14 Neural Quantum States for Strongly Correlated Systems

Annabelle Bohrdt
Ludwig-Maximilians-Universität München
Faculty of Physics and
Arnold Sommerfeld Center for Theoretical Physics
Theresienstr. 37, 80333 München

Contents

E. Pavarini and E. Koch (eds.)
Correlated Materials: Methods and Applications
Modeling and Simulation Vol. 16, Forschungszentrum Jülich, 2026
ISBN 978-3-95806-909-1 http://www.cond-mat.de/events/correl26

1 Introduction

Numerically simulating strongly correlated quantum systems is in general a challenging task: as the Hilbert space dimension grows exponentially, an exact representation is no longer feasible. Analytical approximations are often based on weak interactions or correlations and cannot be applied in the strongly correlated regime. In one dimension, matrix product states (MPS) constitute the state of the art to represent quantum many-body states efficiently, i.e., with a number of parameters that scales polynomially with the system size; see Ref. [1] for a review. Tensor network states in general make use of locality in order to efficiently represent states. For MPS and their one-dimensional structure, which renders the optimization through the density matrix renormalization group algorithm highly efficient, this however poses limitations on the class of states that can be represented with a number of parameters that scales polynomially and not exponentially with the system size. In particular the simulation of two-dimensional systems becomes highly challenging for MPS. A conceptually different approach has been taken in variational Monte Carlo: Before the advent of neural quantum states, variational Monte Carlo was commonly based on physically motivated trial wavefunctions. Typical examples include Slater–Jastrow states in continuum electronic-structure calculations and Gutzwiller-projected Fermi-sea or BCS states for lattice models. In these approaches, the qualitative structure of the wavefunction is chosen from physical intuition, while a comparatively small or structured set of variational parameters is optimized stochastically.

Ten years ago, a seminal paper proposed a new class of variational states: neural quantum states [2]. The basic idea is that a neural network is used to approximate the quantum many-body state

$$|\psi_\theta\rangle = \sum_{\mathbf{s}} \psi_\theta(\mathbf{s}) \, |\mathbf{s}\rangle \, , \tag{1}$$

where the network parameters[1] θ are the variational parameters of the ansatz. The sum runs over all possible configurations spanning a basis of the Hilbert space, e.g., for a spin-$1/2$ system all possible spin configurations in the z-basis, or for itinerant systems like a Fermi-Hubbard model all Fock space configurations. The input to the network is then a given basis configuration $\mathbf{s}$, and the output is the corresponding wavefunction coefficient $\psi_\theta(\mathbf{s})$. The same construction can be employed also in first quantization, where the input is the position of all N particles $\{\mathbf{r}_i\}_{i=1,\dots,N}$. We will in the following restrict ourselves to pure states, but note that density matrices can also be represented by neural quantum states with results for finite temperature states as well as open system dynamics. Neural quantum states generalize the philosophy of more conventional variational Monte Carlo approaches by replacing part or all of the hand-designed correlation structure by a highly flexible parametrization with many trainable parameters, thus to some degree trading interpretability against expressivity of the ansatz. We will discuss in more detail below in Sec. 2.1 how physically motivated design choices can be combined with neural networks.

[1] typically the weights and biases, see below for more details.

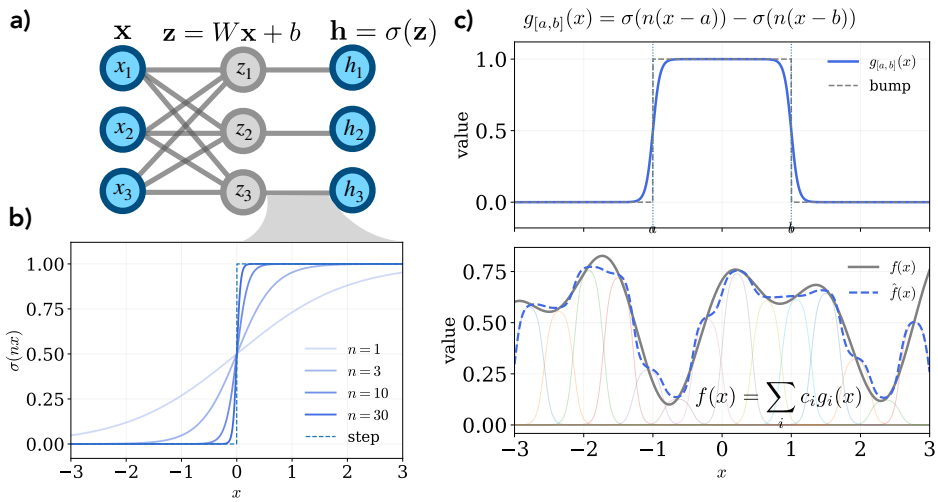

Fig. 1: *a) Basic building blocks of a feed-forward neural network. The input vector* $\mathbf{x}$ *is transformed by an affine map* $\mathbf{z} = W\mathbf{x} + \mathbf{b}$*, followed by a non-linear activation* $\mathbf{h} = \sigma(\mathbf{z})$*, depicted in b). c) a sigmoid activation can be combined to form approximate window functions, which in turn can be summed to approximate a target function.*

Contrary to typical machine learning problems, in general there exists no training data in this approach. The neural network represents the quantum state of interest, and therefore encodes the probability distribution from which one obtains samples according to the state following the Born rule. These samples are distributed as projective measurement outcomes of the current state in the chosen reference or computational basis. The data, i.e., the samples, is thus generated by the network itself, and each training step corresponds to a newly generated dataset. The updates to the network parameters are calculated based on these samples, and ideally, upon updating the network parameters many times, the quantum state represented by the neural network is very close to the desired state, e.g., the ground state of a given Hamiltonian. This optimization procedure, discussed in more detail in Sec. 3, follows the standard variational Monte Carlo approach: sample from the state, estimate expectation values from those samples, and update the variational parameters accordingly.

Compared to tensor networks, locality is not automatically built into a generic neural network architecture. It can, however, be incorporated through architectural choices such as local connectivity, convolutional layers, or equivariant constructions. Given the great expressivity of neural networks (to be discussed more below), the hope behind neural quantum states is to be able to represent a large class of interesting quantum many-body states efficiently.

Neural networks most generally consist of linear parts, i.e., matrix multiplications, and non-linear activation functions, see Fig. 1. These non-linear activation functions essentially give the neural network their power. Neural networks with already one hidden layer are universal function approximators: they can represent any continuous function on a bounded domain to arbitrary accuracy, given enough neurons [3, 4]. More formally, let $K \subset \mathbb{R}^n$ be compact and let $f \in C(K)$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a non-polynomial, continuous activation function. Then for every $\varepsilon > 0$, there exists an integer $N \in \mathbb{N}$ and parameters $a_i \in \mathbb{R}$, $w_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ such that

the function

$$\hat{f}(x) = \sum_{i=1}^{N} a_i \, \sigma(w_i \cdot x + b_i) \tag{2}$$

satisfies

$$\sup_{x \in K} \left| f(x) - \hat{f}(x) \right| < \varepsilon. \tag{3}$$

The basic intuition behind this theorem is as follows: a small network with a non-linear activation like a sigmoid, defined as

$$\sigma(x) = 1/(1 + e^{-x}), \tag{4}$$

can approximate a 'bump'. Consider an interval $[a, b]$, then the function

$$g(x) = \sigma(n(x-a)) - \sigma(n(x-b)) \tag{5}$$

approximates a function that is 1 on the interval $[a, b]$ and 0 else in the limit of large n, see Fig. 1c). Any function f that is piece-wise constant can now be approximated by sums of such small neural networks. A neural network essentially is a finite sum of functions of the form $c\sigma(ax+b)$, and the number of neurons determines how many such functions we sum over. In the case of neural quantum states, the neural network often maps from discrete numbers, like ± 1 on each site for a spin-1/2 system, to a wavefunction coefficient. Restricted Boltzmann machines have been shown to approximate probability distributions over discrete variables arbitrarily well [5]. Note that in practice, there are two potential problems with this theorem: (i) it does not guarantee an *efficient* representation, such that potentially prohibitively many parameters are needed; and (ii) it only makes a statement about expressivity, not about trainability: finding the set of parameters that corresponds for example to a desired many-body state is not guaranteed. In the remainder of these notes, we will first discuss the question of expressivity in Section 2 and then move to the optimization in Section 3. The focus of these notes is on variational pure-state NQS for strongly correlated lattice systems, with short comments on continuum fermions and time evolution. Other important directions, such as density-matrix parametrizations, open-system dynamics, and neural representations of operators, are mentioned only briefly. These lecture notes are intended as an introduction and rough overview, not as a comprehensive and exhaustive review. For the latter, see e.g. Refs. [6] and [7] for general reviews and [8] for a review on time evolution methods with neural quantum states.

