\left(\frac{1}{|\mathcal{T}|}\sum_{\boldsymbol{\sigma}^S\in\mathcal{T}(\boldsymbol{\sigma})}\log\left[\psi_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}^S\right)\right]\right) \tag{21}$$

(ii) exp-symmetry

$$\psi^S_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \left(\frac{1}{|\mathcal{T}|}\sum_{\boldsymbol{\sigma}^S\in\mathcal{T}(\boldsymbol{\sigma})}\psi_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}^S\right)\right) \tag{22}$$

(iii) sep-symmetry

$$\psi^S_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \sqrt{\exp\left(\frac{1}{|\mathcal{T}|}\sum_{\boldsymbol{\sigma}^S\in\mathcal{T}(\boldsymbol{\sigma})}2\mathrm{Re}\left\{\log\left[\psi_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}^S\right)\right]\right\}\right)\cdot\exp\left(i\arg\left(\sum_{\boldsymbol{\sigma}^S\in\mathcal{T}(\boldsymbol{\sigma})}\exp\left(i\mathrm{Im}\left\{\log\left[\psi_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}^S\right)\right]\right\}\right)\right)\right)}. \tag{23}$$

Note that only the last keeps the autoregressive property intact since $\sum_{\boldsymbol{\sigma}^S\in\mathcal{T}(\boldsymbol{\sigma})}|\psi^S_{\boldsymbol{\theta}}(\boldsymbol{\sigma}^S)|^2 = \sum_{\boldsymbol{\sigma}^S\in\mathcal{T}(\boldsymbol{\sigma})}|\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}^S)|^2$. Furthermore, architectures that preserve certain symmetries explicitly can be used, such as group CNNs [91] and gauge equivariant neural Networks [169].

Furthermore, the authors of [54] show for the exemplary case of a FFNN that the choice of activation function can strongly influence the performance. Lastly, the number of parameters of the NQS can be varied. In [170] it is argued for the exemplary architecture of an RBM that using overparameterized NQS and subsequently pruning small parameters of the trained model can improve the performance. The compression by pruning is also discussed in [171]. However, increasing the number of parameters does not always yield an improvement: In [172] it is shown that the accuracy of an RBM increases for small widths of the hidden layer α, but saturates at high α. The authors observe that this behavior coincidents with a saturation of the quantum geometric tensor's rank, see equation (31) in section 3.1, i.e. the local dimension of the relevant manifold for the optimized NQS saturates. Lastly, the open-source packages allow to readily optimize code for graphics processing units (GPUs), which can highly accelerate NQS implementations, see also section 2.9. Another direction for fast and energy efficient NQS implementations is spiking neuromorphic hardware [173–175], as implemented for a RBM in [174].

For further reading on design choices beyond the discussion provided here, we refer the reader to Reh *et al* [45], where the performance of RBMs, CNNs and RNNs with different symmetrization strategies are compared.

2.9. Open-source toolboxes

There are several toolboxes that provide open-source implementations of NQS: *NetKet* allows for ground state search, dynamics calculations based on TDVP and p-tVMC as well as state tomography using various architectures and comes with many implemented bosonic and fermionic Hamiltonians [176, 177]. *jVMC* [178], designed for computationally efficient variational Monte Carlo, provides several architectures for ground state search and dynamics simulations as well. *FermiNet* [150] provides ground state simulations for atoms and molecules. All of them are based on Google's JAX library [179]. Lastly, we would like to mention *QuCumber* [180], a RBM based tomography implementation.

3. Applications of NQS

3.1. Ground states

3.1.1. Variational Monte Carlo

To represent the ground state of a given system, neural quantum states are normally trained using variational Monte Carlo (VMC) [181, 182]. VMC is based on variational wave functions $|\psi_{\boldsymbol{\theta}}\rangle$ such as NQS, parameterized by parameters $\boldsymbol{\theta}$. To approximate ground states with NQS, the energy

$$E_{\boldsymbol{\theta}} = \frac{\langle\psi_{\boldsymbol{\theta}}|\hat{H}|\psi_{\boldsymbol{\theta}}\rangle}{\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle} \geqslant E_{\mathrm{gs}}, \tag{24}$$

should be as close as possible to the ground state energy E_{gs}. For variational wave functions $\psi_{\boldsymbol{\theta}}$, this expectation value $E_{\boldsymbol{\theta}}$ can be evaluated from samples $\boldsymbol{\sigma}$ drawn from the wave function's amplitude $|\psi_{\boldsymbol{\theta}}|^2$

according to equation (3). To approximate ground states, NQS are usually trained by minimizing the expectation value of the Hamiltonian, $E_{\boldsymbol{\theta}} = \langle \hat{H} \rangle \approx \langle H_{\mathrm{loc}}(\boldsymbol{\sigma}) \rangle_{\boldsymbol{\sigma}}$, i.e. parameters $\boldsymbol{\theta}$ are updated according to

$$\begin{aligned}
\partial_{\boldsymbol{\theta}_k} E_{\boldsymbol{\theta}} &= 2\mathrm{Re}\left[\sum_{\boldsymbol{\sigma}} P_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right) \frac{\partial_{\boldsymbol{\theta}_k} \psi^*_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right)}{\psi^*_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right)} H^{\mathrm{loc}}_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right)\right] - 2\mathrm{Re}\left[\sum_{\boldsymbol{\sigma}'} P_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}'\right) \frac{\partial_{\boldsymbol{\theta}_k} \psi^*_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}'\right)}{\psi^*_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}'\right)} \sum_{\boldsymbol{\sigma}} P_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right) H^{\mathrm{loc}}_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right)\right] \\
&= \frac{2}{N_s} \mathrm{Re}\left[\sum_{i}^{N_s} \partial_{\boldsymbol{\theta}_k} \mathrm{log}\psi^*_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}_i\right)\left[H_{\mathrm{loc}}\left(\boldsymbol{\sigma}_i\right) - \langle H_{\mathrm{loc}}\left(\boldsymbol{\sigma}'\right)\rangle_{\boldsymbol{\sigma}'}\right]\right] \\
&= 2\mathrm{Re}\langle \partial_{\boldsymbol{\theta}_k} \mathrm{log}\psi^*_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right)\left[H_{\mathrm{loc}}\left(\boldsymbol{\sigma}\right) - \langle H_{\mathrm{loc}}\left(\boldsymbol{\sigma}'\right)\rangle_{\boldsymbol{\sigma}'}\right]\rangle_{\boldsymbol{\sigma}}. \qquad (25)
\end{aligned}$$

For autoregressive architectures, $\langle \psi_{\boldsymbol{\theta}} | \psi_{\boldsymbol{\theta}} \rangle = 1$ and hence the second term vanishes when calculating the derivative of equation (3). However, often $H_{\mathrm{loc}}(\boldsymbol{\sigma})$ is replaced by the covariate $(H_{\mathrm{loc}}(\boldsymbol{\sigma}) - \langle H_{\mathrm{loc}} \rangle)$ to reduce the variance of the gradients [110, 183], leading to the same expression as equation (25). Another approach is to (pre-)train the NQS with experimental or numerical data, see section 4.1.

The optimization of NQS can be done with methods commonly used in machine learning, such as stochastic gradient descent, Adam [184] and AdamW [185]. A more elaborate approach is the SR algorithm [15, 186, 187], which incorporates the knowledge of the geometric structure of the parameter space to adjust the gradient direction [188–190]. The underlying idea[6] of SR is to perform an imaginary time evolution of the variational state $|\psi_{\boldsymbol{\theta}}\rangle$, i.e.

$$|\psi\left(\tau + \delta\tau\right)\rangle = \exp\left(-\delta\tau \hat{H}\right)|\psi_{\boldsymbol{\theta}}\rangle \underbrace{\approx}_{\text{small}\,\delta\tau} |\psi_{\boldsymbol{\theta}}\rangle + |\delta\psi_{\hat{H}}\rangle. \qquad (26)$$

For the latter equality, we have assumed small time steps $\delta\tau$, when the change of the state from the imaginary time evolution is

$$|\delta\psi_{\hat{H}}\rangle = -\delta\tau \hat{H} |\psi_{\boldsymbol{\theta}}\rangle = -\delta\tau \sum_{\boldsymbol{\sigma}} \psi_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right) H^{\mathrm{loc}}\left(\boldsymbol{\sigma}\right)|\boldsymbol{\sigma}\rangle. \qquad (27)$$

Naturally, the evolved state $|\psi(\tau + \delta\tau)\rangle$ has less contributions from higher energy states, decreasing the energy with every imaginary time step $\delta\tau$ [87]. In order to *translate* the evolved state $|\psi(\tau + \delta\tau)\rangle$ to a parameter update $\boldsymbol{\theta} \to \boldsymbol{\theta}'$, a projection onto the variational manifold of $\psi_{\boldsymbol{\theta}'}$ is needed, which is done by minimizing the Fubini-Study (FS) distance $d(\psi(\tau + \delta\tau), \psi_{\boldsymbol{\theta}'})$ [192]. Expanding also for the projected state for small $\delta\tau$, $|\psi_{\boldsymbol{\theta}'}\rangle = |\psi_{\boldsymbol{\theta}}\rangle + |\delta\psi_{\boldsymbol{\theta}}\rangle$, with

$$|\delta\psi_{\boldsymbol{\theta}}\rangle = \sum_{k,\boldsymbol{\sigma}} \frac{\partial\psi_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right)}{\partial\theta_k} \delta\theta_k |\boldsymbol{\sigma}\rangle = \sum_{\boldsymbol{\sigma}} \psi_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right) \sum_k O_k\left(\boldsymbol{\sigma}\right)\delta\theta_k |\boldsymbol{\sigma}\rangle \qquad (28)$$

and $O_k(\boldsymbol{\sigma}) = \frac{1}{\psi_{\boldsymbol{\theta}(\boldsymbol{\sigma})}} \partial_{\boldsymbol{\theta}_k} \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ [25], the FS distance can be written as

$$d\left(\psi\left(\tau + \delta\tau\right), \psi_{\boldsymbol{\theta}'}\right) = ||\bar{O}\delta\boldsymbol{\theta} - \bar{H}^{\mathrm{loc}}||_2 \qquad (29)$$

with $|| \ldots ||_2$ the L^2 norm, $\bar{O}_k(\boldsymbol{\sigma}) = \frac{1}{\sqrt{N_s}}(O_k(\boldsymbol{\sigma}) - \langle O_k(\boldsymbol{\sigma}) \rangle_{\boldsymbol{\sigma}})$ and $\bar{H}^{\mathrm{loc}}(\boldsymbol{\sigma}) = -\frac{\delta\tau}{\sqrt{N_s}}(H^{\mathrm{loc}}(\boldsymbol{\sigma}) - \left\langle H^{\mathrm{loc}}(\boldsymbol{\sigma}) \right\rangle_{\boldsymbol{\sigma}})$. This results in the SR equation

$$\bar{O}\delta\theta = \bar{H}_{\mathrm{loc}} \quad \Leftrightarrow \quad \delta\theta = S^{-1}\bar{O}^{\dagger}\bar{H}_{\mathrm{loc}} = S^{-1}F, \qquad (30)$$

with the quantum geometric tensor

$$S = \bar{O}^{\dagger}\bar{O}, \qquad (31)$$

and the vector of forces

$$F_k = \langle H_{\mathrm{loc}} O^*_k \rangle - \langle H_{\mathrm{loc}} \rangle \langle O^*_k \rangle.$$

Hence, the SR parameter update at the p-th iteration is given by

$$\boldsymbol{\theta}\left(p + 1\right) = \boldsymbol{\theta}\left(p\right) - \gamma\left(p\right) S^{-1} F, \qquad (32)$$

[6] For the motivation of the SR algorithm from imaginary time evolution, we follow [87, 191].

with a scaling parameter $\gamma(p)$ [25].

The SR update equation (32) hence involves an inversion of the S matrix. This inversion comes with the following problems: (*i*) The matrix S has to be estimated to a very high precision to avoid instabilities in the optimization, which requires a large number of samples. (*ii*) S is not necessarily invertible and hence, often a regularization of S is needed for a stable optimization, especially if considered close to critical points or in large spin systems [25, 70, 75]. Recent works indicate that the spectra of the S matrix are distinctly different for non-autoregressive and autoregressive architectures, which can cause problems for regularizing S for the latter [35, 193]. (*iii*) S is a matrix of typically very large dimensions $N_{\boldsymbol{\theta}} \times N_{\boldsymbol{\theta}}$, making the inversion computationally costly since the complexity of inverting S scales with $\mathcal{O}(N_s N_{\boldsymbol{\theta}}^2 + N_{\boldsymbol{\theta}}^3)$ when using direct linear solvers [87]. To enlarge the allowed number of parameters by some orders of magnitudes, large-scale supercomputers are needed [84, 194].

To overcome problem (*iii*), several modifications of SR have been proposed. Among them are iterative solvers such as MINRES [25, 195] which avoid this scaling by iteratively computing the pseudo-inverse of S [25]. However, for large N_s and $N_{\boldsymbol{\theta}}$ typically also the required number of iterations grows. Another method is the sequential local optimization approach, in which SR only optimizes a portion of all parameters to reduce the time cost [196]. Other recent works have proposed modifications of the SR update rule which involve the inversion of a $N_s \times N_s$ matrix instead of S, with N_s the number of samples to estimate the gradient, which is usually smaller than the number of parameters [87, 100]. Apart from that, the performance of SR can be improved with adaptive learning rate solvers, such as the second order Runge Kutta integrator, allowing for an optimal choice of the learning rate [197].