2 Expressivity of neural network quantum states

The main promise, and at the same time the main mystery, of neural quantum states is their expressivity: per the universal function approximator theorem, neural networks can in principle represent any smooth function with arbitrary accuracy. In practice, we are however limited to a finite number of variational (read: network) parameters. This poses the question: how expressive is a neural network with a fixed number of parameters? A closely related question pertains to how the network makes use of these parameters to represent a desired quantum state, and

how we should choose the network architecture for a given problem. While the original work in Ref. [2] focused on restricted Boltzmann machines, many different network architectures are now being used as neural quantum states, ranging from convolutional neural networks to Transformers. Additionally, standard neural network architectures can be combined with physically motivated constructions, such as determinants. Depending on the problem under consideration, different representations can be chosen, e.g., regarding the reference basis, reference sign structure, complex versus real output, or representation of a density matrix for mixed states.
Early work has shown that some interesting quantum many-body states can be represented exactly by a neural network, sometimes by default, sometimes per clever choice of the network architecture. We will discuss some of these instances in section 2.1 to gain a feeling for how the specific architecture and/or non-linearities can be used to enforce physical properties of the state under consideration. Through physically inspired architectures, we can gain not only an advantage in expressivity for the problem under consideration, but also some sense of interpretability. However, the guiding principle for what neural quantum states can represent efficiently remains a subject of ongoing research. This is to be contrasted with one of the state-of-the-art methods to numerically capture quantum many-body systems, namely matrix product states (MPS). In the case of MPS, the entanglement constitutes a guiding principle, which limits which states can be efficiently represented. Entanglement properties of neural quantum states depend, once again, on architectural choices, such as number and choice of non-linearities, but at least some volume law entangled states can be represented efficiently using neural quantum states [9, 10]. We will discuss the relation between matrix product states and neural quantum states as well as the latter's entanglement properties in more detail in section 2.2

2.1 Exactly representable states

Historically, restricted Boltzmann machines (RBMs), see Fig. 2a) have been used as the first neural quantum states [2]. While other architectures may exhibit more expressivity, RBMs often enable a better understanding. We can write an RBM wavefunction for spins $s_i = \pm 1$ with M hidden units as

$$\psi_{\mathrm{RBM}}(\mathbf{s}) = \exp\left(\sum_i a_i s_i\right) \prod_{\mu=1}^{M} 2\cosh\left(b_\mu + \sum_i W_{i\mu} s_i\right). \tag{6}$$

In the following, we want to use this RBM representation to demonstrate how different types of quantum states can be represented. We start by the N-spin GHZ state,

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{2}}\big(|\uparrow\uparrow\cdots\uparrow\rangle + |\downarrow\downarrow\cdots\downarrow\rangle\big), \tag{7}$$

whose computational-basis amplitudes are nonzero only if $s_1 = s_2 = \cdots = s_N$. This global condition can be enforced by nearest-neighbor equality constraints,

$$\psi_{\mathrm{GHZ}}(\mathbf{s}) \propto \prod_{i=1}^{N-1} \delta_{s_i, s_{i+1}}. \tag{8}$$

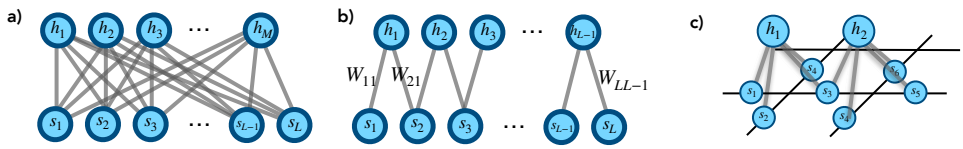

Fig. 2: *a) Restricted Boltzmann machine architecture; b) Connectivity for the complex restricted Boltzmann machine to represent the GHZ state, where alignment of each pair of nearest neighboring spins is mediated through a hidden unit; c) Connectivity for a complex restricted Boltzmann machine to represent the Toric code ground state with one hidden unit per vertex.*

Each equality constraint can be implemented by one complex hidden unit[2], see Fig. 2b)

$$2\cosh\left[\frac{i\pi}{4}(s_i - s_{i+1})\right] = \begin{cases} 2, & s_i = s_{i+1}, \\ 0, & s_i \neq s_{i+1}. \end{cases} \tag{9}$$

Thus, choose $M = N-1$ hidden units, no visible or hidden biases, $a_i = 0$, $b_\mu = 0$, and weights

$$W_{i\mu} = \frac{i\pi}{4}\delta_{i,\mu} - \frac{i\pi}{4}\delta_{i,\mu+1}, \qquad \mu = 1, \ldots, N-1. \tag{10}$$

The resulting RBM wavefunction is

$$\psi_{\mathrm{RBM}}(\mathbf{s}) = \prod_{\mu=1}^{N-1} 2\cosh\left[\frac{i\pi}{4}(s_\mu - s_{\mu+1})\right]. \tag{11}$$

Therefore,

$$\psi_{\mathrm{RBM}}(\mathbf{s}) = \begin{cases} 2^{N-1}, & s_1 = s_2 = \cdots = s_N, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

After normalization, this gives exactly

$$|\psi_{\mathrm{RBM}}\rangle = \frac{1}{\sqrt{2}}\big(|\uparrow\uparrow\cdots\uparrow\rangle + |\downarrow\downarrow\cdots\downarrow\rangle\big). \tag{13}$$

This example demonstrates how hidden units can be used to enforce local constraints, in this case $s_i = s_{i+1}$.
Similarly, hidden units can be used to enforce local parity constraints, see also [9]. Our next example is the Toric-code Hamiltonian [11] with spins on the links (edges) of a square lattice, indicated by the index e,

$$H_{\mathrm{TC}} = -\sum_v A_v - \sum_p B_p, \qquad A_v = \prod_{e\in+_v} \sigma_e^x, \qquad B_p = \prod_{e\in\partial p} \sigma_e^z. \tag{14}$$

We work in the σ^x-basis,

$$\sigma_e^x|s_e\rangle_x = s_e|s_e\rangle_x, \qquad s_e = \pm 1. \tag{15}$$

[2]For real finite RBM parameters, these hard zeros cannot be enforced exactly; see the discussion below.

In this basis, the vertex constraint $A_v = 1$ becomes

$$\prod_{e \in +_v} s_e = 1. \tag{16}$$

Thus a toric-code ground state may be written as an equal-weight superposition over closed-loop configurations,

$$|\psi_{\mathrm{TC}}\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{\mathbf{s}:\, \prod_{e \in +_v} s_e = 1 \ \forall v} |\mathbf{s}\rangle_x. \tag{17}$$

Using an RBM with binary hidden variables $h_v \in \{0, 1\}$,

$$\psi_{\mathrm{RBM}}(\mathbf{s}) = \prod_v \left[1 + \exp\left(b_v + \sum_e W_{ev} s_e \right) \right], \tag{18}$$

one hidden unit per vertex, i.e., connecting to four visible units, enforces the local parity constraint. Choose

$$b_v = \frac{i\pi d_v}{2}, \qquad W_{ev} = \begin{cases} -\frac{i\pi}{2}, & e \in +_v, \\ 0, & \text{otherwise}, \end{cases} \tag{19}$$

where d_v is the degree of vertex v. Then

$$1 + \exp\left(b_v + \sum_e W_{ev} s_e \right) = 1 + \prod_{e \in +_v} s_e. \tag{20}$$

Therefore

$$\psi_{\mathrm{RBM}}(\mathbf{s}) = \prod_v \left(1 + \prod_{e \in +_v} s_e \right) = \begin{cases} 2^{N_v}, & \prod_{e \in +_v} s_e = 1 \quad \forall v, \\ 0, & \text{otherwise}. \end{cases} \tag{21}$$

After normalization, this is exactly the toric-code ground state in the σ^x-basis. The plaquette terms B_p flip closed loops, and the equal-weight superposition is invariant under these flips.
Apart from constraints, hidden units can also be viewed as effectively mediating correlations between visible units. Consider a spin Jastrow wavefunction

$$\psi_{\mathrm{J}}(\mathbf{s}) = \exp\left(\sum_i a_i s_i + \sum_{i<j} J_{ij} s_i s_j \right), \qquad s_i = \pm 1. \tag{22}$$

This state can be represented exactly by an RBM with one hidden unit for each nonzero pair coupling J_{ij} [12].
For a single pair (i, j), introduce a hidden unit coupled only to s_i and s_j,

$$\psi_{ij}(s_i, s_j) = 2\cosh\left[w_{ij} \left(s_i + \eta_{ij} s_j \right) \right], \tag{23}$$

where

$$\eta_{ij} = \mathrm{sgn}(J_{ij}), \qquad w_{ij} = \frac{1}{2} \operatorname{arcosh}\left(e^{2|J_{ij}|} \right). \tag{24}$$

If $s_i s_j = \eta_{ij}$, then

$$\psi_{ij}(s_i, s_j) = 2\cosh(2w_{ij}) = 2e^{2|J_{ij}|}, \tag{25}$$

whereas if $s_i s_j = -\eta_{ij}$, then

$$\psi_{ij}(s_i, s_j) = 2. \tag{26}$$

Thus,

$$\psi_{ij}(s_i, s_j) = C_{ij} \exp\left(J_{ij} s_i s_j\right), \tag{27}$$

with the spin-independent constant $C_{ij} = 2e^{|J_{ij}|}$. Constants of this form only affect the overall normalization. Therefore, choosing one hidden unit for every pair (i, j) gives

$$\psi_{\mathrm{RBM}}(\mathbf{s}) = \exp\left(\sum_i a_i s_i\right) \prod_{i<j} 2\cosh\left[w_{ij}\left(s_i + \mathrm{sgn}(J_{ij}) s_j\right)\right]. \tag{28}$$

Using the identity above,

$$\psi_{\mathrm{RBM}}(\mathbf{s}) \propto \exp\left(\sum_i a_i s_i + \sum_{i<j} J_{ij} s_i s_j\right) = \psi_{\mathrm{J}}(\mathbf{s}). \tag{29}$$

Hence a spin Jastrow state is exactly representable by an RBM with one hidden unit per nonzero pair interaction.