The optimization of NQS can become very difficult due to the in general very rugged and chaotic optimization landscape with many local minima [192, 197, 198]. To overcome this problem, many techniques have been developed. Among them are *variational neural annealing* that applies an artificial temperature to avoid getting stuck in local minima [113, 144, 199], the application of *symmetries* [45, 62, 91, 110, 112] to enforce a training only in the target symmetry sector, *transfer learning*, i.e. the transfer learned properties of small systems to larger system sizes, [111, 200, 201], and *weight pruning* [170, 171].

3.1.2. Optimization challenges

Further improvement can be achieved using complementary optimization methods: Firstly, the NQS can be pretrained with external data, see section 4. Second, the energy resulting from the VMC optimization can be improved by applying a few *Lanczos* steps after optimizing the network parameters using the techniques described above [202]. Applying the Hamiltonian in the Lanczos algorithm to obtain the next Krylov vector corresponds to minimizing the infidelity to the corresponding state and thus necessitates a separate training, rendering typically only very few Krylov vectors accessible. Similar in spirit to Lanczos algorithms, power methods can be used to find the ground states of (gapped) Hamiltonians [203].

A related approach, also based on applying imaginary time evolution, maximizes the fidelity to a target state at each iteration. In contrast to SR, this is done explicitly, i.e. a small imaginary time step $\Delta\tau$ has to be applied to the current state using e.g. the Euler [204] or the Heun method [193]. In [205], the general idea of minimizing the difference between the current parameterization and an explicitly improved wave function is introduced as supervised wave function optimization (SWO). The improved wave function, which constitutes the target in this optimization, can e.g. be obtained through power methods or imaginary time evolution.

Lastly, in [206] an optimization scheme based on stochastic representations of wave functions is proposed. In this representation, not the configurations, here in terms of particle positions $\{\boldsymbol{R}_i\}_{i=1,\ldots,N_s}$ of a continuous system, but a set of N_s samples $(\boldsymbol{R}_i, \psi_s(\boldsymbol{R}_i))$ is used for the optimization. The NQS is given by

$$\psi_s(\boldsymbol{R}_i) = e^{-\delta\tau\hat{H}} \hat{P}_{s/a} \psi_{\boldsymbol{\theta}}(\boldsymbol{R})\,|_{\boldsymbol{R}=\boldsymbol{R}_i}, \tag{33}$$

with a variational function $\psi_{\boldsymbol{\theta}}$ parameterized by a FFNN and stochastic projection $\hat{P}_{s/a}$ onto the symmetric or antisymmetric subspace. In order to train the NQS, first a set of samples $(\boldsymbol{R}_i, \psi_s(\boldsymbol{R}_i))$ is generated and projected onto the target subspace. Then, simple regression is applied, with the goal of minimizing the sum of squared residuals between the projected samples and $\hat{P}_{s/a}\psi_{\boldsymbol{\theta}}(\boldsymbol{R}_i)$. The updated trial function $\hat{P}_{s/a}\psi_{\boldsymbol{\theta}'}$ is then used to generate new sample coordinates $\boldsymbol{R}_i'$, and imaginary time evolution is performed on $\hat{P}_{s/a}\psi_{\boldsymbol{\theta}'}(\boldsymbol{R})|_{\boldsymbol{R}=\boldsymbol{R}_i'}$. In contrast to VMC, this method does not require that the samples are distributed according to the wave functions' amplitudes. Furthermore, no evaluation of the energy or its gradients is required.

One reason why the optimization is so challenging is the intricate interplay between phase and amplitude parts during the optimization. In some cases, the optimization outcomes are improved by imposing a certain sign structure of the target state, e.g. the Marshall sign rule for the Heisenberg model (restricted to bipartite

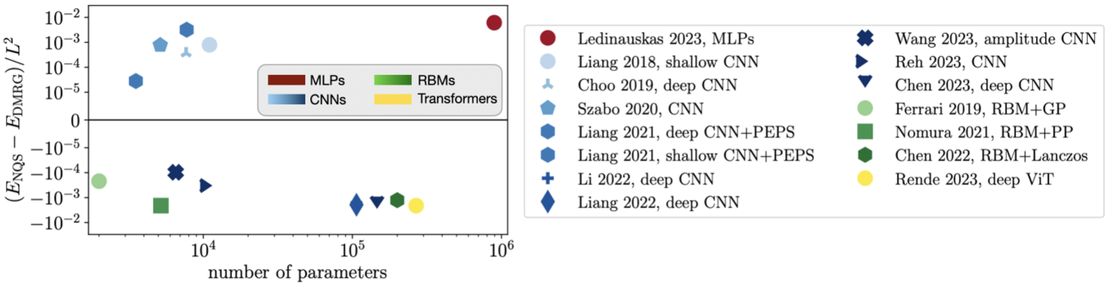

Figure 10. Comparison of energies obtained with NQS, E_{NQS}, and matrix-product states (MPS) using density matrix renormalization group (DMRG), E_{DMRG} with 8192 $SU(2)$ states [207], for ground states of the prototypical $J_1 - J_2$ model at $J_2/J_1 = 0.5$ on 10×10 lattices: We show energies obtained from FFNNs/MLPs in red (Ledinauskas *et al* (2023) [204]), CNNs in blue (Liang *et al* (2018) [80], Choo *et al* (2019) [81], Szabo *et al* (2020) [82], Liang *et al* (2021) [83], Li *et al* (2022) [84], Liang *et al* (2022) [85], Wang *et al* (2023) [86], Reh *et al* (2023) [45], Chen *et al* (2023) [87]), RBMs in green (Ferrari *et al* (2019) [60], Nomura *et al* (2021) [61], Chen *et al* (2022) [202]) and transformers in yellow (Rende *et al* (2023) [100]). The color coding denotes the appearance of the respective works. This plot is adapted from a table in [100].

lattices) [82, 110, 111]. Furthermore, the interplay between phase and amplitude during the optimization can be investigated by considering the partial optimization problem. In [35, 197], either the exact phases or the exact amplitudes are set and kept constant during the training, such that only amplitude *or* phase, respectively, have to be learned. In both works, considering the $J_1 - J_2$ model and the bosonic and fermionic $t - J$ model, the authors find that none of the two optimization strategies can systematically improve the ground state representation, and hence conclude that the interplay between phase and amplitude seems to play a crucial role for the optimization.

3.1.3. Results on the $J_1 - J_2$ model

A commonly used model for benchmarking new NQS architectures and comparing different optimization methods is the $J_1 - J_2$ model,

$$\mathcal{H}_{J_1-J_2} = J_1 \sum_{\langle i,j \rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j, \tag{34}$$

with spin-1/2 operators $\hat{\mathbf{S}}_i$ and $\langle i,j \rangle$ ($\langle\langle i,j \rangle\rangle$) denoting nearest (next-nearest) neighbors. In this model, the nearest neighbor J_1 and the next-nearest neighbor J_2 interactions compete. For $J_2 \to 0$ and $J_1 > 0$, the system reduces to a Heisenberg antiferromagnet, whereas for $J_1 \to 0$ and $J_2 > 0$ the system favors antiferromagnetic stripes. In the intermediate regime around $J_2/J_1 = 0.5$, the nature of the ground state is not clear yet, ranging from numerical results for gapped or gapless quantum spin liquids, different types of valence bond solids or both of them [61]. With the frustration controlled by the ratio J_2/J_1, the $J_1 - J_2$ model has become a paradigmatic model for the evaluation of the performance of NQS architectures [45, 60, 61, 80–87, 92, 100, 202, 204].

The results obtained with variants of FFNNs, CNNs, RBMs and transformers for the $J_1 - J_2$ model on 10×10 square lattices at $J_2/J_1 = 0.5$ are shown in figure 10. It can be seen that in recent years, results from CNNs, RBMs and Transformers have become competitive with or have even outperformed DMRG results obtained for a bond dimension of 8192 with implemented $SU(2)$ symmetry (corresponding to a bond dimension 32000 with only $U(1)$ symmetry). In particular, recent modifications of SR as in [87, 100] have allowed to use more than 10^5 parameters, systematically improving results obtained with smaller NQS with around $10^3 - 10^4$ parameters. However, also for $10^3 - 10^4$ parameters some works have obtained energy errors that are competitive with the NQS architectures using more than one order of magnitude more parameters [45, 61, 86]. It hence becomes evident that in principle many NQS architectures can achieve competitive results to conventional methods, but details on the implementation and the optimization procedure can have a significant impact on the performance. Consequently, architecture, its hyperparameters and optimization parameters have to be carefully chosen, but at the same time they can mostly only be determined by try and error.

3.2. Excited states

The methods for calculating excited states using NQS fall into two categories, depending on the type of excited states that are targeted: (i) Lowest energy states in a different symmetry sector than the ground state, e.g. different momentum or magnetization sectors. (ii) Low-energy states in the same symmetry sector as the ground state.

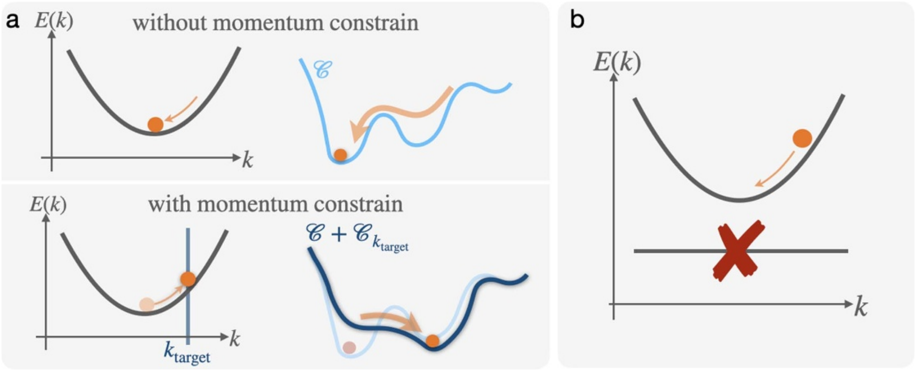
a
E(k)
without momentum constrain
k
$\mathscr{C}$
E(k)
with momentum constrain
$\mathscr{C} + \mathscr{C}_{k_{\mathrm{target}}}$
k_{target}
k
b
E(k)
k

Figure 11. Excited States: (a) Optimization for different momentum sectors. (b) Low-energy states in the same symmetry by enforcing orthogonality to the lower excited states and the ground state as in [53, 69]. Reproduced from [35]. CC BY 4.0.

The former usually rely on the same VMC scheme, but with a restriction to the targeted symmetry sector implemented in the wave function or the training loss [35, 53, 62, 131, 168, 208]. This enables e.g. the calculation of quasiparticle dispersions from NQS [35, 53, 62, 69, 131]:

In [62], excited states are targeted using quantum-number projections [209], i.e. for a NQS parameterization $\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ one defines the wave function using the total momentum $\boldsymbol{k}$ projection

$$\psi_{\boldsymbol{k}}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{R}} e^{-i\boldsymbol{k}\cdot\boldsymbol{R}} \psi_{\boldsymbol{\theta}}\left(\hat{T}_{\mathrm{R}}\boldsymbol{\sigma}\right) \tag{35}$$

with the translation operator $\hat{T}_{\boldsymbol{R}}$, shifting the system by the vector $\boldsymbol{R}$. This can be combined with other quantum-number projections, including e.g. spin parity and spatial symmetries, to improve the accuracy. In another approach [35], specific momentum $\boldsymbol{k}_{\mathrm{target}}$ sectors are targeted by adding a mean square error between $\boldsymbol{k}_{\mathrm{target}}$ and $\boldsymbol{k}_{\mathrm{NQS}}$, see figure 11(a). Hereby, the momentum of the NQS is given by

$$\boldsymbol{k}^{\mu}_{\mathrm{NQS}} = \frac{i}{a} \log\langle\psi_{\boldsymbol{\theta}}|\hat{T}_{\boldsymbol{e}_\mu}|\psi_{\boldsymbol{\theta}}\rangle, \tag{36}$$

with the translation operator $\hat{T}_{\boldsymbol{e}_\mu}$ in direction of the unit vector $\boldsymbol{e}_\mu$ by a lattice constant a. In contrast to MPS calculations, this global operator can be evaluated at low computational cost. Another approach to target excited states in different symmetry sectors consists in the use of group convolutional neural networks, as discussed in section 2.3.

For the second category, figure 11(b), usually more modifications need to be made. In [53, 69] excited states are targeted by enforcing orthogonality to the ground state ψ_0 or lower lying excited states ψ_i. In [69], the first excited state $\psi_{1,\boldsymbol{\theta}}$ is calculated by adding the normalized overlap with the ground state,

$$\frac{\langle\psi_{0,\boldsymbol{\theta}'}|\psi_{1,\boldsymbol{\theta}}\rangle}{\langle\psi_{0,\boldsymbol{\theta}'}|\psi_{0,\boldsymbol{\theta}'}\rangle\langle\psi_{1,\boldsymbol{\theta}}|\psi_{1,\boldsymbol{\theta}}\rangle}, \tag{37}$$

to the cost function. In [53], the excited state is defined as

$$\psi_1 := \Phi_{\boldsymbol{\theta}} - \lambda\psi_{0,\boldsymbol{\theta}'}. \tag{38}$$

To enforce orthogonality, $\langle\psi_1|\psi_{0,\boldsymbol{\theta}'}\rangle = 0$, they use $\lambda = \langle\Phi_{\boldsymbol{\theta}}/\psi_{0,\boldsymbol{\theta}'}\rangle$. Both methods require the explicit representation of the ground state $\psi_{0,\boldsymbol{\theta}'}$, rendering the calculation of higher excited states computationally demanding.

Another method, requiring no explicit orthogonalization of the different states, transforms the problem of finding the K lowest excited states of a given system into that of finding the ground state of an expanded system given by all targeted excited states [210]. The ansatz for the expanded ground state is written as

$$\Psi(\boldsymbol{x}) := \det\begin{pmatrix} \psi_1(\boldsymbol{x}^1) & \dots & \psi_K(\boldsymbol{x}^1) \\ \vdots & & \vdots \\ \psi_1(\boldsymbol{x}^K) & \dots & \psi_K(\boldsymbol{x}^K) \end{pmatrix}, \tag{39}$$

i.e. an unnormalized Slater determinant of many-particle wave functions ψ_i instead of single-particle orbitals. Here, $\boldsymbol{x}^i$ denotes a set of N particles $x^i_1,\ldots,x^i_N$. The Hamiltonian is correspondingly expanded to act

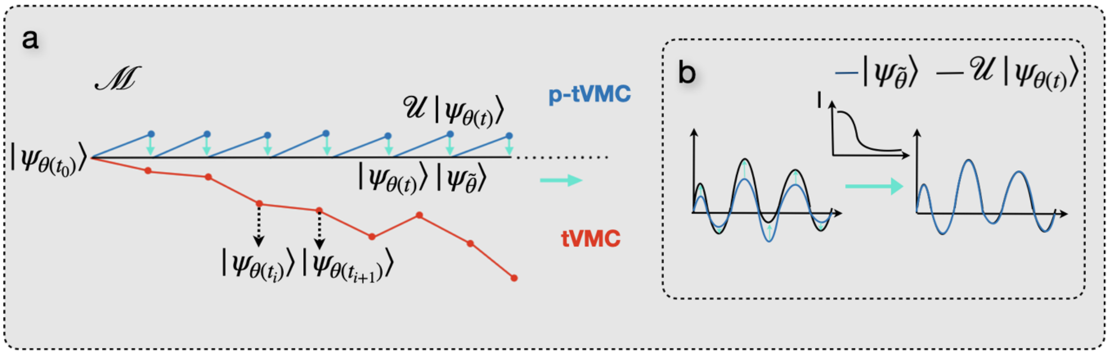
a
$\mathcal{M}$
$\mathscr{U}\,|\psi_{\theta(t)}\rangle$
p-tVMC
$|\psi_{\theta(t_0)}\rangle$
$|\psi_{\theta(t)}\rangle\,|\psi_{\tilde{\theta}}\rangle$
tVMC
$|\psi_{\theta(t_i)}\rangle\,|\psi_{\theta(t_{i+1})}\rangle$
b
$-|\psi_{\tilde{\theta}}\rangle\ -\mathscr{U}\,|\psi_{\theta(t)}\rangle$

Figure 12. (a) Adapted visualisation from [211] of the p-tVMC method. In the cases when the tVMC evolution of a state ψ breaks down, p-tVMC can resolve this problem by projecting the exactly evolved state onto the variational manifold $\mathcal{M}$. The tVMC evolution of the state (red) starts to break down whereas the p-tVMC method projects the exactly evolved state $U|\psi_{\boldsymbol{\theta}}\rangle$ back onto the variational manifold $\mathcal{M}$ (blue) at each timestep. (b) To project the exactly evolved state onto the variational wave function, the distance in the Hilbert space needs to be minimized, resulting in an optimization problem of the infidelity. Reproduced from [211]. CC BY 4.0.

on all K particle sets, and VMC is performed to find the ground state of the expanded system. Subsequently, energies, expectation values, and overlaps in the excited states can be retrieved from $\Psi(\boldsymbol{x})$.

Finally, in [167] the authors propose a method based on the assumption that one-particle excitations dominate the low-lying spectrum, allowing to construct the excited states by single-particle excitations on top of the ground state.

3.3. Dynamics

Neural network quantum states are furthermore capable of describing time-dependent systems. The quantum dynamics can be obtained using time-dependent network weights $\boldsymbol{\theta}(t)$ [15, 25]. Numerically exact results for timescales comparable to or exceeding the capabilities of TN algorithms have been obtained for the paradigmatic two-dimensional transverse-field Ising model [75]. However, even though NQS are able to capture strongly entangled states, the required number of parameters needed to represent a quantum state after a global quench can grow exponentially in time, according to [212]. This can potentially render the use of NQS to represent the dynamics of a state inefficient. Dynamically increasing the network size and choosing a network architecture incorporating the symmetry of the state are promising approaches to overcome this problem.

Following the Dirac-Frenkel Variational principle, the time derivative of the neural network weights are to be optimized such that the variational residuals

$$\mathrm{R}\left(\dot{\boldsymbol{\theta}}\left(t\right)\right) = \mathrm{dist}\left(\partial_t \psi_{\tilde{\boldsymbol{\theta}}}, -i\hat{H}\psi_{\boldsymbol{\theta}}\right) \tag{40}$$

are minimized [25, 213]. This is achieved within the stochastic approach using time-dependent Variational Monte Carlo method (t-VMC). In most cases t-VMC is used in combination with SR, see section 3.1. The iteration scheme to find the ground state energy can be interpreted as an effective imaginary time evolution, such that an iteration scheme to approximate the real-time evolution of the quantum spin system can be derived in an analogous way [70]. At each time step, $|\psi_{\boldsymbol{\theta}(t)}|^2$ is sampled and the variational residual is evaluated. Minimization of the variational residuals with respect to $\dot{\boldsymbol{\theta}}$ gives a first order differential equation for the weights $\boldsymbol{\theta}(t)$ [75]. The final equation to be solved for the variational parameters $\boldsymbol{\theta}$ is then given by the time-dependent variational principle (TDVP) equation:

$$S\left(t\right)\dot{\boldsymbol{\theta}}\left(t\right) = -iF\left(t\right), \tag{41}$$

with the covariance matrix S and the vector of forces F are the same as in section 3.1.

To solve this equation for the variational parameters, the covariance matrix needs to be inverted. Since this matrix can be non-invertible, S^{-1} denotes the Moore-Penrose pseudo-inverse. This inversion leads to various problems, see also section 3.1. Moreover, the unstable Moore-Penrose pseudo-inverse of the S matrix requires choosing a right cut-off tolerance for small singular values. In practice, one finds that the chosen cut-off for the pseudo-inverse of S can alter t-VMC. Krylov subspace methods (i.e. the conjugate gradient method or MINRES algorithm) avoid this sensitivity problem, but are not always converging [213]. When calculating the dynamics of a system, this error accumulates with each time step. [193, 213]

Regularization schemes for S have been developed, see section 3.1, but these still require a very large number of samples [213] and impact the accuracy [193]. The choice of regularization method impacts the stability of the dynamics [71]. In [75], this problem is approached by disregarding the contributions to the TDVP of which there is insufficient information available due to constraints of a finite number of samples N_{MC}. This still requires the diagonalization of S, which has a high computational cost ($O(N_{\boldsymbol{\theta}}^3)$) [213]. In [55], they adopt the minSR approach to reduce this computational cost to $O(N_s^3)$, as it allows to reshape S into a $N_s \times N_s$ matrix.

Successful applications of t-VMC have been obtained for system sizes up to $N = 256$ spins using an RBM and for time scales up till 10 s in [25, 76, 214]. In [77], an RBM ansatz is used to capture the dynamics of the 2D Heisenberg model, predicting strong magnon-magnon interactions leading to supermagnonic propagation. In [89] the dynamics of the 2D transverse-field Ising model around the quantum phase transition has been studied with t-VMC, leading to insights in the quantum Kibble-Zurek mechanism. Numerical simulations obtained with t-VMC have been used to validate experimental data interpretation, as is done in [119] for a network description of wave function snapshots. However, accessing long times via t-VMC stays challenging. The stability of t-VMC strongly depends on the chosen variational ansatz [193], and is affected by systematic statistical bias or an exponential sample complexity when the wave-function contains zeros. This is also the case for ground state calculations, but is less harmful due to the accumulation of error that affects dynamics [211].

A new method is proposed to circumvent these issues in [193] and [211], see figure 12: Projected time-dependent Variational Monte Carlo (p-tVMC). The scheme consists of casting a Runge-Kutta integration scheme into minimizing a variational distance at each time step. Starting again from the Dirac-Frenkel Variational Principle, an s-order Runge-Kutta approximant is to be used, instead of a first order expansion of the time propagator [193]. In practice, a metric based upon infidelity can now be used, in contrast to the normally used Fubini-Study distance. This infidelity can be estimated through Monte Carlo sampling and should be considered the distance in the Hilbert space between $U|\psi_{\boldsymbol{\theta}}\rangle$ and $|\psi_{\tilde{\boldsymbol{\theta}}}\rangle$, which is to be optimized [211]:

$$\min_{\tilde{\theta}} \mathrm{I}\left(|\psi_{\tilde{\boldsymbol{\theta}}}\rangle, U|\psi_{\boldsymbol{\theta}}\rangle\right). \tag{42}$$

No expansion of $|\psi_{\tilde{\boldsymbol{\theta}}}\rangle$ with respect to the variational parameters is necessary for p-tVMC, opposed to tVMC where a second order expansion leads to equation (41). However, the unitary time evolution $e^{-i\hat{H}\delta t}$ needs to be decomposed by the Trotter-Suzuki decomposition, leading to a scaling of the number of required samples with the Trotter order. This is costly and breaks translational invariance. Alternatively, an expansion of the time propagator into its Taylor decomposition is needed, typically up to second order in $\hat{H}$ in order to evaluate long time dynamics. This results in a quadratic increase of the connected matrix elements that need to be computed, and thereby the computational cost. Furthermore, for higher s-order integration schemes, the computational cost scales with the system size N as N^s [193]. In [211] the use of an RBM avoids this issue by computing the off-diagonal elements in the transverse field Ising model exactly. Because of this, the p-tVMC method only scales linearly with the number of parameters, which makes this a promising method to compute dynamics for large neural network architectures. As opposed to an update rule for the network parameters $\boldsymbol{\theta}$, standard gradient descent based techniques to minimize the infidelity can be used. Since p-tVMC is not affected by biases or vanishing SNR, it can simulate dynamics in cases where t-VMC fails or is inefficient [211].

An alternative approach, using the implicit midpoint method, has been explored in [213]. Here, the network parameters are optimized to minimize the error between the state at the next time step $t + \Delta t$ and the discrete flow of the implicit midpoint method applied to the Schrödinger equation. This has shown some advantages in preserving the symplectic form of Hamiltonian dynamics while not complicating the network optimization with intermediate quantities [213].

In [215], a dynamical strong disorder renormalization group approach is used to map the quantum dynamics of a disordered spin chain onto a quantum circuit generated by local unitaries. These local unitaries are applied to the NQS in a supervised scheme, similar to the SWO discussed in section 3.1 and the infidelity minimization in p-tVMC.

Other methods to calculate the dynamics of a system consist of training with time evolved states that have been exactly calculated using ED, such that the evolution of new initial states can be predicted by a neural network without evolving the wave function explicitly with the Hamiltonian [216]. To speed up the simulation of the dynamics of many-body systems, hybrid methods are used such as using neural quantum states with calculations on quantum devices to determine expectation values with high computational cost [214]. Another approach are *variational classical networks* [217–219], i.e. efficient and perturbatively controlled representations of (time-evolved) wave functions in terms of classical spins, where the latter are used to construct a NQS representation of the state under consideration.

3.3.1. Spectral functions

Most of the work discussed so far focus on global quenches. Time-dependent NQS can however also be used to simulate local quenches, such as the response of a system to a local perturbation, relevant for spectral functions. In [88], the dynamical spin structure factor of different two-dimensional quantum Ising models is calculated by applying t-VMC following a local perturbation (application of $\hat{S}^z$ operator) on top of the ground state represented by a convolutional neural network. Subsequent Fourier transformation yields the momentum- and frequency-resolved structure factor. A complementary approach is demonstrated in [220]. Here, the dynamical structure factor of the one- and two-dimensional Heisenberg model is calculated explicitly using a Chebyshev expansion, where the corresponding wave functions are represented as RBMs. In [221], the Green's function

$$G_{ij}(z) = \langle\psi|\hat{A}_i^\dagger \frac{1}{z-\hat{H}}\hat{A}_j|\psi\rangle \tag{43}$$

is directly calculated by an extension of the SR approach to obtain the *correction vector*

$$|\chi_j(z)\rangle = \frac{1}{z-\hat{H}}\hat{A}_j|\psi\rangle\,, \tag{44}$$

where the corresponding ground state $|\psi\rangle$ of the system has been obtained beforehand using SR.