These examples demonstrate how hidden units can encode correlations as well as enforce constraints between the visible units. Along the same lines of reasoning, it has been shown that an exact representation of a variety of states is efficiently possible using neural quantum states, e.g., the Laughlin state [13–15] or Toric code ground states with independently tunable topological-sector amplitudes [16]. Using deeper networks and more advanced architectures, the same intuition applies in principle.

The exact RBM constructions above rely on complex-valued parameters. This is important: with real finite parameters in the standard RBM form, the factors $2\cosh(b_\mu + \sum_i W_{i\mu} s_i)$ are nonzero for generic configurations, such that hard zeros in the wavefunction are not obtained exactly. Complex parameters, on the other hand, allow one to use zeros of the hyperbolic cosine to enforce constraints exactly. Thus, hidden units can be viewed not only as mediating correlations, but also as enforcing hard constraints through destructive interference.

Matrix Product States.– A different angle is provided by relating matrix product states to neural quantum states: here, we do not explicitly construct the weights and biases of the neural network to achieve certain conditions on $\psi(\mathbf{s})$ as above, but instead use the non-linearities of the neural network to mimic the tensor contractions within the matrix product state. Different approaches can be used to represent a given MPS as an NQS: in Ref. [17], a tensor network contraction is expressed as a sum over discrete auxiliary variables, i.e., the hidden variable marginalization of a Boltzmann machine. This can be viewed as the virtual indices of the matrices/tensors in the MPS become the hidden spins in the network. This Boltzmann-machine representation[3] of the state is efficient in the sense that it has an almost-linear overhead in the number of nonzero MPS parameters. In a related work [18], explicit mappings between tensor

[3] Note that this is not a restricted Boltzmann machine.

networks and restricted Boltzmann machines are constructed, with the important consequence that a non-local restricted Boltzmann machine can correspond to a exponentially growing bond dimension (i.e. not an efficient representation) for the local tensor network. Beyond marginalizing over hidden units, one can also view the network itself, i.e., the computation of hidden variables throughout the layers, as a form of contraction. In [19], a neural network is constructed such that the layers emulate the tensor contraction. The network here is viewed as a computational circuit that evaluates the contraction. Using this algorithm, efficiently contractible tensor networks can be represented by polynomially sized neural networks, with network size controlled by the contraction complexity. A conceptually slightly different approach is taken in Ref. [20], where the mapping from an MPS to a recurrent neural network (RNN) is shown. In this case, one can make use of the fact that the left (or right) environment of an MPS can be defined recursively, such that the hidden state of the RNN can be identified with the current MPS boundary matrix. The RNN then updates its memory exactly as the MPS contraction would update the boundary as one moves through the system. The squared MPS bond dimension χ^2 in this case corresponds to the RNN memory size.

In general, a tensor network represents the state $\psi(\mathbf{s})$ by contracting virtual indices (i.e. performing tensor products), while neural quantum states represent $\psi(\mathbf{s})$ by computing or marginalizing over hidden variables. This enables mappings between the two different representations since contracting over virtual indices can be viewed as a hidden variable computation. These results show that, if a quantum many-body state can be expressed as a matrix product state, then there exists an efficient neural quantum state representation of this state.

Fermions.– Among the most challenging systems for numerical simulations are interacting fermions in two or three spatial dimensions. In order to gain intuition how to construct neural networks that can represent quantum states of interacting fermions, we will start from non-interacting fermions on a lattice, following Ref. [21], described by the Hamiltonian

$$\hat{H}_0 = \sum_{i,j} \sum_{\alpha,\beta} h_{i\alpha,j\beta}\, \hat{c}^\dagger_{i\alpha} \hat{c}_{j\beta}, \tag{30}$$

where i, j label lattice sites and α, β label spin. It is useful to combine site and spin into a single index $I = (i, \alpha)$.

Diagonalizing the single-particle Hamiltonian gives orbitals $\Phi_{I\epsilon}$ and operators

$$\hat{\gamma}^\dagger_\epsilon = \sum_I \Phi_{I\epsilon} \hat{c}^\dagger_I, \qquad \hat{H}_0 = \sum_\epsilon \varepsilon_\epsilon \hat{\gamma}^\dagger_\epsilon \hat{\gamma}_\epsilon. \tag{31}$$

For N fermions, the many-body ground state is obtained by occupying the N lowest single-particle orbitals,

$$|\psi_0\rangle = \prod_{\epsilon=1}^{N} \hat{\gamma}^\dagger_\epsilon |0\rangle = \prod_{\epsilon=1}^{N} \Big(\sum_I \Phi_{I\epsilon} \hat{c}^\dagger_I \Big) |0\rangle. \tag{32}$$

Now consider a Fock basis state $|\mathbf{s}\rangle$ with occupied spin-orbitals $I_1, \ldots, I_N$, written in a fixed canonical order,

$$|\mathbf{s}\rangle = \hat{c}^\dagger_{I_1} \hat{c}^\dagger_{I_2} \cdots \hat{c}^\dagger_{I_N} |0\rangle. \tag{33}$$

Its overlap with the free-fermion ground state is

$$\langle \mathbf{s}|\psi_0\rangle = \langle 0|\hat{c}_{I_N}\cdots\hat{c}_{I_1}\prod_{\epsilon=1}^{N}\left(\sum_I \Phi_{I\epsilon}\hat{c}_I^\dagger\right)|0\rangle. \tag{34}$$

Only those terms survive in which the N occupied orbitals $\epsilon = 1,\ldots,N$ are assigned to the occupied spin-orbitals $I_1,\ldots,I_N$. Thus,

$$\langle \mathbf{s}|\psi_0\rangle = \sum_{p\in S_N}\left[\prod_{\epsilon=1}^{N}\Phi_{I_{p(\epsilon)},\epsilon}\right]\langle 0|\hat{c}_{I_N}\cdots\hat{c}_{I_1}\hat{c}_{I_{p(1)}}^\dagger\cdots\hat{c}_{I_{p(N)}}^\dagger|0\rangle. \tag{35}$$

Using the fermionic anticommutation relations, the creation operators are reordered into the canonical order,

$$\hat{c}_{I_{p(1)}}^\dagger\cdots\hat{c}_{I_{p(N)}}^\dagger = \mathrm{sgn}(p)\,\hat{c}_{I_1}^\dagger\cdots\hat{c}_{I_N}^\dagger. \tag{36}$$

Therefore,

$$\langle 0|\hat{c}_{I_N}\cdots\hat{c}_{I_1}\hat{c}_{I_{p(1)}}^\dagger\cdots\hat{c}_{I_{p(N)}}^\dagger|0\rangle = \mathrm{sgn}(p), \tag{37}$$

and hence

$$\langle \mathbf{s}|\psi_0\rangle = \sum_{p\in S_N}\mathrm{sgn}(p)\prod_{\epsilon=1}^{N}\Phi_{I_{p(\epsilon)},\epsilon}. \tag{38}$$

This is precisely the Leibniz formula for the determinant of the matrix obtained by selecting from Φ the rows corresponding to the occupied spin-orbitals in $\mathbf{s}$

$$\langle \mathbf{s}|\psi_0\rangle = \det\begin{pmatrix} \Phi_{I_1,1} & \Phi_{I_1,2} & \cdots & \Phi_{I_1,N} \\ \Phi_{I_2,1} & \Phi_{I_2,2} & \cdots & \Phi_{I_2,N} \\ \vdots & \vdots & & \vdots \\ \Phi_{I_N,1} & \Phi_{I_N,2} & \cdots & \Phi_{I_N,N} \end{pmatrix}. \tag{39}$$

Equivalently, if $\mathbf{s}\circ\Phi$ denotes the submatrix of Φ obtained by selecting the occupied rows specified by $\mathbf{s}$, then $\langle \mathbf{s}|\psi_0\rangle = \det(\mathbf{s}\circ\Phi)$. At this point, we have an exact expression for the wavefunction coefficients $\psi_0(\mathbf{s})$ of the ground state of a fermionic system without interactions and without neural network.

In order to include interactions, we have to go beyond the single-particle orbitals used so far. As a concrete example, we consider the prototypical model for interacting spin-$1/2$ fermions on a lattice, see also [22]: the Fermi-Hubbard model, given by

$$\hat{H} = -t\sum_{\langle i,j\rangle,\sigma}\left(\hat{c}_{i\sigma}^\dagger\hat{c}_{j\sigma} + \mathrm{h.c.}\right) + U\sum_i \hat{n}_{i\uparrow}\hat{n}_{i\downarrow}. \tag{40}$$

Here $\hat{c}_{i\sigma}^\dagger$ and $\hat{c}_{i\sigma}$ create and annihilate a fermion with spin $\sigma\in\{\uparrow,\downarrow\}$ on lattice site i. The first term describes hopping between nearest-neighbor sites $\langle i,j\rangle$ with amplitude t, while the second term describes an on-site interaction of strength U between opposite spins.