3.4. Finite temperature states

In many experimentally relevant situations, we are dealing with quantum many-body systems at a finite temperature, and in order to compute thermodynamics properties of the system one needs to work with the thermal density matrix

$$\hat{\rho} = \frac{1}{Z}e^{-\beta\hat{H}} \tag{45}$$

where $\hat{H}$ is the Hamiltonian of the system, $\beta = \frac{1}{k_B T}$ is the inverse temperature, and k_B is the Boltzmann constant. Hence, the task boils down to evaluating this density matrix efficiently. One approach that has been developed and is commonly applied in the context of MPS is the idea of purification, also known as the thermofield approach [2, 146, 222]. In the purification method, an additional auxiliary site is introduced for each physical site of the system, known as an ancilla. As a result, one deals with a pure instead of a mixed state, where said pure state lives in a higher dimensional Hilbert space. The algorithm then starts from an infinite temperature state, followed by imaginary time evolution to cool down the system to the desired temperature, where imaginary time τ here denotes the inverse temperature β. The desired thermal density matrix is then obtained by tracing out the auxiliary degrees of freedom a, i.e.

$$\hat{\rho}(\boldsymbol{\sigma},\boldsymbol{\sigma}') = \sum_a \langle\boldsymbol{\sigma},a|\psi\rangle\langle\psi|\boldsymbol{\sigma}',a\rangle. \tag{46}$$

In the context of neural network quantum states, one approach to purification is through a modified RBM, see figure 13(a). A similar type of architecture was also used in [223] to reconstruct mixed states. [191, 224] employ the purification method to obtain finite temperature expectation values for a Heisenberg chain and a 6 × 6 $J_1 - J_2$ model. On top of the imaginary time evolution, [225] deals with real-time evolution, leveraging an RNN architecture. &percentage;For thermal states, the system's configuration is not static but evolves according to thermal fluctuations. RNNs can model this evolution by considering each spin's state in relation to its predecessors, allowing for an accurate representation of the system's thermal dynamics without simplifying the complex interactions between particles.

Another promising approach also developed in the context of tensor networks is the idea of minimally entangled typical thermal states (METTS) [226, 227]. METTS is designed to efficiently sample from the thermal ensemble instead of dealing with the full complexity of the mixed state directly. The idea is to construct an ensemble of pure states, which provides a good approximation of the thermal equilibrium state. Concretely, the trace in the evaluation of finite temperature expectation values can be expanded in terms of an orthonormal basis as

$$\langle\hat{O}\rangle = \mathrm{Tr}\left(\hat{\rho}\hat{O}\right) = \frac{1}{Z}\sum_i \langle\boldsymbol{\sigma}_i|e^{-\beta\hat{H}/2}\hat{O}e^{-\beta\hat{H}/2}|\boldsymbol{\sigma}_i\rangle = \frac{1}{Z}\sum_i P(i)\langle\psi_{\boldsymbol{\sigma}_i}(\beta)|\hat{O}|\psi_{\boldsymbol{\sigma}_i}(\beta)\rangle\,. \tag{47}$$

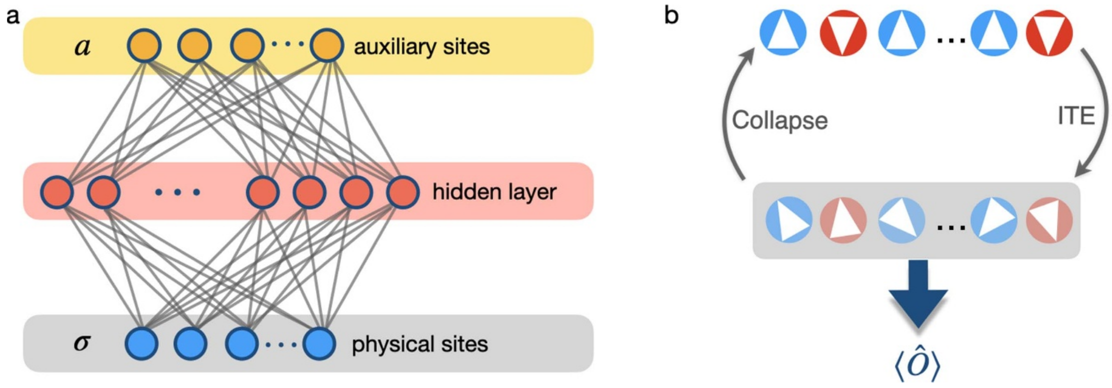

Figure 13. Finite temperature state simulations: (a) In the purification method, ancilla sites a are introduced in addition to the physical sites σ, see equation (46). The mixed state of the system is obtained by integrating out the auxiliary sites. (b) METTS: Starting from a pure infinite temperature state (top), the state is evolved in imaginary time (ITE) to produce a METT state (bottom). Next, a projective measurement (collapse) to a random basis state is done in order to obtain a new pure state. The expectation value of desired observables is calculated through the ensemble of METT states.

To this end, one starts from a pure product state $|\boldsymbol{\sigma}_0\rangle$. This product state is evolved in imaginary time to generate a state $|\psi_{\boldsymbol{\sigma}_i}(\beta)\rangle = e^{-\beta\hat{H}/2}|\boldsymbol{\sigma}_i\rangle$. This procedure gives us a so-called METTS state $|\psi_{\boldsymbol{\sigma}}(\beta)\rangle$. After this step, a projective measurement in the computational basis (collapse) is performed in order to produce a new pure state $|\boldsymbol{\sigma}_\mathbf{1}\rangle$ to start over with imaginary time evolution. This procedure of sampling the states $|\boldsymbol{\sigma}_i\rangle$ ensures that the resulting states represent the thermal ensemble accurately [227]. At the end, we have a set of states $\{|\psi_{\boldsymbol{\sigma}_0}(\beta)\rangle, |\psi_{\boldsymbol{\sigma}_1}(\beta)\rangle, \ldots, |\psi_{\boldsymbol{\sigma}_n}(\beta)\rangle\}$ from which the thermal average of a given operator $\hat{O}$ can be estimated as:

$$\langle \hat{O} \rangle_\beta = \frac{1}{N_s}\sum_{i=1}^{n}\langle \psi_{\boldsymbol{\sigma}_i}(\beta)|\hat{O}|\psi_{\boldsymbol{\sigma}_i}(\beta)\rangle, \tag{48}$$

where N_s is the number of METTS state samples. In [191, 228], the product states $|\boldsymbol{\sigma}_0\rangle$ are prepared by adjusting the parameters of an RBM correspondingly.

For the imaginary time evolution employed both in the purification and the METTS algorithm, the following equation must be solved:

$$\frac{\partial}{\partial\beta}|\psi(\beta)\rangle = -\frac{1}{2}\hat{H}|\psi(\beta)\rangle, \tag{49}$$

where the new wave function after each imaginary time step $\delta\tau$ must stay within the variational manifold. This constraint can lead to a modification of β.

Another approach to simulate finite temperature states is based on quantum typicality [191] which utilizes the concept that a single pure state can accurately reproduce the expectation values of an observable in the Gibbs ensemble for large systems. This method approximates an infinite temperature state using a combination of a pair product (PP) wave function Φ_{PP} and a neural network component $\psi_{\boldsymbol{\theta}}$, i.e.

$$\Psi(\boldsymbol{\sigma}) = \psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})\,\Phi_{\mathrm{PP}}(\boldsymbol{\sigma}). \tag{50}$$

Pair product wave functions can model electron interactions within the system, including the prohibition of double occupancy through the use of the Gutzwiller projection. The typical state is then evolved in imaginary time to simulate finite temperatures.

In [229], a CNN with two input channels $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ is used to represent a mixed state $\hat{\rho}(\boldsymbol{\sigma},\boldsymbol{\sigma}')$ of a one-dimensional bosonic system. Starting from an infinite temperature state, imaginary time evolution is performed, such that the output of the network is the corresponding matrix elements of density matrix at the desired temperature. As opposed to e.g. purification, this approach does not guarantee the hermiticity and positive definiteness of the density matrix.

In contrast to the works discussed so far, which all use imaginary time evolution, a recent paper [230] instead minimizes a modified free energy. Here, the von-Neumann entropy is replaced by the second Rényi entropy, which can be evaluated fairly efficiently. The optimization of neural network parameters is guided by the goal of minimizing this approximation to the free energy.

3.5. Open systems

The state of an open quantum system is described by its density operator $\hat{\rho}$. This makes the simulation of open systems even more challenging than for closed systems, since for density matrices, the curse of dimensionality is even more pronounced as for wave functions, e.g. for a system of N spin-$1/2$ particles the number of coefficients to parameterize $\hat{\rho}$ scales as 4^N coefficients [114]. The dynamics of open quantum systems is governed by the Lindblad master equation,

$$\dot{\hat{\rho}} = -i\left[\hat{H},\hat{\rho}\right] + \sum_i \gamma_i \left(\left(\hat{L}^i\right)^\dagger \hat{\rho}\hat{L}^i - \frac{1}{2}\left\{\left(\hat{L}^i\right)^\dagger \hat{L}^i,\hat{\rho}\right\}\right) \tag{51}$$

where $[\dots](\{\dots\})$ denote (anti-)commutators and $\hat{L}$ are so-called jump operators. The first term describes the unitary dynamics of the system given by $\hat{H}$, the second the non-unitary dynamics due to the dissipation to the environment with strength γ. Equation (51) can also be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\rho} = \hat{\mathcal{L}}\hat{\rho} \tag{52}$$

with the Liouvillian $\hat{\mathcal{L}}$.

In most cases, the second form is used to determine the solution $\hat{\rho}$ using NQS. In order to do so, a neural representation of the density operator is needed. This is typically realized by (i) using positive operator valued measures (POVMs) [145] or (ii) introducing additional nodes that encode the mixing to the environment [223, 231]. We would like to point out that the works discussed here in the context of (i) and (ii) are inherently different from machine learning approaches to open quantum systems that e.g. aim learn a parameterization of $\hat{\mathcal{L}}$ and not $\hat{\rho}$ as e.g. in [232, 233].

In the POVM approach [145], the density matrix is represented by a probability distribution over measurement outcomes a of an informationally complete (IC) set of measurement operators $\hat{M}_a$, inspired from Born's rule

$$P_{\boldsymbol{\theta}}(a) = \mathrm{Tr}\left(\hat{M}_a \hat{\rho}_{\mathrm{POVM}}\right). \tag{53}$$

This leads to the definition

$$\hat{\rho}_{\mathrm{POVM}} = \sum_{a,a'} P_{\boldsymbol{\theta}}(a)\, T^{-1}_{a,a'} \hat{M}_{a'}, \tag{54}$$

with the overlap matrix $T_{a,a'} = \mathrm{Tr}\hat{M}_a\hat{M}_{a'}$ and different possible choices of $\hat{M}$. The advantage of the IC-POVM representation is that in equation (54) only the positive amplitudes $P_{\boldsymbol{\theta}}(a)$ have to be modeled by a neural network. However, $\hat{\rho}_{\mathrm{POVM}}$ is, in general, not a positive-definite matrix. This problem does not occur in the purification ansatz [234].

The second approach is e.g. taken in [231], where an RBM with an additional hidden layer is used, and both visible and hidden state representations are split into two representations for rows and columns of the density matrix $\hat{\rho}(\boldsymbol{\sigma},\boldsymbol{\sigma}')$, see figure 14. Other neural density operators exist, e.g. in terms of CNNs [235, 236] or in form of an autoregressive network, see [234]. In the latter work, the density matrix is defined as

$$\hat{\rho}(\boldsymbol{\sigma},\boldsymbol{\sigma}') = \prod_{i=1}^{N}\sum_{a=1}^{R} \psi_{\boldsymbol{\sigma}_{\leqslant i},a}\left(\psi_{\boldsymbol{\sigma}'_{\leqslant i},a}\right)^*, \tag{55}$$

with ancillas a and neural network representations of $\psi_{\boldsymbol{\sigma}_{\leqslant i},a}$. This is an example of the purification approach discussed in section 3.4. In contrast to the POVM approach, this purification via the ancilla nodes in equation (55) makes the density matrix positive semi-definite. Furthermore, each factor $\sum_{a=1}^{R}\psi_{\boldsymbol{\sigma}_{\leqslant i},a}(\psi_{\boldsymbol{\sigma}'_{\leqslant i},a})^*$ in equation (55) can be normalized, making the neural network representation of $\hat{\rho}$ autoregressive.