As a minimal example, we consider the two-site Fermi–Hubbard model in the $N_v = 2$ particle sector with $S^z_v = 0$. The exact ground state can be written as

$$|\psi_0\rangle = \left[\frac{\cos\theta}{\sqrt{2}}\left(\hat{c}^\dagger_{1\uparrow}\hat{c}^\dagger_{1\downarrow} + \hat{c}^\dagger_{2\uparrow}\hat{c}^\dagger_{2\downarrow}\right) + \frac{\sin\theta}{\sqrt{2}}\left(\hat{c}^\dagger_{1\uparrow}\hat{c}^\dagger_{2\downarrow} + \hat{c}^\dagger_{2\uparrow}\hat{c}^\dagger_{1\downarrow}\right)\right]|0\rangle. \tag{41}$$

The angle θ is determined by

$$\frac{1}{\tan\theta} = u\left(\sqrt{1+\frac{1}{u^2}} - 1\right), \qquad u = \frac{U}{4t}. \tag{42}$$

For $U = 0$, one has $\theta = \pi/4$, such that all four configurations have equal weight. In this case the state is a single Slater determinant, $\langle \mathbf{s}|\psi_{\mathrm{SD}}\rangle = \det(\mathbf{s} \circ \Phi)$, with

$$\Phi = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix}. \tag{43}$$

Here the rows correspond to the physical spin-orbitals $(1\uparrow, 2\uparrow, 1\downarrow, 2\downarrow)$, and the two columns correspond to the occupied spin-up and spin-down orbitals.

For $U > 0$, the relative weight of doubly occupied configurations is reduced, while inter-site singlet configurations become enhanced. This can no longer be represented by a single Slater determinant of the visible fermions. In order to express this state, we introduce a single (spinless) hidden fermion, that can occupy two possible hidden states, denoted by S and D. The hidden orbital is chosen as $\widetilde{\Phi} = (c\tan\theta, c)^T$, where c is a normalization constant. The hidden occupation $\widetilde{\mathbf{s}}$ is made explicitly dependent on the configuration of the visible fermions,

$$\widetilde{\mathbf{s}}(\mathbf{s}) = \begin{cases} S, & \text{if there is no double occupation in } \mathbf{s}, \\ D, & \text{otherwise.} \end{cases} \tag{44}$$

Thus the hidden fermion selects different hidden amplitudes depending on whether the visible configuration is singly occupied on both sites or contains a doublon. In order to obtain the wave function of the physical fermions, we project back onto the physical Hilbert space by fixing the hidden occupation according to $\widetilde{\mathbf{s}}(\mathbf{s})$, expressed as $\hat{P}$ below

$$|\psi_{\mathrm{HFDS}}\rangle = \hat{P}\left[\frac{c}{2}\left(\hat{c}^\dagger_{1\uparrow}\hat{c}^\dagger_{1\downarrow} + \hat{c}^\dagger_{2\uparrow}\hat{c}^\dagger_{2\downarrow}\right)|0\rangle_v \otimes |D\rangle_h + \frac{c\tan\theta}{2}\left(\hat{c}^\dagger_{1\uparrow}\hat{c}^\dagger_{2\downarrow} + \hat{c}^\dagger_{2\uparrow}\hat{c}^\dagger_{1\downarrow}\right)|0\rangle_v \otimes |S\rangle_h\right]. \tag{45}$$

The hidden fermion thus, similar to the hidden nodes in the RBM examples above, mediates correlations between the physical or visible fermions and allow us to express correlated fermionic states as the determinant of an enlarged matrix.

The two-site example illustrates the basic idea behind hidden fermion determinant states, originally introduced in Ref. [23]: one augments the physical Hilbert space by auxiliary fermionic

degrees of freedom and then projects back to the physical Hilbert space. The hidden degrees of freedom allow the effective determinant to depend on the visible configuration.
We will now introduce this approach a bit more generally, i.e., going beyond the toy example of two visible and a single hidden fermion. We denote a Fock-space configuration in the enlarged Hilbert space by $|\mathbf{s}, \widetilde{\mathbf{s}}\rangle$, where $\mathbf{s}$ labels the physical fermion occupations and $\widetilde{\mathbf{s}}$ labels the hidden fermion occupations. Let Φ_v and Φ_h denote the orbital matrices associated with the visible and hidden fermions, respectively. Before projection, the wavefunction amplitude in the enlarged Hilbert space is given by a determinant,

$$\langle \mathbf{s}, \widetilde{\mathbf{s}} | \psi_{\mathrm{HFDS}} \rangle = \det \begin{pmatrix} \mathbf{s} \circ \Phi_v \\ \widetilde{\mathbf{s}} \circ \Phi_h \end{pmatrix}. \tag{46}$$

Here $\mathbf{s} \circ \Phi_v$ denotes the rows of Φ_v selected by the occupied visible spin-orbitals in $\mathbf{s}$, and $\widetilde{\mathbf{s}} \circ \Phi_h$ denotes the rows of Φ_h selected by the occupied hidden orbitals. To obtain a wavefunction in the physical Hilbert space, the hidden configuration is made dependent on the visible one, $\widetilde{\mathbf{s}} = \widetilde{\mathbf{s}}(\mathbf{s})$. This gives the projected amplitude

$$\langle \mathbf{s} | \psi_{\mathrm{HFDS}} \rangle = \det \begin{pmatrix} \mathbf{s} \circ \Phi_v \\ \widetilde{\mathbf{s}}(\mathbf{s}) \circ \Phi_h \end{pmatrix}. \tag{47}$$

The lower blocks are configuration-dependent because the hidden occupation $\widetilde{\mathbf{s}}(\mathbf{s})$ is chosen as a function of the visible configuration. Thus, hidden fermions turn a fixed Slater determinant in an enlarged Hilbert space into a configuration-dependent determinant in the physical Hilbert space,

$$\psi_{\mathrm{HFDS}}(\mathbf{s}) = \det M(\mathbf{s}). \tag{48}$$

In practical hidden-fermion determinant states, the lower blocks of the matrix are parametrized by a neural network. A different view on the entire hidden fermion construction is thus that the final non-linear activation function is a determinant [24]. From a standard machine-learning perspective, a determinant is an unusual choice of non-linear activation. However, as we have seen above, from a physical perspective, this choice comes very naturally, as free fermions are captured exactly, and correlations can be encoded efficiently. An additional angle on why the determinant can be a highly efficient non-linear activation function is that it naturally leads to high-order correlations between the different inputs and thus visible nodes, which can, e.g., easily lead to a favorable entanglement scaling as discussed in more detail in Sec. 2.2.
This expressive determinant activation comes with an additional numerical cost: each wavefunction evaluation requires a determinant of the enlarged visible-plus-hidden matrix, scaling naively as $O(N_{\mathrm{tot}}^3)$. During Monte Carlo sampling, however, successive configurations often differ only locally, so determinant ratios and inverse matrices can be updated using low-rank formulas rather than recomputed from scratch. Recent fermionic NQS architectures exploit this idea explicitly, trading off expressivity and efficient local updates to reach a favorable accuracy-cost Pareto frontier [23, 25, 26].
Apart from the hidden fermion determinant states, there are other neural quantum state constructions in similar spirit to describe interacting fermions:

- multiplying a fermionic reference state, such as a Slater determinant or pair-product/geminal state, by a correlation factor. This factor can be a conventional Jastrow factor or a neural-network correlator, for example an RBM or autoregressive network [27, 28].

- neural backflow, where a neural network is used to make the single particle orbitals configuration dependent [29]. It has been shown that the hidden fermion determinant can be expressed as a neural backflow state with a Jastrow factor [25].

- replacing the determinant in the above constructions by a Pfaffian, which is naturally equipped to capture pairing [30].

The Fermi–Hubbard model discussed above is a minimal setting in which the competition between kinetic energy and local interactions is present. The fermionic constructions discussed above, however, are not restricted to this model. They apply more generally to lattice fermion Hamiltonians, including longer-range hopping [31] and interactions, multi-orbital models [32, 33], disordered systems, and all-to-all interacting fermionic models [34, 10].
The determinant- and Pfaffian-based ideas discussed above for lattice fermions also have natural analogues in continuum systems. In this setting, one often works in a first-quantized representation, where a configuration is given by the particle coordinates $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$, possibly together with spin labels, and the wavefunction is represented directly as $\psi_\theta(\mathbf{R})$. Fermionic antisymmetry is then imposed at the level of the coordinate-space wavefunction, rather than through anticommuting creation and annihilation operators. A common strategy is to combine neural-network correlation factors or backflow transformations with antisymmetric structures such as Slater determinants or Pfaffians. This is the basic idea behind continuum fermionic neural-network wavefunctions such as FermiNet and PauliNet, as well as more recent Pfaffian–Jastrow–backflow ansätze for continuum Fermi gases [35–37]. These continuum constructions are conceptually close to the lattice determinant-based NQS discussed above: neural networks make the orbitals, pair functions, or correlation factors configuration-dependent, thereby introducing correlations beyond a fixed mean-field determinant or Pfaffian. Recent work also investigates the expressivity of fermionic neural-network architectures and asks under which conditions antisymmetric wavefunctions can be approximated systematically by neural ansätze [38, 39].