With these ansätze (i) and (ii), the solution of equation (52) can be obtained using different approaches:

3.5.1. Time dependent solution of the Lindblad equation

Equation (52) can be solved directly by minimizing $||\frac{\mathrm{d}}{\mathrm{d}t}\hat{\rho} - \hat{\mathcal{L}}_{\boldsymbol{\theta}}||$ using SR, where $||\dots||$ can e.g. be taken to be the Fubini-Study distance or the trace norm [231, 237]. This is done e.g. in [231] using an RBM with additional nodes to simulate a 1D anisotropic Heisenberg model or in [237] using a deep (quantum) FFNN for a dissipative 1D TFIM and 2D $J_1 - J_2$ spin systems. For systems that lack translational invariance, more elaborate sampling and optimization procedures are necessary [238, 239]. In [105, 114, 240] the POVM

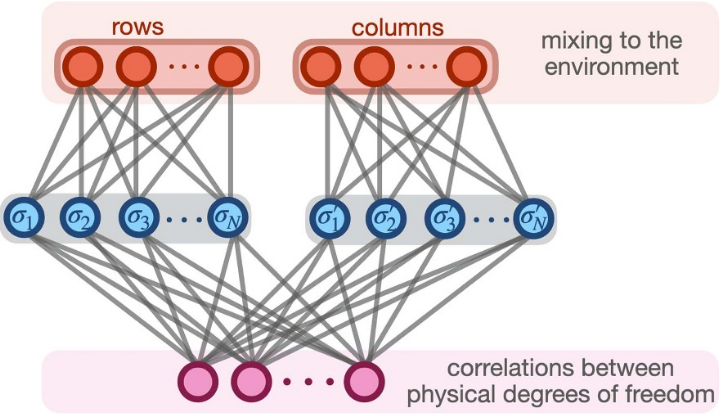
rows
columns
mixing to the environment
σ_1 σ_2 σ_3 σ_N
σ'_1 σ'_2 σ'_3 σ'_N
correlations between physical degrees of freedom

Figure 14. Neural density operator based on a RBM as used in [231]: The state of the physical system is encoded by the visible (blue) layer, with correlations encoded in the bottom hidden layer (purple) and mixing due to the environment encoded in the top hidden layer (red). The left represents row, the right part column indices of the density matrix $\hat{\rho}(\boldsymbol{\sigma}, \boldsymbol{\sigma}')$.

ansatz implemented with autoregressive networks is used for the time dependent solution of 1D and 2D dissipative Heisenberg models and prototypical states from quantum computing. In order to do so, the time evolution has to be represented in the stochastic representation (54), i.e. an operator $\hat{O}$ is calculated that time evolves $P_{\boldsymbol{\theta}_t}(a)$. Then, the parameters $\boldsymbol{\theta}_{t+1}$ are selected such that the distance between $\hat{O}P_{\boldsymbol{\theta}_t}(a)$ and $P_{\boldsymbol{\theta}_{t+1}}(a)$ is minimal. In [105, 240], this distance is calculated explicitly, e.g. in [240] the the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(\hat{O}P_{\boldsymbol{\theta}_t}(a)|P_{\boldsymbol{\theta}_{t+1}}(a))$, see equation (57), is minimized. In [105], the network parameters are optimized to minimize the error between the new, time evolved state and the target state given by the discrete flow according to a second-order forward-backward trapezoid method applied to the Lindblad equation.

In [114], the distance is measured by the KL or the Hellinger distance, but in this work the distance metrics are expanded around small times, leading to the the time dependent variational principle update, see equation (32), for $P_{\boldsymbol{\theta}_t}(a)$. This reduces the sampling cost and makes the optimization problem convex.

3.5.2. Steady states

Other works use the fact that stationary states $\hat{\rho}_{ss}$ in open systems fulfill

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\rho}_{ss} = \hat{\mathcal{L}}\hat{\rho}_{ss} = 0. \tag{56}$$

The neural density operator is trained to fulfill this condition by minimizing e.g. the expectation value of $\hat{\mathcal{L}}$ [241, 242] or the L_2-norm [236, 243]. Furthermore, $\hat{\mathcal{L}}^\dagger$ can be applied from the left to equation (56), yielding an optimization problem of $\hat{\mathcal{L}}^\dagger\hat{\mathcal{L}}$ instead of $\hat{\mathcal{L}}$, with the advantage that the former operator is hermitean and hence has a real spectrum [244]. In [242], the dynamics of a 2D dissipative XYZ spin model is simulated using an RBM with additional nodes. [243, 244] consider 2D transverse-field Ising models and other similar spin systems.

4. Learning from data

4.1. Quantum state tomography

QST, i.e. the reconstruction of a quantum state from measurement data, plays a crucial role for the characterization and verification of quantum devices [245]. For example, it can be applied to compare the experimentally prepared state against the target state to estimate the error of the quantum device under consideration, see figure 15. Furthermore, QST enables the evaluation of complex observables that would not be accessible directly from experiments [36].

Full QST relies on two assumptions: (*i*) Since typically several measurements are needed to infer the quantum state, it is assumed that identical copies of the state can be prepared from which the measurements can be taken. (*ii*) The set of measurements, described by POVMs, is IC and hence the probability distribution over measurement outcomes uniquely determine the quantum state via Born's rule. Since these conditions are not fulfilled in most cases, approximate QST schemes are necessary.

Conventional methods for QST, such as linear inversion and maximum likelihood estimation [246, 247], are based on inverting Born's rule and hence suffer from an exponential scaling with the system size, resulting from an exponential growth of both the sampling complexity and the number of parameters needed to

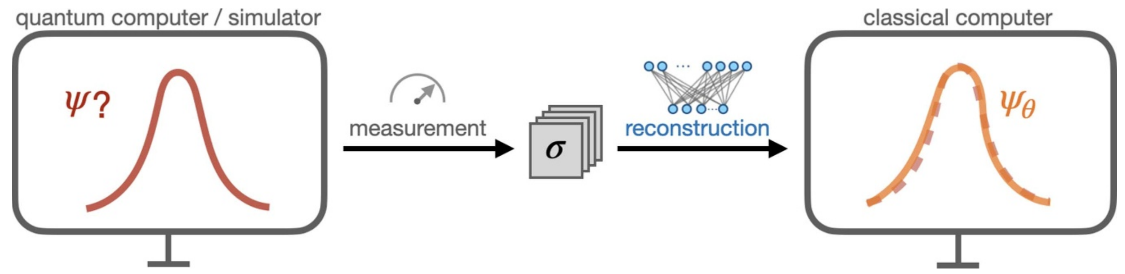

Figure 15. Quantum state tomography: Reconstruction of an unknown quantum state ψ on a quantum device, by taking measurements $\boldsymbol{\sigma}$ and reconstruction with a neural network. The network is trained such that the amplitudes of the reconstructed state $\psi_{\boldsymbol{\theta}}$ are as close as possible to the real measurement distribution generated from ψ, see equation (57).

represent the state. Under these aspects, machine learning techniques have enormous potential for QST: Firstly, machine learning models can learn the structure of a state under consideration, i.e. symmetries or correlations, allowing them to efficiently represent typical physical states with a reduced number of parameters [40]. Furthermore, they have the ability to generalize from an incomplete dataset, tackling the exponential scaling of the sample complexity [248]. In [249], the authors show that a simple FFNN can outperform conventional methods both in terms of reconstruction time and quality.

The potential of neural quantum states for QST has been explored for various pure and mixed quantum states. One of the first works reconstructs finite temperature states of the 1D and 2D Ising model using real-valued RBMs [250]. Furthermore, highly entangled states with more than a hundred qubits are reconstructed in [124] using a complex RBM. In these works, the NQS is trained by minimizing the KL divergence between the measurement distribution q and the NQS amplitudes $p_{\boldsymbol{\theta}}$,

$$D_{\mathrm{KL}}\left(q|p_{\boldsymbol{\theta}}\right) = \sum_{\boldsymbol{\sigma}} q\left(\boldsymbol{\sigma}\right) \log\left(\frac{q\left(\boldsymbol{\sigma}\right)}{p_{\boldsymbol{\theta}}\left(\boldsymbol{\sigma}\right)}\right) \tag{57}$$

on the underlying dataset $\mathcal{D}$ with measurements $\boldsymbol{\sigma} \in \mathcal{D}$. In [251], instead of D_{KL} the classical shadow formalism (see below) is used to approximate the infidelity between target and reconstructed state.

This procedure assumes pure quantum states, which is typically not the case in experimental settings. In some cases, pure representations can nevertheless be used to approximately represent the states under consideration and effect of measurement errors in the training data can be mitigated by modifications of the NQS architecture, as shown in [252] for data from 1D Rydberg tweezer arrays using an RBM with an additional noise layer. However, in many cases, the reconstruction of the full density matrix is needed. Requirements on the density matrix, such as positivity, can be enforced using a purification scheme with additional latent units [223]. Other works considering mixed state representations involve generative models, neural density operators and POVM representatins [145, 253, 254], see section 3.5. In [255], an iterative scheme to promote any pure state reconstruction to a mixed state reconstruction is proposed. The reconstruction performance can be increased by filtering the experimental data [256] or constraining the density matrix, e.g. to positivity or global symmetries, improve the performance, in particular in the presence of measurement imperfections [112, 257].

To further reduce the amount of measurement data needed for the reconstruction, an efficient evaluation of the typically incomplete set of measurements is needed. In [258], the authors present an efficient method for constructing approximate classical descriptions of quantum states from very few measurements, so-called classical shadows, with an information-theoretic bound for the precision of estimated expectation values. The procedure relies on randomly selecting unitaries $\hat{U}$ from an ensemble of particular unitaries that allow one to calculate

$$\mathcal{M}\left(\hat{\rho}\right) = \mathbb{E}\left[\hat{U}^{\dagger}|\sigma\rangle\langle\sigma|\hat{U}\right] \tag{58}$$

from measurements in the computational basis $|\sigma\rangle$. Here, $\mathbb{E}\left[\ldots\right]$ denotes the average over both the choice of unitary and the outcome distribution. The density matrix $\hat{\rho}$ can be approximated by the so-called classical shadow $\mathcal{S}$ consisting of N_s samples, with

$$\mathcal{S} = \left\{\hat{\rho}_i = \mathcal{M}^{-1}\left(\hat{U}_i^{\dagger}|\sigma\rangle_i\langle\sigma|_i\hat{U}_i\right) \quad \text{with} \quad i \in 1,\ldots,N_s\right\}. \tag{59}$$

Further works consider the effect of local measurements [259] or an efficient choice of measurement configurations [208, 260]. Hereby, adaptive schemes, that incorporate the knowledge gained from previous

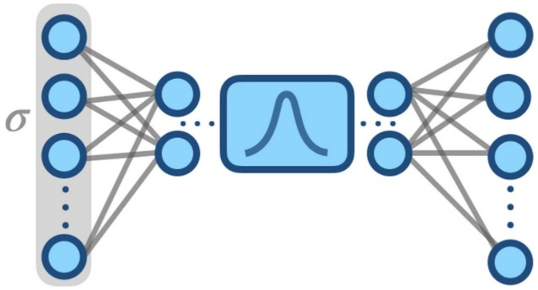

Figure 16. Variational autoencoder consisting of an encoder, the latent space or bottleneck layer, and a decoder. The latent space representation is a multivariate Gaussian, i.e. the encoder generates the values for mean and variances of the Gaussian distribution. By sampling from this Gaussian, the input for the decoder layer is generated.

measurements to propose the next measurement configuration, are of great interest [208, 261]. Moreover, NQS can be pretrained with artificial data before the measurements to enhance the reconstruction [262].

Also in the setting of QST, the choice of NQS depends on the state under consideration, with potential advantages e.g. of autoregressive networks and their perfect sampling [109] or of networks which can represent a high degree of entanglement [104, 263]. Typical architectures are RBMs [37, 124, 250, 264] (see section 2.2), RNNs [37, 112] (see section 2.5) and transformer networks [104, 263, 265] (see section 2.6) and CNNs [109] (see section 2.3). Furthermore, latent space representations like variational autoencoders are used [266–268]. Since these architecture has not been introduced yet, we we shortly describe autoencoders and their application to quantum state reconstruction in the following.

4.1.1. Latent space representations

Latent space representations consist of an encoder, the latent space or bottleneck layer, and a decoder, see figure 16. The encoder, typically several fully connected or convolutional layers, compresses the input into just a few nodes in the bottleneck layer. The decoder subsequently generates an output based on the information in the bottleneck layer. The network parameters are optimized such that the generated output is as close to the input as possible.

In a variational autoencoder [269], the encoder generates the values for mean and variances in the bottleneck layer, and the values used as input for the decoder are then sampled from a multivariate Gaussian.

Variational autoencoders have been used in [266–268] for (quantum) state reconstruction, where the input consists of the measured data, which the autoencoder learns to compress and de-compress using encoder and decoder. After training, the encoder can be dropped, and by sampling random numbers as input to the latent space, new, uncorrelated samples can be generated.

In [266], this approach is used to reconstruct positive wave functions, i.e. effectively, the probability distribution of the samples in the computational basis is learned. The efficiency of the compression is quantified by the ratio of the number of network parameters to the Hilbert space dimension. In this case, the size of the latent space is always chosen to correspond to the system size. [267] uses a conditional variational autoencoder to perform state reconstruction of ground states of the 1D transverse field Ising model based on IC positive-operator valued measures. The magnetic field h is the *condition*, which is used as additional input to the decoder. Furthermore, the autoencoder representation of the quantum state is appealing since its latent space contains information on the state under consideration: In [268], the low-dimensional latent space representation of finite temperature samples of the 2D Ising model is used to extract physical features. The authors of [270] determine the minimal size of the latent space needed to reproduce local observations to measure the local complexity of time-evolved states.

4.2. Hybrid training

The ground state search described in section 3.1 typically starts from an NQS with randomly initialized parameters $\boldsymbol{\theta}_0$. Assuming convergence of the variational Monte Carlo procedure, the details of this initialization should not matter. However, if the ground state search is challenging, e.g. due to local minima, the question of convergence itself, as well as how many iterations are needed to reach the ground state can crucially depend on the choice of $\boldsymbol{\theta}_0$. Using existing data, e.g. from other numerical simulations or an experimental realization, the initial parameters can be chosen such that the ground state search starts from a highly promising region of the parameter space. In this case, the data is used to perform state reconstruction as described in section 4.1. Subsequently, the parameters of the same neural quantum state are variationally minimized to find the ground state.

In an experiment, the ground state is typically not perfectly realized. The combination of experimental data from a state close to the ground state with a subsequent numerical ground state search can yield better results than either approach exhibits on its own, as demonstrated for large, interacting two-dimensional Rydberg atom arrays using quantum Monte Carlo data in [271] and using experimental snapshots from large two-dimensional Rydberg atom arrays [272] in [117]. This is done by minimizing the KL divergence between the measurement distribution q and the NQS amplitudes $p_{\boldsymbol{\theta}} = |\psi_{\boldsymbol{\theta}}|^2$, see equation (57). In both cases, a recurrent neural network was used to represent the quantum state. In these examples, the Hamiltonian under consideration is stoquastic, and thus a wave function with real coefficients can represent its ground state. This means in particular that measurements in the computational basis are sufficient for a faithful state reconstruction.