2.2 Entanglement properties

In order to quantify the expressivity of a variational state, some measure has to be chosen. A popular choice is the entanglement, as it provides a resource in quantum computing, plays a crucial role in many quantum systems, and is known to be the guiding principle for matrix product states. Typically, we consider the bipartite entanglement, i.e., we split our full system into a subsystem A and B, trace out part B, and consider properties of the resulting reduced density matrix ρ_A of part A in order to quantify how entangled A and B are. Upon writing the

full state as a Schmidt decomposition

$$|\psi\rangle = \sum_j \sqrt{\lambda_j}\, |\psi_j^A\rangle \otimes |\psi_j^B\rangle \tag{49}$$

with Schmidt coefficients λ_j, we can write the reduced density matrix as

$$\rho_A = \sum_j e^{-\xi_j}\, |\psi_j^A\rangle \langle\psi_j^A| \tag{50}$$

with entanglement energies $\xi_j = -\log \lambda_j$. The von Neumann entanglement entropy is then

$$S_A = -\operatorname{Tr} \rho_A \log \rho_A = -\sum_i \lambda_i \log \lambda_i = \sum_i \lambda_i \xi_i. \tag{51}$$

Note that the von Neumann entanglement entropy is just one scalar summary of the entire spectrum, and therefore naturally more information is contained in the full entanglement spectrum. Often the 'amount' of entanglement, and in particular the scaling of S_A with the area (boundary) and volume of the region A is considered. Interesting entanglement properties can however go beyond this scaling, e.g., the topological entanglement entropy. For neural quantum states on system sizes beyond exact diagonalization, the von Neumann entanglement entropy cannot be evaluated efficiently and one resorts typically to the second Rényi entropy, which can be evaluated based on Monte Carlo samples through the SWAP trick [40] as

$$S_{2,A} = -\log \operatorname{Tr} \rho_A^2 = -\log \sum_i \lambda_i^2 = -\log \langle \mathrm{SWAP}_A \rangle\,. \tag{52}$$

Similar to the von Neumann entanglement entropy, the second Rényi entropy is also a scalar summary of the full entanglement spectrum. The latter is however more sensitive to the largest eigenvalues, while the von Neumann entropy weights the tail of the distribution in comparison more.

Matrix product states are constructed such that the bond dimension χ directly corresponds to the number of Schmidt coefficients λ_j that is taken into account at each bond along the one-dimensional path through the system. Schmidt values λ_j with $j > \chi$ are set to zero, effectively truncating the entanglement spectrum and thereby limiting how much entanglement S_A can be represented. For neural quantum states, the natural question arises: what determines how much entanglement can be represented by a given architecture? In order to tackle this question, we have to be even more precise and distinguish: (i) capability – what can the architecture represent if the best choice of parameters is found, which may provide a highly non-trivial task; (ii) trainability – (how) can we determine these optimal parameters; and (iii) typicality – how much entanglement[4] does this architecture feature upon random initialization, i.e., without any training. We discuss some results on the capability and typicality of neural quantum states with regard to entanglement below, and refer to section 3 for more details on trainability in general.

[4] Or whatever other measure we want to apply to quantify expressivity.

2.2.1 Capability

A number of research papers following different strategies has by now found that there are (at least) two ingredients with a strong influence on the capabilities of a given architecture:

Locality.– The connectivity within the network has – at least in the context of RBMs – been shown to determine the network's capability to represent area versus volume law entanglement: locality, i.e., restricting connections to a few nodes, leads to the capability to represent area but not volume law entanglement [9]. This is conceptually similar to the case of MPS, where locality is exploited to efficiently optimize the variational state, but also leads to the discussed limitations regarding entanglement behavior. An RBM with long-range connectivity, which is the standard case of each hidden node being connected to each visible node, can represent volume law entanglement.

Non-linearity.– We have seen in Sec. 2.1 at the example of the RBM how the hidden units effectively can mediate higher-order correlation function. Beyond the concrete examples shown above, one can also see how (higher-order) correlations between the visible units mediated by the hidden units emerge through a Taylor expansion, see [41] for more details. Similarly, Ref. [42] employs a Taylor expansion of the non-linear activation function to prove that an extensive number of non-linearities is necessary in a feed-forward neural network to represent a volume law entangled state. This proof relies on the fact that only a small number of sites is coupled in each non-linearity, such that only a finite amount of entanglement can be generated. The results of [42] are in line with earlier work emphasizing the role of the depth [43]. In Ref. [24], the standard non-linear activation function is fully replaced by an expansion in orders of products of the visible units or correlations. This correlator transformer approach provides a particularly clear picture on the role of the non-linearity: it is shown in [24] that in order to represent volume law entanglement, correlations of the order of the system size are required. Typical non-linear activation functions however have very quickly decaying pre-factors of such high-order correlations of the visible units. The required large pre-factors can be restored through deeper networks, where the repeated application of the corresponding non-linearity enables higher-order correlation between the visible units.
As pointed out in [24] and [42], the determinant (and similarly the Pfaffian) in the fermionic ansätze described above can effectively be viewed as a special non-linear activation function, which immediately enables access to high order polynomials of the inputs. These determinant and Pfaffian based constructions therefore more naturally represent states featuring high amounts of entanglement. While this can be naturally seen through the arguments given in [24] in terms of high order polynomials of the inputs which encode the entanglement, one can also see this from a more physical point of view, as it has been shown that states with a Fermi surface naturally exhibit an entanglement scaling of $L^{d-1} \log L$, so a multiplicative logarithmic correction to an area law scaling [44]. The determinant construction naturally encompasses states featuring a Fermi surface and thus the corresponding entanglement scaling.

This favorable behavior of the determinant has also been observed in practice: Ref. [45] found that representing the ground state of the SYK model, which features volume law entanglement, requires an exponential in system size number of parameters if the state is represented by a feedforward neural network. At the time, the authors concluded that neural quantum states cannot represent volume law entangled states. A comment on this paper, Ref. [10], argued that the failure to efficiently represent the state is due to the attempt to represent a fermionic ground state with an architecture that is not tailored towards fermions. A complementary perspective is provided by the role of the non-linearity: determinant-based ansätze provide high-order polynomial structure from the start, and therefore offer a much more suitable inductive bias for fermionic states with large entanglement than generic feed-forward architectures.

2.2.2 Typicality

In order to determine how much entanglement a given architecture typically features, the network parameters are randomly initialized, and the entanglement entropy is evaluated without any training. Ref. [9] demonstrates that the von Neumann entanglement entropy for randomly initialized RBMs exhibits volume law behavior for the small system sizes considered, as long as the ratio of hidden to visible units does not become too large. Many hidden units, and therefore many variational parameters, may seem favorable for the expressivity of the ansatz. However, at least in the case of random initialization considered here, more hidden units lead effectively to a state closer to a product state because there is less dependence on the visible units [9].
In [46], different autoregressive architectures, which commonly feature a softmax as their output activation function, are considered. Autoregressive architectures are designed using a chain rule of probabilities, such that each local probability can be normalized, yielding a fully normalized probability. The output activation function is therefore commonly chosen to be a softmax. The normalized probabilities enable perfect sampling and avoid Markov-chain Monte Carlo sampling, i.e., there are no autocorrelations. The authors of Ref. [46] evaluate the second Rényi entropy $S_{2,A}$ for different widths of the distribution from which the random network parameters are sampled and examine the scaling of $S_{2,A}$ with the size of the region A. The two main findings of this work are: (i) the non-linear (output) activation function matters: the standard softmax normalization can suppress fluctuations and entanglement in randomly initialized autoregressive wavefunctions, especially when the logits become large. They proposed a square-modulus normalization as an alternative output map, which restores stronger fluctuations and larger entanglement in the random-state ensemble; and (ii) the width of the random initialization distribution can strongly affect both the entanglement of autoregressive NQS and their subsequent VMC performance. For Gaussian-initialized RNN and autoregressive transformer wavefunctions, the average entanglement is typically maximized at an intermediate width: too small widths give nearly product states, while too large widths can lead, in softmax-based architectures, to localized output distributions and suppressed entanglement. The Gaussian width that minimizes VMC convergence time is also found at intermediate values and shifts to smaller values as the network size increases. This suggests that initialization should be chosen with the entanglement structure of the ansatz in mind.

In Ref. [47], the second Rényi entropy is evaluated for randomly initialized non-autoregressive vision transformers as well as Gutzwiller projected hidden fermion determinant states. For various widths of the distribution from which the network weights are sampled, the hidden fermion determinant state exhibits a higher value of the second Rényi entropy than the vision transformer. Simultaneously, a larger width of the distribution of the network weights consistently leads to a higher entanglement entropy for the vision transformer, as predicted by the arguments in [24]. From the arguments discussed in Sec. 2.2.1, a high entanglement content is expected for the determinant based construction.