In [273], molecular Hamiltonians for LiH and H_2, as well as the one-dimensional lattice Schwinger model are considered. The Hamiltonians under consideration are not stoquastic, and thus measurements in different bases are necessary to faithfully reconstruct their ground states. The computational cost for a single iteration in the reconstruction training is 2^K, where K is the number of sites on which measurements outside of the computational basis (i.e. in the x- or y-basis) are performed. In order to constrain this computational cost, measurements with only a few rotations out of the z-basis are used for state reconstruction. The ground state is prepared using the variational quantum eigensolver on IBM quantum devices, based on superconducting qubits, as well as using numerical simulations. Again, the results show that this hybrid approach improves the numerical and experimental results to yield lower errors for the ground state energy. Moreover, accurate estimations of more complex observables, such as the entanglement entropy, are directly possible without requiring additional quantum resources.

The problem of the exponential cost for a reconstruction based on measurements from configurations away from the computational basis is overcome in [103] by training on observables like spin-spin correlations instead of the snapshots in the rotated basis. In order to do so, instead of the KL divergence, a variant of the mean-square error loss is used for the rotated basis. This loss function does not incorporate information on the measurement statistics, but allows to compensate for systematic errors that are e.g. present in experimental data, for example by explicitly applying spatial symmetries when calculating the experimental observables. Furthermore, for the computational basis [103] compares the KL divergence to the Wasserstein (or earth movers distance), with the advantage of the latter that it can incorporate information on the energy to the data-driven pretraining.

In [274] samples from a quantum state produced by the variational quantum eigensolver, a quantum algorithm which generates the ground state on a quantum device, are used. In contrast to the other works discussed in this context, the experimental samples directly replace the samples needed in the VMC, i.e. at the beginning of the training these samples are used to calculate the expectation values for the VMC optimization instead of samples generated from the trial function. The authors argue that this gives a better estimation of the expectation values at the beginning of the training, speeding up the convergence. After this first stage, the usual VMC algorithm is used. The method is applied for trial wave functions in form of an NQS and the Gutzwiller wave function to find the ground state of the 1D TFIM and Fermi-Hubbard model, respectively.

5. Summary and outlook

In this review, we discuss NQS, i.e. variational wave functions that are represented by neural networks. The extensive study of NQS since their proposal in 2017 [25], has revealed that the strengths of neural networks—namely their great expressive power, their ability to compress information very efficiently and their capability to generalize from a given dataset—turn out to be extremely helpful for the simulation of quantum systems:

On the one hand, their expressivity permits the representation of a broad range of quantum states, including a variety of (frustrated) spin systems as well as bosonic and fermionic quantum many-body states in one, two and even three dimensions that we review in this article. The limits of this expressive power are still topic of current research. Second, the efficiency of NQS allows to compress the exponential number of wave function coefficients w.r.t. the system size into a tractable number of network parameters, competitive with state of the art numerical methods. This makes NQS very versatile and applicable in many different contexts in the field of numerical simulation of quantum systems and QST. However, finding the targeted state in the huge and complicated optimization landscape with many local minima remains one of the main challenges of the NQS approach, and the optimization depends sensitively on the choice of architecture, hyperparameters and the specific optimization strategies, as we discuss in this review. Advanced training strategies, among them SR which incorporates the geometric structure of the loss landscape, are crucial to overcome this problem. For example, recent results on spin systems obtained with a modified version of SR even start to reach numerical precision in terms of the obtained ground state energies [87]. Furthermore, hybrid approaches, allowing to choose a good starting point of the optimization obtained from training on experimental or numerical data [117, 271] or from the initialization of certain NQS from TNs [49], allow to combine the advantages of existing methods and NQS and are hence promising to overcome this challenge. Finally, the capacity of neural networks to generalize from the training data makes NQS an excellent platform for exploring innovative ideas that go beyond existing methods, such as the creation of toolboxes for simulating entire phase diagrams rather than individual states [57, 98]. Harnessed with these strengths and versatility, neural quantum states offer a new and promising perspective on the challenges posed by simulating quantum many-body systems.

Data availability statement

No new data were created or analysed in this study.

Acknowledgments

We gratefully acknowledge discussions with Annika Böhler, Fabian Döschl, Fabian Grusdt, Juan Carrasquilla, Lukas Vetter, Markus Schmitt, Moritz Reh, Roeland Wiersema, Roger Melko, Schuyler Moss, Simon Linsel, Stefanie Czischek, and Tizian Blatz. We acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC-2111-390814868 and within Project-ID 314695032—SFB 1277. HL acknowledges support by the International Max Planck Research School.

ORCID iDs

Hannah Lange https://orcid.org/0000-0002-0051-2087
Anka Van de Walle https://orcid.org/0009-0008-6478-4509
Atiye Abedinnia https://orcid.org/0009-0008-6625-6590
Annabelle Bohrdt https://orcid.org/0000-0002-3339-5200

References

[1] Orús R 2014 *Ann. Phys., NY* **349** 117–58
[2] Zwolak M and Vidal G 2004 *Phys. Rev. Lett.* **93** 207205
[3] Evenbly G and Vidal G 2014 *Phys. Rev. Lett.* **112** 240502
[4] Vidal G 2007 *Phys. Rev. Lett.* **99** 220405
[5] Shi Y Y, Duan L M and Vidal G 2006 *Phys. Rev.* A **74** 022320
[6] Cincio L, Dziarmaga J and Rams M M 2008 *Phys. Rev. Lett.* **100** 240603
[7] Verstraete F and Cirac J I 2004 Renormalization algorithms for quantum-many body systems in two and higher dimensions (arXiv:cond-mat/0407066)
[8] Klümper A, Schadschneider A and Zittartz J 1993 *Europhys. Lett.* **24** 293
[9] Schollwöck U 2011 *Ann. Phys., NY* **326** 96–192
[10] White S R 1992 *Phys. Rev. Lett.* **69** 2863–6
[11] Schollwöck U 2005 *Rev. Mod. Phys.* **77** 259–315
[12] Hastings M B 2007 *J. Stat. Mech.* 08024
[13] Eisert J, Cramer M and Plenio M B 2010 *Rev. Mod. Phys.* **82** 277–306
[14] Orús R 2019 *Nat. Rev. Phys.* **1** 538–50
[15] Becca F and Sorella S 2017 *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press)
[16] Ceperley D and Alder B 1986 *Science* **231** 555–60
[17] Foulkes W M C, Mitas L, Needs R J and Rajagopal G 2001 *Rev. Mod. Phys.* **73** 33–83
[18] Pan G and Meng Z Y 2024 The sign problem in quantum monte carlo simulations *Encyclopedia of Condensed Matter Physics* 2nd edn, ed T Chakraborty (Academic) pp 879–93
[19] Troyer M and Wiese U J 2005 *Phys. Rev. Lett.* **94** 170201
[20] Carrasquilla J 2020 *Adv. Phys.* X **5** 1797528
[21] Cybenko G 1989 *Math. Control Signals Syst.* **2** 303–14
[22] Hornik K 1991 *Neural Netw.* **4** 251–7
[23] Kim T and Adalí T 2003 *Neural Comput.* **15** 1641–66
[24] Le Roux N and Bengio Y 2008 *Neural Comput* **20** 1631–49
[25] Carleo G and Troyer M 2017 *Science* **355** 602–6
[26] Sharir O, Shashua A and Carleo G 2022 *Phys. Rev.* B **106** 205136
[27] Deng D L, Li X and Das Sarma S 2017 *Phys. Rev.* X **7** 021021
[28] Gao X and Duan L M 2017 *Nat. Commun.* **8** 662
[29] Denis Z, Sinibaldi A and Carleo G 2023 Comment on "can neural quantum states learn volume-law ground states?" (arXiv:2309.11534)
[30] Levine Y, Sharir O, Cohen N and Shashua A 2019 *Phys. Rev. Lett.* **122** 065301
[31] Lu S, Gao X and Duan L M 2019 *Phys. Rev.* B **99** 155136
[32] Luo D, Chen Z, Hu K, Zhao Z, Hur V M and Clark B K 2023 Gauge-invariant and anyonic-symmetric autoregressive neural network for quantum lattice models (available at: https://link.aps.org/doi/10.1103/PhysRevResearch.5.013216)

[33] Huang Y and Moore J E 2021 *Phys. Rev. Lett.* **127** 170601
[34] Sharir O, Levine Y, Wies N, Carleo G and Shashua A 2020 *Phys. Rev. Lett.* **124** 020503
[35] Lange H, Döschl F, Carrasquilla J and Bohrdt A 2023 Neural network approach to quasiparticle dispersions in doped antiferromagnets (arXiv:2310.08578)
[36] Torlai G, Mazzola G, Carleo G and Mezzacapo A 2020 *Phys. Rev. Res.* **2** 022060
[37] Iouchtchenko D, Gonthier J F, Perdomo-Ortiz A and Melko R G 2023 *Mach. Learn.: Sci. Technol.* **4** 015016
[38] Dawid A *et al* 2022 Modern applications of machine learning in quantum sciences (arXiv:2204.04198)
[39] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborova L 2019 *Rev. Mod. Phys.* **91** 045002
[40] Carrasquilla J and Torlai G 2021 *PRX Quantum* **2** 040201
[41] Melko R G and Carrasquilla J 2024 *Nat. Comput. Sci.* **4** 11–18
[42] Jia Z A, Yi B, Zhai R, Wu Y C, Guo G C and Guo G P 2019 *Adv. Quantum Technol.* **2** 1800077
[43] Yang Y, Cao H and Zhang Z 2019 *Sci. China Phys. Mech. Astron.* **63** 210312
[44] Vivas D R, Madroñero J, Bucheli V, Gómez L O and Reina J H 2022 Neural-network quantum states: a systematic review (arXiv:2204.12966)
[45] Reh M, Schmitt M and Gärttner M 2023 *Phys. Rev.* B **107** 195115
[46] Medvidović M and Moreno J R 2024 Neural-network quantum states for many-body physics (arXiv:2402.11014)
[47] Chen J, Cheng S, Xie H, Wang L and Xiang T 2018 *Phys. Rev.* B **97** 085104
[48] Glasser I, Pancotti N, August M, Rodriguez I D and Cirac J I 2018 *Phys. Rev.* X **8** 011006
[49] Wu D, Rossi R, Vicentini F and Carleo G 2023 *Phys. Rev. Res.* **5** L032001
[50] Passetti G, Hofmann D, Neitemeier P, Grunwald L, Sentef M A and Kennes D M 2023 *Phys. Rev. Lett.* **131** 036502
[51] Chen Z, Newhouse L, Chen E, Luo D and Soljačić M 2023 Antn: bridging autoregressive neural networks and tensor networks for quantum many-body simulation (arXiv:2304.01996)
[52] Van Den Nest M 2011 *Quantum Inf. Comput* **11** 784–812
[53] Choo K, Carleo G, Regnault N and Neupert T 2018 *Phys. Rev. Lett.* **121** 167204
[54] Cai Z and Liu J 2018 *Phys. Rev.* B **97** 035116
[55] Zhang W, Xing B, Xu X and Poletti D 2024 arXiv:2406.03381
[56] Çeven K, Oktel M O and Keleş A 2022 *Phys. Rev.* A **106** 063320
[57] Zhu Z, Mattheakis M, Pan W and Kaxiras E 2023 *Phys. Rev. Res.* **5** 043084
[58] Saito H and Kato M 2018 *J. Phys. Soc. Japan* **87** 014001
[59] Westerhout T, Astrakhantsev N, Tikhonov K S, Katsnelson M I and Bagrov A A 2020 *Nat. Commun.* **11** 1593
[60] Ferrari F, Becca F and Carrasquilla J 2019 *Phys. Rev.* B **100** 125131
[61] Nomura Y and Imada M 2021 *Phys. Rev.* X **11** 031034
[62] Nomura Y 2021 *J. Phys.: Condens. Matter* **33** 174003
[63] Vieijra T, Casert C, Nys J, De Neve W, Haegeman J, Ryckebusch J and Verstraete F 2020 *Phys. Rev. Lett.* **124** 097201
[64] Vieijra T and Nys J 2021 *Phys. Rev.* B **104** 045123
[65] Li C X, Yang S and Xu J B 2021 *Sci. Rep.* **11** 16667
[66] Deng D L, Li X and Das Sarma S 2017 *Phys. Rev.* B **96** 195145
[67] Clark S R 2018 *J. Phys. A: Math. Theor.* **51** 135301
[68] Kaubruegger R, Pastori L and Budich J C 2018 *Phys. Rev.* B **97** 195136
[69] Valenti A, Greplova E, Lindner N H and Huber S D 2022 *Phys. Rev. Res.* **4** L012010
[70] Czischek S, Gärttner M and Gasenzer T 2018 *Phys. Rev.* B **98** 024311
[71] Hofmann D, Fabiani G, Mentink J H, Carleo G and Sentef M A 2022 *SciPost Phys.* **12** 165
[72] Saito H 2017 *J. Phys. Soc. Japan* **86** 093001
[73] McBrian K, Carleo G and Khatami E 2019 *J. Phys.: Conf. Ser.* **1290** 012005
[74] Vargas-Calderón V, Vinck-Posada H and González F A 2020 *J. Phys. Soc. Japan* **89** 094002
[75] Schmitt M and Heyl M 2020 *Phys. Rev. Lett.* **125** 100503
[76] Fabiani G and Mentink J H 2019 *SciPost Phys.* **7** 004
[77] Fabiani G, Bouman M D and Mentink J H 2021 *Phys. Rev. Lett.* **127** 097202
[78] Nomura Y, Darmawan A S, Yamaji Y and Imada M 2017 *Phys. Rev.* B **96** 205152
[79] Xia R and Kais S 2018 *Nat. Commun.* **9** 4195
[80] Liang X, Liu W Y, Lin P Z, Guo G C, Zhang Y S and He L 2018 *Phys. Rev.* B **98** 104426
[81] Choo K, Neupert T and Carleo G 2019 *Phys. Rev.* B **100** 125124
[82] Szabó A and Castelnovo C 2020 *Phys. Rev. Res.* **2** 033075
[83] Liang X, Dong S J and He L 2021 *Phys. Rev.* B **103** 035138
[84] Li M, Chen J, Xiao Q, Wang F, Jiang Q, Zhao X, Lin R, An H, Liang X and He L 2022 *IEEE Trans. Parallel Distrib. Syst.* **33** 2846–59
[85] Liang X, Li M, Xiao Q, Chen J, Yang C, An H and He L 2023 *Mach. Learn.: Sci. Technol.* **4** 015035
[86] Wang J Q, He R Q and Lu Z Y 2023 Variational optimization of the amplitude of neural-network quantum many-body ground states (arXiv:2308.09664)
[87] Chen A and Heyl M 2023 Efficient optimization of deep neural quantum states toward machine precision *Nat. Phys.* **20** 1476–81
[88] Mendes-Santos T, Schmitt M and Heyl M 2023 *Phys. Rev. Lett.* **131** 046501
[89] Schmitt M, Rams M M, Dziarmaga J, Heyl M and Zurek W H 2022 *Sci. Adv.* **8** eabl6850
[90] Fu C, Zhang X, Zhang H, Ling H, Xu S and Ji S 2022 Lattice convolutional networks for learning ground states of quantum many-body systems (arXiv:2206.07370)
[91] Roth C and MacDonald A H 2021 Group convolutional neural networks improve quantum state accuracy (arXiv:2104.05085)
[92] Roth C, Szabó A and MacDonald A H 2023 *Phys. Rev.* B **108** 054410
[93] Duric T, Chung J H, Yang B and Sengupta P 2024 Spin-1/2 kagome heisenberg antiferromagnet: Machine learning discovery of the spinon pair density wave ground state (arXiv:2401.02866)
[94] Beck J, Bodky J, Motruk J, Müller T, Thomale R and Ghosh P 2024 Phase diagram of the j-j_d heisenberg model on the maple-leaf lattice: Neural networks and density matrix renormalization group (arXiv:2401.04995)
[95] Kochkov D, Pfaff T, Sanchez-Gonzalez A, Battaglia P and Clark B K 2021 Learning ground states of quantum hamiltonians with graph networks (arXiv:2110.06390)
[96] Yang L, Hu W and Li L 2020 Scalable variational monte carlo with graph neural ansatz (arXiv:2011.12453)
[97] Viteritti L L, Rende R and Becca F 2023 *Phys. Rev. Lett.* **130** 236401