3 Training neural quantum states

In order to represent a desired quantum state, we need to find the best variational parameters θ for the problem at hand. The procedure of optimizing these parameters is often referred to as training, and typically consists of an iterative procedure, where the parameters are changed by some small amount $\delta\theta$ in each training step. Given the representation we have (configuration in, wavefunction coefficient out), everything is based on samples from the quantum many-body state.

Below, we first outline how to evaluate observables using samples from the probability distribution defined by our neural quantum state. This constitutes the main tool required to implement the variational Monte Carlo (VMC) method. The core objective of VMC is to optimize the parameters θ of a trial wavefunction $|\psi_\theta\rangle$ in order to find the best possible approximation to our target state, e.g., the ground state of a given Hamiltonian. The same variational framework can be used for several tasks. In ground-state searches, one minimizes the variational energy. In real- or imaginary-time evolution, one updates the parameters such that the variational state follows the projected time-dependent Schrödinger equation. If a target state is known, one may instead minimize the infidelity, and if measurement data are available, one can train the NQS by state reconstruction. In all cases, the central computational ingredients are wavefunction ratios, Monte Carlo samples, and parameter derivatives of the log-amplitude.

3.1 Variational Monte Carlo

The expectation value of a general quantum-mechanical observable $\hat{O}$ with respect to the trial wavefunction $|\psi_\theta\rangle$ is

$$\langle \hat{O} \rangle_\theta = \frac{\langle \psi_\theta | \hat{O} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}. \tag{53}$$

By inserting a resolution of the identity,

$$\sum_{\mathbf{s}} |\mathbf{s}\rangle\langle\mathbf{s}| = \mathbb{I}, \tag{54}$$

and using the Born-rule probability distribution

$$p_\theta(\mathbf{s}) = \frac{|\psi_\theta(\mathbf{s})|^2}{\sum_{\mathbf{s}'} |\psi_\theta(\mathbf{s}')|^2}, \tag{55}$$

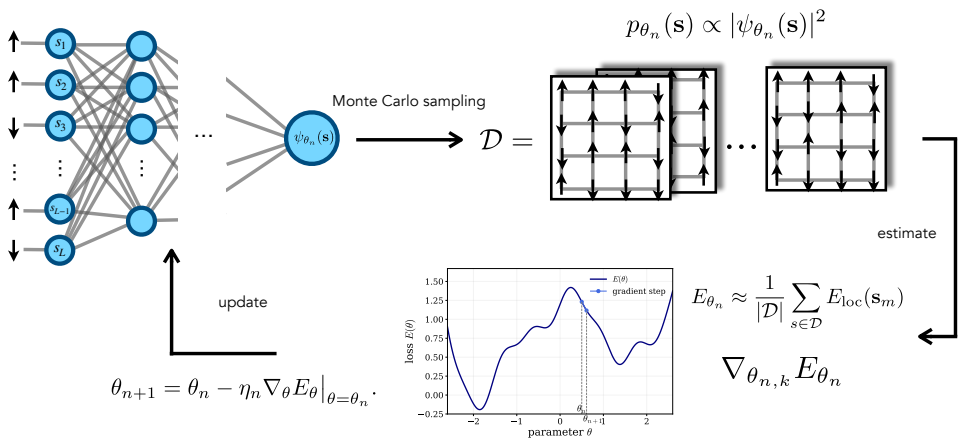

Fig. 3: *Updating the neural network parameters according to the approximative gradient of the energy obtained via Monte Carlo sampling.*

we can rewrite the expectation value as an average over classical configurations:

$$\langle \hat{O} \rangle_\theta = \sum_{\mathbf{s}} p_\theta(\mathbf{s}) O_{\text{loc}}(\mathbf{s}) = \mathbb{E}_{p_\theta} \left[O_{\text{loc}}(\mathbf{s}) \right]. \tag{56}$$

Here we have defined the local observable

$$O_{\text{loc}}(\mathbf{s}) = \sum_{\mathbf{s}'} \langle \mathbf{s} | \hat{O} | \mathbf{s}' \rangle \frac{\psi_\theta(\mathbf{s}')}{\psi_\theta(\mathbf{s})}. \tag{57}$$

This formulation allows us to estimate the quantum expectation value using a classical Monte Carlo average. We first generate M samples $\{\mathbf{s}_1, \ldots, \mathbf{s}_M\}$ from the distribution $p_\theta(\mathbf{s})$, for example using the Metropolis algorithm:
Our quantum state $|\psi_\theta\rangle$ encodes a probability distribution over samples $\mathbf{s}$ given by

$$p_\theta(\mathbf{s}) = \frac{|\langle \mathbf{s} | \psi_\theta \rangle|^2}{\langle \psi_\theta | \psi_\theta \rangle} = \frac{|\psi_\theta(\mathbf{s})|^2}{\sum_{\mathbf{s}'} |\psi_\theta(\mathbf{s}')|^2}. \tag{58}$$

This expression defines a probability distribution over the entire computational basis. For our Neural Quantum State ansatz, $|\psi_\theta\rangle$, the probability of a configuration $\mathbf{s}$ is therefore proportional to $|\psi_\theta(\mathbf{s})|^2$. We can now employ Markov-chain Monte Carlo to draw samples from this quantum mechanical probability distribution. Starting from a configuration $\mathbf{s}$, one proposes a local or nonlocal update $\mathbf{s} \to \mathbf{s}'$ and accepts it with probability

$$A(\mathbf{s} \to \mathbf{s}') = \min \left[1, \frac{|\psi_\theta(\mathbf{s}')|^2}{|\psi_\theta(\mathbf{s})|^2} \frac{T(\mathbf{s}' \to \mathbf{s})}{T(\mathbf{s} \to \mathbf{s}')} \right], \tag{59}$$

where T denotes the proposal probability. After sufficiently many such Monte Carlo proposals and updates, the samples follow the distribution $p_\theta(\mathbf{s})$ as desired. In contrast, autoregressive neural quantum states factorize the probability distribution as

$$p_\theta(\mathbf{s}) = \prod_{i=1}^{N} p_\theta(s_i | s_1, \ldots, s_{i-1}), \tag{60}$$

such that configurations can be sampled directly from the normalized conditional probabilities, without a Markov-chain. This avoids autocorrelation between samples, but imposes an ordering of the degrees of freedom and requires an architecture compatible with the autoregressive factorization. This is often achieved through an explicit masking.
Upon sampling M samples from the distribution $p_\theta(\mathbf{s})$, we compute the sample mean of the local observable

$$\langle \hat{O} \rangle_\theta \approx \frac{1}{M} \sum_{m=1}^{M} O_{\mathrm{loc}}(\mathbf{s}_m). \tag{61}$$

In order to find the ground state, we need to evaluate the expectation value of the Hamiltonian $\hat{H}$, which defines the variational energy, $E_\theta = \langle \hat{H} \rangle_\theta$. The corresponding local observable is the local energy,

$$E_{\mathrm{loc}}(\mathbf{s}) = \sum_{\mathbf{s}'} H_{\mathbf{s},\mathbf{s}'} \frac{\psi_\theta(\mathbf{s}')}{\psi_\theta(\mathbf{s})}, \tag{62}$$

where

$$H_{\mathbf{s},\mathbf{s}'} = \langle \mathbf{s} | \hat{H} | \mathbf{s}' \rangle. \tag{63}$$

The variational energy can therefore be estimated as

$$E_\theta = \mathbb{E}_{p_\theta} \left[E_{\mathrm{loc}}(\mathbf{s}) \right] \approx \frac{1}{M} \sum_{m=1}^{M} E_{\mathrm{loc}}(\mathbf{s}_m), \qquad \mathbf{s}_m \sim p_\theta(\mathbf{s}). \tag{64}$$

To evaluate the local energy for a given sample $\mathbf{s}$, one only needs the configurations $\mathbf{s}'$ connected to $\mathbf{s}$ by nonzero Hamiltonian matrix elements $H_{\mathbf{s},\mathbf{s}'}$, together with the corresponding wavefunction ratios

$$\frac{\psi_\theta(\mathbf{s}')}{\psi_\theta(\mathbf{s})}. \tag{65}$$

3.1.1 Simple ground state search

For a ground state search, we want to minimize the energy, i.e., our cost function is E_θ. In order to optimize the network parameters θ, we must compute its gradient with respect to each variational parameter θ_k. The analytic expression for this gradient, often derived using the log-derivative trick, is given by a covariance

$$\nabla_{\theta_k} E_\theta = 2\mathrm{Re} \left(\mathbb{E}_{p_\theta}[E_{\mathrm{loc}} \mathcal{O}_k^*] - \mathbb{E}_{p_\theta}[E_{\mathrm{loc}}] \mathbb{E}_{p_\theta}[\mathcal{O}_k^*] \right). \tag{66}$$

This introduces a new quantity, $\mathcal{O}_k(\mathbf{s})$, which is the logarithmic derivative of the wavefunction

$$O_k(\mathbf{s}) = \frac{1}{\psi_\theta(\mathbf{s})} \frac{\partial \psi_\theta(\mathbf{s})}{\partial \theta_k} = \partial_{\theta_k} \log \psi_\theta(\mathbf{s}). \tag{67}$$

For a neural network, this derivative is readily available via backpropagation. It measures how sensitively the log-amplitude changes under a variation of parameter θ_k.

Crucially, the gradient can be estimated from the same set of M samples used to compute the energy

$$\nabla_{\theta_k} E_\theta \approx 2\mathrm{Re}\left(\frac{1}{M}\sum_{i=1}^{M} E_{\mathrm{loc}}(\mathbf{s}_i)\mathcal{O}_k^*(\mathbf{s}_i) - \left(\frac{1}{M}\sum_{i=1}^{M} E_{\mathrm{loc}}(\mathbf{s}_i)\right)\left(\frac{1}{M}\sum_{i=1}^{M} \mathcal{O}_k^*(\mathbf{s}_i)\right)\right) \tag{68}$$

With the ability to compute this stochastic gradient, we can employ standard gradient-based optimization algorithms, such as Stochastic Gradient Descent (SGD) or Adam, to iteratively update the network parameters and minimize the energy

$$\theta \leftarrow \theta - \eta \nabla_\theta E_\theta, \tag{69}$$

where η is the learning rate. This iterative process of sampling, calculating the energy and its gradient, and updating the parameters constitutes the complete VMC optimization loop.

3.1.2 Stochastic reconfiguration

Simple gradient descent updates the parameters according to the Euclidean gradient in parameter space. However, the physically relevant distance is not the Euclidean distance between parameter vectors, but the distance between the corresponding quantum states. Stochastic reconfiguration (SR), an established tool in the variational Monte Carlo community, takes this geometry into account. The basic idea of SR is to perform an approximate imaginary time evolution [48] and thereby ultimately projecting onto the ground state: Let $\{|n\rangle\}$ be the eigenstates of the Hamiltonian,

$$\hat{H}|n\rangle = E_n|n\rangle, \qquad E_0 \leq E_1 \leq E_2 \leq \cdots . \tag{70}$$

Starting from an initial state with nonzero overlap with the ground state,

$$|\psi(0)\rangle = \sum_n c_n|n\rangle, \qquad c_0 \neq 0, \tag{71}$$

imaginary-time evolution gives

$$|\psi(\tau)\rangle = e^{-\tau\hat{H}}|\psi(0)\rangle = \sum_n c_n e^{-\tau E_n}|n\rangle. \tag{72}$$

After factoring out the ground-state contribution,

$$|\psi(\tau)\rangle = e^{-\tau E_0}\left[c_0|0\rangle + \sum_{n>0} c_n e^{-\tau(E_n - E_0)}|n\rangle\right]. \tag{73}$$

Thus all excited-state contributions are exponentially suppressed relative to the ground state. After normalization,

$$\frac{|\psi(\tau)\rangle}{\||\psi(\tau)\rangle\|} \xrightarrow{\tau\to\infty} |0\rangle. \tag{74}$$

Stochastic reconfiguration can be understood as projecting this imaginary-time evolution onto the variational manifold spanned by the neural quantum state ansatz.

For a small parameter update $\delta\theta$, we linearize the variational state as

$$|\psi_{\theta+\delta\theta}\rangle \approx |\psi_\theta\rangle + \sum_k \delta\theta_k \partial_{\theta_k} |\psi_\theta\rangle. \tag{75}$$

Imaginary-time evolution, on the other hand, changes the state in linear approximation as

$$|\psi_\theta\rangle \longrightarrow \left(1 - \delta\tau \hat{H}\right)|\psi_\theta\rangle. \tag{76}$$

Stochastic reconfiguration chooses $\delta\theta$ such that the variational change of the wavefunction best approximates this imaginary-time step. This leads to the linear system[5]

$$\sum_l S_{kl}\, \delta\theta_l = -\delta\tau\, F_k, \tag{77}$$

where

$$S_{kl} = \langle O_k^* O_l \rangle_c = \langle O_k^* O_l \rangle - \langle O_k^* \rangle \langle O_l \rangle \tag{78}$$

is the quantum geometric tensor, or covariance matrix of logarithmic derivatives, and

$$F_k = \langle O_k^* E_{\mathrm{loc}} \rangle_c = \langle O_k^* E_{\mathrm{loc}} \rangle - \langle O_k^* \rangle \langle E_{\mathrm{loc}} \rangle . \tag{79}$$

Here $O_k(\mathrm{s})$ is the logarithmic derivative as before. The SR update therefore reads

$$\delta\theta = -\delta\tau\, S^{-1} F. \tag{80}$$

Equivalently, SR can be viewed as a natural-gradient method on the variational manifold. The quantum geometric tensor or S-matrix measures how much the physical quantum state changes when one changes the parameters θ_k. Compared to ordinary gradient descent, the SR update therefore takes into account how sensitively the quantum state changes under variations of the parameters. Note that, as SR for ground state search is based on imaginary time evolution, one can use the same derivation to simulate a real time evolution by simply inserting the complex i in the SR equation.

Historically, SR was developed in the variational Monte Carlo community before the use of neural quantum states, where typically only a few variational parameters are used [48]. Neural networks on the other hand often operate with orders of magnitude more variational parameters, such that the inversion of the S matrix can become computationally prohibitive. This has motivated approximate or iterative variants of SR, including minimum-step stochastic reconfiguration (minSR) [49] and related linear-algebra reformulations of SR [51], which aim to retain the geometric advantages of SR while scaling to modern deep NQS architectures.

Building on these scalable formulations of stochastic reconfiguration, several recent works have adapted ideas from modern optimization to the SR update. The SPRING optimizer (Subsampled Projected-Increment Natural Gradient Descent) combines the minimum-step formulation of SR with randomized Kaczmarz-type updates, using subsampled projected increments to accelerate the solution of the SR linear system [52]. A related direction is the MARCH optimizer

[5] For a detailed derivation, see e.g. Refs. [48–50].

(Moment-Adaptive ReConfiguration Heuristic), which further incorporates first- and second-moment information, in analogy with the way Adam extends momentum-based gradient descent. In practice, MARCH can be viewed as an adaptive-momentum variant of SR/MinSR designed to improve stability and convergence in large-scale NQS optimizations [53]. These scalable formulations address the cost of solving the SR equations, but they do not remove the need for regularization: in practice, small singular values or noisy directions are still damped or truncated to stabilize the update.

3.2 Trainability and design choices

The fact that a given network architecture has the capability to represent a certain class of quantum states, such as volume law entangled states, does not necessarily imply the conclusion that a specific or even any volume law entangled state can be found upon training the neural quantum state. As a concrete example, for restricted Boltzmann machines, Ref. [5] showed that any probability distribution over discrete variables, and therefore the wavefunction amplitudes for any quantum state, can be approximated arbitrarily well. However, such an approximation might require the network weights to become extremely large, which can be very hard to optimize to [24].
In general, beyond theoretical considerations as discussed above, these questions can be extremely hard to separate: if we perform variational Monte Carlo for example to find the ground state of a system and do not converge to the true ground state, is the reason the limited capability of the ansatz, or the challenging optimization landscape which keeps us from finding the network parameters encoding the desired quantum state? Below we discuss some design choices as well as possible routes towards convergence.

Reference basis.– First, one must choose a reference basis. Since the network represents amplitudes $\psi_\theta(\mathrm{s})$ in this basis, the complexity of the amplitudes and phases, as well as the optimization problem, can depend strongly on the basis choice. A basis close to an approximate eigenbasis can make the wavefunction compact, with most weight concentrated on a reference configuration and physically interpretable excitations [54]. This compactness, however, is not automatically advantageous for a generic NQS. A very peaked distribution can make sampling difficult, since rare configurations may still contribute substantially to the local energy or gradient. This sampling issue is closely related to the support-mismatch problem discussed in Ref. [55]: even if a wavefunction is compact in a given basis, standard VMC estimators can become unstable or biased when the sampling distribution fails to cover configurations that are needed to evaluate local energies, gradients, or time-evolution updates. A peaked distribution can also make optimization difficult: representing near-zero amplitudes over a large part of Hilbert space often requires large logits, weights, or biases, which may be hard to reach in practice and can lead to ill-conditioned updates. A flatter basis distribution can therefore be preferable, even if it is less compact, because the required correlations can be encoded with lower-order terms and more moderate parameter values. This viewpoint is consistent with anal-

yses of NQS basis dependence in terms of correlator, cluster, or cumulant expansions, where the relevant question is not only how localized the probability distribution is, but how simple the amplitude and phase structure are in the chosen basis [24, 56].

A further complication is that basis choice can affect trainability even when the target state remains representable. Kožić et al. showed in a rotated Ising model that local basis rotations can relocate the exact wavefunction within the variational loss landscape, increasing its geometric distance from typical initializations and making optimization more likely to become trapped in saddle-point or high-curvature regions. Thus, basis dependence is not only a question of expressivity or sampling, but also of the geometry of the optimization landscape [57]. The best basis is therefore not universally the most localized or the flattest one, but the one in which the target state is simple for the chosen architecture, sampling strategy, and optimizer.