[98] Zhang Y H and Di Ventra M 2023 *Phys. Rev.* B **107** 075147

[99] Rende R, Gerace F, Laio A and Goldt S 2023 Optimal inference of a generalised Potts model by single-layer transformers with factored attention (arXiv:2304.07235 [cond-mat, stat])

[100] Rende R, Viteritti L L, Bardone L, Becca F and Goldt S 2023 A simple linear algebra identity to optimize large-scale neural network quantum states (arXiv:2310.05715 [cond-mat])

[101] Sprague K and Czischek S 2023 Variational monte carlo with large patched transformers (arXiv:2306.03921)

[102] Fitzek D, Teoh Y H, Fung H P, Dagnew G A, Merali E, Moss M S, MacLellan B and Melko R G 2024 arXiv:2405.21052

[103] Lange H, Bornet G, Emperauger G, Chen C, Lahaye T, Kienle S, Browaeys A and Bohrdt A 2024 Transformer neural networks and quantum simulators: a hybrid approach for simulating strongly correlated systems (arXiv:2406.00091)

[104] Cha P, Ginsparg P, Wu F, Carrasquilla J, McMahon P L and Kim E A 2021 *Mach. Learn.: Sci. Technol.* **3** 01LT01

[105] Luo D, Chen Z, Carrasquilla J and Clark B K 2022 *Phys. Rev. Lett.* **128** 090501

[106] von Glehn I, Spencer J S and Pfau D 2023 A self-attention ansatz for ab-initio quantum chemistry (arXiv:2211.13672)

[107] Shang H, Guo C, Wu Y, Li Z and Yang J 2023 Solving schrödinger equation with a language model (arXiv:2307.09343)

[108] Wu Y, Guo C, Fan Y, Zhou P and Shang H 2023 Nnqs-transformer: an efficient and scalable neural network quantum states approach for ab initio quantum chemistry (arXiv:2306.16705)

[109] Schmale T, Reh M and Gärttner M 2022 *npj Quantum Inf.* **8** 115

[110] Hibat-Allah M, Ganahl M, Hayward L E, Melko R G and Carrasquilla J 2020 *Phys. Rev. Res.* **2** 23358

[111] Roth C 2020 Iterative retraining of quantum spin models using recurrent neural networks (arXiv:2003.06228 [cond-mat, physics:physics])

[112] Morawetz S, Vlugt I J S D, Carrasquilla J and Melko R G 2021 *Phys. Rev.* A **104** 12401

[113] Hibat-Allah M, Inack E M, Wiersema R and Carrasquilla J 2021 *Nat. Mach. Intell.* **3** 2522–5839

[114] Reh M, Schmitt M and Gärttner M 2021 *Phys. Rev. Lett.* **127** 230501

[115] Hibat-Allah M, Melko R G and Carrasquilla J 2023 *Phys. Rev.* B **108** 075152

[116] Döschl F, Palm F A, Lange H, Grusdt F and Bohrdt A 2024 Neural network quantum states for the interacting hofstadter model with higher local occupations and long-range interactions (arXiv:2405.04472)

[117] Moss M S, Ebadi S, Wang T T, Semeghini G, Bohrdt A, Lukin M D and Melko R G 2023 Enhancing variational Monte Carlo using a programmable quantum simulator (arXiv:2308.02647 [cond-mat, physics:quant-ph])

[118] Malyshev A, Arrazola J M and Lvovsky A I 2023 Autoregressive neural quantum states with quantum number symmetries (arXiv:2310.04166)

[119] Mendes-Santos T *et al* 2024 *Phys. Rev.* X **14** 021029

[120] Mehta P, Bukov M, Wang C H, Day A G, Richardson C, Fisher C K and Schwab D J 2019 *Phys. Rep.* **810** 1–124

[121] Teng P 2018 *Phys. Rev.* E **98** 033305

[122] Hopfield J J 1982 *Proc. Natl Acad. Sci. USA* **79** 2554–8

[123] Melko R G, Carleo G, Carrasquilla J and Cirac J I 2019 *Nat. Phys.* **15** 887–92

[124] Torlai G, Mazzola G, Carrasquilla J, Troyer M, Melko R and Carleo G 2018 *Nat. Phys.* **14** 447–50

[125] Aoki K I and Kobayashi T 2016 *Mod. Phys. Lett.* B **30** 1650401

[126] Liu D, Ran S J, Wittek P, Peng C, García R B, Su G and Lewenstein M 2019 *New J. Phys.* **21** 073059

[127] Gan W C and Shu F W 2017 *Int. J. Mod. Phys.* D **26** 1743020

[128] Borin A and Abanin D A 2020 *Phys. Rev.* B **101** 95141

[129] Pastori L, Kaubruegger R and Budich J C 2019 *Phys. Rev.* B **99** 165123

[130] Sfondrini A, Cerrillo J, Schuch N and Cirac J I 2010 *Phys. Rev.* B **81** 214426

[131] Nomura Y 2020 *J. Phys. Soc. Japan* **89** 054706

[132] Alcalde Puente D and Eremin I M 2020 *Phys. Rev.* B **102** 195148

[133] Karthik V and Medhi A 2024 Convolutional restricted boltzmann machine (crbm) correlated variational wave function for the hubbard model on a square lattice: Mott metal-insulator transition (arXiv:2402.02794)

[134] Bronstein M M, Bruna J, Cohen T and Veličković P 2021 Geometric deep learning: Grids, groups, graphs, geodesics, and gauges (arXiv:2104.13478)

[135] Kipf T N and Welling M 2017 Semi-supervised classification with graph convolutional networks (arXiv:1609.02907)

[136] Li Y, Tarlow D, Brockschmidt M and Zemel R 2017 Gated graph sequence neural networks (arXiv:1511.05493)

[137] Pescia G, Nys J, Kim J, Lovato A and Carleo G 2023 Message-passing neural quantum states for the homogeneous electron gas (arXiv:2305.07240)

[138] Kim J, Pescia G, Fore B, Nys J, Carleo G, Gandolfi S, Hjorth-Jensen M and Lovato A 2023 Neural-network quantum states for ultra-cold fermi gases (arXiv:2305.08831)

[139] Luo D, Dai D D and Fu L 2023 Pairing-based graph neural network for simulating quantum materials (arXiv:2311.02143)

[140] Bortone M, Rath Y and Booth G H 2023 Impact of conditional modelling for universal autoregressive quantum states (arXiv:2306.05917)

[141] Steffen, Maximilian Z H G S A and Udluft 2006 Learning long term dependencies with recurrent neural networks *Artificial Neural Networks - Icann 2006*, ed W D Andreas, D O E K S Stafylopatis (Springer) pp 71–80

[142] Chung J, Gulcehre C, Cho K and Bengio Y 2014 Empirical evaluation of gated recurrent neural networks on sequence modeling (arXiv:1412.3555)

[143] Graves A, Fernandez S and Schmidhuber J 2007 Multi-dimensional recurrent neural networks (arXiv:0705.2011)

[144] Hibat-Allah M, Melko R G and Carrasquilla J 2022 Supplementing recurrent neural network wave functions with symmetry and annealing to improve accuracy (arXiv:2207.14314 [cond-mat, physics:physics])

[145] Carrasquilla J, Torlai G, Melko R G and Aolita L 2019 *Nat. Mach. Intell.* **1** 155–61

[146] Verstraete F, García-Ripoll J J and Cirac J I 2004 *Phys. Rev. Lett.* **93** 207204

[147] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł and Polosukhin I 2017 *Advances in Neural Information Processing Systems* vol 30

[148] Rende R and Viteritti L L 2024 Are queries and keys always relevant? a case study on transformer wave functions (arXiv:2405.18874)

[149] Viteritti L L, Rende R, Parola A, Goldt S and Becca F 2024 Transformer wave function for the shastry-sutherland model: emergence of a spin-liquid phase (arXiv:2311.16889)

[150] Pfau D, Spencer J, deG Matthews A and Foulkes W 2020 *Phys. Rev. Res.* **2** 033429

[151] Hermann J, Schätzle Z and Noé F 2020 *Nat. Chem.* **12** 1755–4349

[152] Schätzle Z, Hermann J and Noé F 2021 *J. Chem. Phys.* **154** 124108
[153] Han J, Zhang L and Weinan E 2019 *J. Comput. Phys.* **399** 108929
[154] Pang T, Yan S and Lin M 2022 $o(n^2)$ universal antisymmetry in fermionic neural networks (arXiv:2205.13205)
[155] Stokes J, Moreno J R, Pnevmatikakis E A and Carleo G 2020 *Phys. Rev.* B **102** 205122
[156] Liu Z and Clark B K 2023 A unifying view of fermionic neural network quantum states: From neural network backflow to hidden fermion determinant states (arXiv:2311.09450)
[157] Moreno J R, Carleo G, Georges A and Stokes J 2022 *Proc. Natl Acad. Sci.* **119** e2122059119
[158] Gauvin-Ndiaye C, Tindall J, Moreno J R and Georges A 2023 Mott transition and volume law entanglement with neural quantum states (arXiv:2311.05749)
[159] Feynman R P and Cohen M 1956 *Phys. Rev.* **102** 1189–204
[160] Luo D and Clark B K 2019 *Phys. Rev. Lett.* **122** 226401
[161] Romero I, Nys J and Carleo G 2024 Spectroscopy of two-dimensional interacting lattice electrons using symmetry-aware neural backflow transformations (arXiv:2406.09077)
[162] Li X, Qian Y, Ren W, Xu Y and Chen J 2024 Emergent wigner phases in moiré superlattice from deep learning (arXiv:2406.11134)
[163] Luo D, Dai D D and Fu L 2024 Simulating moiré quantum matter with neural network (arXiv:2406.17645)
[164] Humeniuk S, Wan Y and Wang L 2022 Autoregressive neural slater-jastrow ansatz for variational monte carlo simulation (arXiv:2210.05871)
[165] Barrett T D, Malyshev A and Lvovsky A I 2022 *Nat. Mach. Intell.* **4** 351–8
[166] Inui K, Kato Y and Motome Y 2021 *Phys. Rev. Res.* **3** 043126
[167] Yoshioka N, Mizukami W and Nori F 2021 *Commun. Phys.* **4** 2399–3650
[168] Viteritti L L, Ferrari F and Becca F 2022 *SciPost Phys.* **12** 166
[169] Luo D, Carleo G, Clark B K and Stokes J 2021 *Phys. Rev. Lett.* **127** 276402
[170] Sehayek D, Golubeva A, Albergo M S, Kulchytskyy B, Torlai G and Melko R G 2019 *Phys. Rev.* B **100** 195125
[171] Golubeva A and Melko R G 2022 *Phys. Rev.* B **105** 125124
[172] Dash S, Vicentini F, Ferrero M and Georges A 2024 Efficiency of neural quantum states in light of the quantum geometric tensor (arXiv:2402.01565)
[173] Klassert R, Baumbach A, Petrovici M A and Gärttner M 2022 *iScience* **25** 104707
[174] Czischek S *et al* 2022 *SciPost Phys.* **12** 039
[175] Czischek S, Pawlowski J M, Gasenzer T and Gärttner M 2019 *Phys. Rev.* B **100** 195120
[176] Vicentini F *et al* 2022 *SciPost Phys. Codebases* 7
[177] Sinibaldi A and Vicentini F 2023 Netket fidelity package (available at: https://github.com/netket/netket_fidelity)
[178] Schmitt M and Reh M 2022 *SciPost Phys. Codebases* 2
[179] Bradbury J, Frostig R, Hawkins P, Johnson M J, Leary C, Maclaurin D, Necula G, Paszke A, VanderPlas J, Wanderman-Milne S and Zhang Q 2018 JAX: composable transformations of Python+NumPy programs (available at: http://github.com/google/jax)
[180] Beach M J S, Vlugt I D, Golubeva A, Huembeli P, Kulchytskyy B, Luo X, Melko R G, Merali E and Torlai G 2019 *SciPost Phys.* **7** 009
[181] McMillan W L 1965 *Phys. Rev.* **138** A442–51
[182] Huang L and Wang L 2017 *Phys. Rev.* B **95** 035105
[183] Assaraf R and Caffarel M 1999 *Phys. Rev. Lett.* **83** 4682–5
[184] Kingma D P and Ba J 2017 Adam: A method for stochastic optimization (arXiv:1412.6980)
[185] Loshchilov I and Hutter F 2019 Decoupled weight decay regularization (arXiv:1711.05101)
[186] Sorella S 1998 *Phys. Rev. Lett.* **80** 4558–61
[187] Sorella S 2001 *Phys. Rev.* B **64** 024512
[188] Amari S and Douglas S 1998 Why natural gradient? *Proc. 1998 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP'98* vol 2 pp 1213–6
[189] Amari S, Karakida R and Oizumi M 2019 Fisher information and natural gradient learning in random deep networks *Proc. 22nd Int. Conf. on Artificial Intelligence and Statistics* (*Proc. of Machine Learning Research* vol 89), ed K Chaudhuri and M Sugiyama (PMLR) pp 694–702
[190] Hackl L, Guaita T, Shi T, Haegeman J, Demler E and Cirac J I 2020 *SciPost Phys.* **9** 048
[191] Wagner D, Klümper A and Sirker J 2024 arXiv:2311.13799
[192] Park C Y and Kastoryano M J 2020 *Phys. Rev. Res.* **2** 023232
[193] Donatella K, Denis Z, Le B A and Ciuti C 2023 *Phys. Rev.* A **108** 022210
[194] Zhao X *et al* 2022 Ai for quantum mechanics: High performance quantum many-body simulations via deep learning *SC22: Int. Conf. for High Performance Computing, Networking, Storage and Analysis* pp 1–15
[195] Choi S C T and Saunders M A 2014 *ACM Trans. Math. Softw.* **40** 1–12
[196] Zhang W, Xu X, Wu Z, Balachandran V and Poletti D 2023 *Phys. Rev.* B **107** 165149
[197] Bukov M, Schmitt M and Dupont M 2021 *SciPost Phys.* **10** 147
[198] Inack E M, Morawetz S and Melko R G 2022 *Condensed Matter* **7** 38
[199] Khandoker S A, Abedin J M and Hibat-Allah M 2023 *Mach. Learn.: Sci. Technol.* **4** 15026
[200] Zen R, My L, Tan R, Hébert F, Gattobigio M, Miniatura C, Poletti D and Bressan S 2020 *Phys. Rev.* E **101** 053301
[201] Efthymiou S, Beach M J S and Melko R G 2019 *Phys. Rev.* B **99** 075113
[202] Chen H, Hendry D, Weinberg P and Feiguin A E 2022 Systematic improvement of neural network quantum states using a lanczos recursion (arXiv:2206.14307)
[203] Giuliani C, Vicentini F, Rossi R and Carleo G 2023 *Quantum* **7** 1096
[204] Ledinauskas E and Anisimovas E 2023 *SciPost Phys.* **15** 229
[205] Kochkov D and Clark B K 2018 Variational optimization in the ai era: computational graph states and supervised wave-function optimization (arXiv:1811.12423)
[206] Atanasova H, Bernheimer L and Cohen G 2023 *Nat. Commun.* **14** 3601
[207] Gong S S, Zhu W, Sheng D N, Motrunich O I and Fisher M P A 2014 *Phys. Rev. Lett.* **113** 027201
[208] Lange H, Kebrič M, Buser M, Schollwöck U, Grusdt F and Bohrdt A 2023 *Quantum* **7** 1129
[209] Mizusaki T and Imada M 2004 *Phys. Rev.* B **69** 125110
[210] Pfau D, Axelrod S, Sutterud H, von Glehn I and Spencer J S 2023 Natural quantum monte carlo computation of excited states (arXiv:2308.16848)
[211] Sinibaldi A, Giuliani C, Carleo G and Vicentini F 2023 (arXiv:2305.14294)

[212] Lin S H and Pollmann F 2022 *Phys. Status Solidi* b **259** 2100172
[213] Gutiérrez I L and Mendl C B 2022 *Quantum* **6** 627
[214] Lee C K, Patil P, Zhang S and Hsieh C Y 2021 *Phys. Rev. Res.* **3** 023095
[215] Burau H and Heyl M 2021 *Phys. Rev. Lett.* **127** 050601
[216] Zhang Z W, Yang S, Wu Y H, Liu C X, Han Y M, Lee C H, Sun Z, Li G J and Zhang X 2020 *Chin. Phys. Lett.* **37** 018401
[217] Schmitt M and Heyl M 2018 *SciPost Phys.* **4** 013
[218] Verdel R, Schmitt M, Huang Y P, Karpov P and Heyl M 2021 *Phys. Rev.* B **103** 165103
[219] Karpov P, Verdel R, Huang Y P, Schmitt M and Heyl M 2021 *Phys. Rev. Lett.* **126** 130401
[220] Hendry D, Chen H, Weinberg P and Feiguin A E 2021 *Phys. Rev.* B **104** 205130
[221] Hendry D and Feiguin A E 2019 *Phys. Rev.* B **100** 245123
[222] Feiguin A E and White S R 2005 *Phys. Rev.* B **72** 220401
[223] Torlai G and Melko R G 2018 *Phys. Rev. Lett.* **120** 240503
[224] Nomura Y, Yoshioka N and Nori F 2021 *Phys. Rev. Lett.* **127** 060601
[225] Nys J, Denis Z and Carleo G 2024 *Phys. Rev.* B **109** 235120
[226] White S R 2009 *Phys. Rev. Lett.* **102** 190601
[227] Stoudenmire E M and White S R 2010 *New J. Phys.* **12** 055026
[228] Hendry D, Chen H and Feiguin A 2022 *Phys. Rev.* B **106** 165111
[229] Irikura N and Saito H 2020 *Phys. Rev. Res.* **2** 013284
[230] Lu S, Giudice G and Cirac J I 2024 Variational neural and tensor network approximations of thermal states (arXiv:2401.14243)
[231] Hartmann M J and Carleo G 2019 *Phys. Rev. Lett.* **122** 250502
[232] Mazza P P, Zietlow D, Carollo F, Andergassen S, Martius G and Lesanovsky I 2021 *Phys. Rev. Res.* **3** 023084
[233] Carnazza F, Carollo F, Zietlow D, Andergassen S, Martius G and Lesanovsky I 2022 *New J. Phys.* **24** 073033
[234] Vicentini F, Rossi R and Carleo G 2022 Positive-definite parametrization of mixed quantum states with deep neural networks (arXiv:2206.13488)
[235] Herrera Rodríguez L E and Kananenka A A 2021 *J. Phys. Chem. Lett.* **12** 2476–83
[236] Mellak J, Arrigoni E and von der Linden W 2024 Deep neural networks as variational solutions for correlated open quantum systems (arXiv:2401.14179)
[237] Liu Z, Duan L M and Deng D L 2022 *Phys. Rev. Res.* **4** 013097
[238] Kaestle O and Carmele A 2021 *Phys. Rev.* B **103** 195420
[239] Mellak J, Arrigoni E, Pock T and von der Linden W 2023 *Phys. Rev.* B **107** 205102
[240] Carrasquilla J, Luo D, Pérez F, Milsted A, Clark B K, Volkovs M and Aolita L 2021 *Phys. Rev.* A **104** 032610
[241] Weimer H 2015 *Phys. Rev. Lett.* **114** 040402
[242] Nagy A and Savona V 2019 *Phys. Rev. Lett.* **122** 250501
[243] Vicentini F, Biella A, Regnault N and Ciuti C 2019 *Phys. Rev. Lett.* **122** 250503
[244] Yoshioka N and Hamazaki R 2019 *Phys. Rev.* B **99** 214306
[245] Cramer M, Plenio M B, Flammia S T, Somma R, Gross D, Bartlett S D, Landon-Cardinal O, Poulin D and Liu Y K 2010 *Nat. Commun.* **1** 149
[246] Häffner H *et al* 2005 *Nature* **438** 643–6
[247] Hradil Z 1997 *Phys. Rev.* A **55** R1561–4
[248] Lohani S, Kirby B T, Brodsky M, Danaci O and Glasser R T 2020 *Mach. Learn.: Sci. Technol.* **1** 035007
[249] Koutný D, Motka L, Hradil Z C V, Řeháček J and Sánchez-Soto L L 2022 *Phys. Rev.* A **106** 012409
[250] Torlai G and Melko R G 2016 *Phys. Rev.* B **94** 165134
[251] Wei V, Coish W A, Ronagh P and Muschik C A 2023 Neural-shadow quantum state tomography (arXiv:2305.01078)
[252] Torlai G *et al* 2019 *Phys. Rev. Lett.* **123** 230504
[253] Torlai G and Melko R G 2020 *Annu. Rev. Condens. Matter Phys.* **11** 325–44
[254] Zhao H, Carleo G and Vicentini F 2023 Empirical sample complexity of neural network mixed state reconstruction (arXiv:2307.01840)
[255] Melkani A, Gneiting C and Nori F 2020 *Phys. Rev.* A **102** 022412
[256] Palmieri A M, Kovlakov E, Bianchi F, Yudin D, Straupe S, Biamonte J D and Kulik S 2020 *npj Quantum Inf.* **6** 20
[257] Neugebauer M, Fischer L, Jäger A, Czischek S, Jochim S, Weidemüller M and Gärttner M 2020 *Phys. Rev.* A **102** 042604
[258] Huang H Y, Kueng R and Preskill J 2020 *Nat. Phys.* **16** 1050–7
[259] Xin T, Lu S, Cao N, Anikeeva G, Lu D, Li J, Long G and Zeng B 2019 *npj Quantum Inf.* **5** 109
[260] Smith A W R, Gray J and Kim M S 2021 *PRX Quantum* **2** 020348
[261] Quek Y, Fort S and Ng H K 2021 Adaptive quantum state tomography with neural networks (https://doi.org/10.1038/s41534-021-00436-9)
[262] Zhu Y, Wu Y D, Bai G, Wang D S, Wang Y and Chiribella G 2022 *Nat. Commun.* **13** 6222
[263] Ma H, Sun Z, Dong D, Chen C and Rabitz H 2023 Attention-based transformer networks for quantum state tomography (arXiv:2305.05433)
[264] Tiunov E S, Tiunova V, Ulanov A E, Lvovsky A and Fedorov A K 2020 *Optica* **7** 448–54
[265] Zhong L, Guo C and Wang X 2022 Quantum state tomography inspired by language modeling (arXiv:2212.04940)
[266] Rocchetto A, Grant E, Strelchuk S, Carleo G and Severini S 2018 *npj Quantum Inf.* **4** 28
[267] Luchnikov I A, Ryzhov A, Stas P J, Filippov S N and Ouerdane H 2019 *Entropy* **21** 1091
[268] Walker N, Tam K M and Jarrell M 2020 *Sci. Rep.* **10** 13047
[269] Kingma D P and Welling M 2022 Auto-encoding variational bayes (arXiv:1312.6114)
[270] Schmitt M and Lenarčič Z 2022 *Phys. Rev.* B **106** L041110
[271] Czischek S, Moss M S, Radzihovsky M, Merali E and Melko R G 2022 *Phys. Rev.* B **105** 205108
[272] Ebadi S *et al* 2021 *Nature* **595** 227–32
[273] Bennewitz E R, Hopfmueller F, Kulchytskyy B, Carrasquilla J and Ronagh P 2022 *Nat. Mach. Intell.* **4** 618–24
[274] Montanaro A and Stanisic S 2023 Accelerating variational quantum monte carlo using the variational quantum eigensolver (arXiv:2307.07719)