Reference sign or phase structure.– Second, one may build in a reference sign or phase structure. For real wavefunctions it is often useful to separate amplitudes and signs,

$$\psi(\mathbf{s}) = |\psi(\mathbf{s})|\, S(\mathbf{s}), \qquad S(\mathbf{s}) \in \{-1, +1\}. \tag{81}$$

More generally, for complex wavefunctions one may separate amplitude and phase. The sign or phase structure can be the most difficult part of the wavefunction to learn. Supervised studies of frustrated magnets found that learning the signs generalizes much worse than learning the amplitudes [58], and variational studies of the frustrated J_1-J_2 model showed that optimization can get stuck in low-energy states with an incorrect or overly simple sign structure [59, 60]. A common strategy is therefore to build in part of the sign structure explicitly. For bipartite antiferromagnets, the Marshall sign rule provides such a reference sign, and it has often been used either as an explicit gauge transformation or as a prior to stabilize NQS optimizations [61, 51]. This strategy is powerful when the reference sign is accurate, but it can also bias the optimization if the true ground-state sign structure deviates from it.

For doped or fermionic lattice models, the sign structure becomes even more subtle. In Fock space, the basis states already contain the fermionic operator ordering. The neural network represents the coefficients in this fixed Fock basis. Thus, in a second-quantized formulation, a determinant-based ansatz should not be described as "enforcing antisymmetry" of the coefficient function under particle exchange. Rather, the determinant provides a structured sign and nodal pattern for the Fock-space amplitudes, inherited from the underlying single-particle orbitals or from their configuration-dependent generalizations. This can be a useful physical prior for fermionic systems, as seen in determinant-based NQS for the t-J model [31]. At the same time, studies of autoregressive RNNs for doped antiferromagnets show that the difficulty is not only the fermionic sign: doping also enlarges the Hilbert space and can make both amplitude and phase distributions substantially more complex [62]. More recently, Boolean Fourier analysis has provided another perspective on sign structures by expanding $S(\mathrm{s})$ in parity functions,

$$S(\mathbf{s}) = \sum_{I \subseteq \{1,\ldots,N\}} a_I \prod_{i \in I} s_i. \tag{82}$$

In this language, sign learning is related to identifying the relevant high-order parity features. This suggests that difficult sign structures may not be structureless, but may require architectures or input features adapted to the appropriate correlators [63].

Symmetries.– Third, symmetries can be imposed either by the architecture or by explicit projection. Enforcing particle-number conservation, spin symmetries, translation invariance, point-group symmetries, or gauge constraints reduces the effective variational space and can improve both expressivity and trainability. Autoregressive models can, e.g., be constrained to sample only configurations with the desired quantum numbers. In particular, $U(1)$-symmetric RNNs preserve the autoregressive structure while restricting the samples to a fixed magnetization or particle-number sector [64–66]. Spatial symmetries can be incorporated by symmetrizing the wavefunction over the symmetry group, by targeting fixed quantum-number sectors, or by using equivariant architectures such as group convolutional networks [67, 68]. The practical tradeoff can be computational: explicit projection over a symmetry group often requires evaluating the ansatz on all symmetry-related configurations, increasing the cost by the size of the group. For determinant- or Pfaffian-based fermionic states this overhead can be particularly relevant, since each symmetry image may require an additional determinant or Pfaffian evaluation. Built-in equivariance can reduce the overhead of explicit symmetry projection by enforcing the symmetry at the architectural level. For example, group convolutional neural networks implement lattice symmetries directly through equivariant layers and have been shown to improve NQS accuracy without increasing memory use [68]. In practice, the best-performing NQS are often not generic neural networks, but those in which physical structure is built in from the start.

Initialization.– In order to ease the optimization problem, one can initialize the network parameters in a favorable place in the optimization landscape. This can be done by setting the network weights according to the entries of a matrix product state [69], which can be efficiently optimized within the capacities of MPS using the DMRG. While this is in general an intriguing option, it can in practice limit the architectural choices of the neural quantum state. Another option is to first perform a state reconstruction task by using data from a quantum simulator. If the available data are projective measurements in one basis, that basis can be chosen as the reference basis of the NQS and the Kullback-Leibler divergence or Wasserstein metric can be minimized to match the probability distribution given by the NQS with the measured data distribution [70–72]. Going beyond snapshots in one basis is possible by either considering a few local rotations, which can still be computed [73], or by additionally performing a minimization to match global correlations in one or more bases beyond the reference basis [72]. These works have demonstrated that improved convergence of the subsequent ground state search can be achieved if the NQS is initialized using data from a state that is close to the ground state. This advantage remains when using experimental data, where state preparation and measurement imperfections can yield a data set significantly different from the ideal ground state. In the absence of numerical or experimental data, one can also choose the random distribution and its width in a favorable manner as discussed in Sec. 2.2, see also e.g. Ref. [46].

Quantum geometric tensor.– One perspective on the complexity of the optimization landscape is provided by the quantum geometric tensor of the variational manifold (the S-matrix) introduced in the context of stochastic reconfiguration above. Its spectrum gives local information about which parameter directions correspond to physically distinct changes of the quantum state. Large eigenvalues correspond to directions in which the state changes strongly, while small eigenvalues correspond to flat, redundant, or poorly sampled directions. In the SR update, Eq. (80), the energy gradient is preconditioned with the inverse of S, so directions in which a small parameter change produces a large change of the quantum state are correspondingly rescaled.

The effective rank of the quantum geometric tensor gives an estimate of the number of linearly independent directions in Hilbert space that are accessible by infinitesimal changes of the variational parameters and can thus be used as a diagnostic of the practical representational power of an ansatz. If the rank saturates as the network is widened, additional parameters are largely redundant for the target state. Note however that the relevant geometry of the variational manifold evolves during training: The spectrum of the QGT can also change substantially during training and across physical regimes, as observed for complex RBM wavefunctions in Ref. [74] and for vision transformers and hidden fermion determinant states in [47]. The rank of the quantum geometric tensor should not be interpreted as a global measure of the expressivity of the ansatz. Rather, it measures the dimension of the local tangent space of physically distinct wavefunction variations at the current parameter point. Since this tangent space changes during training, the QGT rank is a local and trajectory-dependent diagnostic. It is nevertheless useful in practice: if increasing the number of parameters no longer increases the effective QGT rank near the optimized state, and the variational accuracy saturates at the same time, then the additional parameters are largely redundant for the state being represented [75]. Conversely, a growing QGT rank indicates that the enlarged ansatz provides additional independent directions in Hilbert space that the optimizer can use. In Ref. [75], the width of the network is increased at fixed depth to increase the number of parameters. From the QGT point of view, increasing width and increasing depth explore different extensions of the variational manifold. Width adds more tangent vectors of the same structural form, whereas depth changes the non-linear map itself, as discussed in Sec. 2.2.1, and can therefore modify the tangent space qualitatively. A saturation of the QGT rank under widening indicates local redundancy within that family, but it does not rule out an increase of the effective rank, or an improvement in accuracy, under a change of architecture such as adding depth.

The matrix S describes the geometry of the variational manifold, but it does not by itself determine the update. The second ingredient is the force vector F_k, which measures the overlap between the tangent vector $\partial_{\theta_k}|\psi_\theta\rangle$ and the imaginary-time direction $-(\hat{H}-E)|\psi_\theta\rangle$. Thus F tells us which directions in the tangent space are energetically useful, while S tells us how these tangent vectors overlap with one another. The SR equation

$$S\,\delta\theta = -\eta F \tag{83}$$

therefore finds the linear combination of tangent vectors that best approximates the imaginary-time update within the variational manifold. Convergence depends not only on the rank of

S, but also on whether the force vector has support in well-conditioned directions of S. If F points mostly into small-eigenvalue or noisy directions, the SR update becomes unstable and regularization is required.
In combination with the variational error and the stochastic-reconfiguration force vector, the QGT spectrum provides a useful diagnostic of whether convergence is limited by insufficient local expressivity, ill-conditioning, or by the global optimization landscape [74, 75].

4 Concluding remarks

Neural quantum states provide a flexible variational ansatz for strongly correlated quantum systems. Their appeal lies in the possibility of combining neural-network expressivity with physical structure, such as locality, symmetries, reference signs, Jastrow factors, determinants, Pfaffians, or autoregressive factorizations. At the same time, expressivity alone is not sufficient: an architecture may be capable of representing a target state, but the corresponding parameters may still be difficult to find in practice. More broadly, understanding how expressive different NQS architectures really are, if and when they work well in practice, and why particular design choices succeed or fail remains an active area of research. These notes aim to provide some intuition for these questions by separating capability, typicality, and trainability, and by illustrating how architectural ingredients such as nonlinearities, connectivity, symmetries, basis choice, and fermionic determinant structures affect the states that can be represented and optimized. The usefulness of an NQS therefore depends on the interplay between capability, typicality, trainability, sampling, and the chosen reference basis. This is both a challenge and an opportunity. Rather than searching for a universally optimal neural network ansatz, current progress suggests that the most successful approaches are those in which machine-learning flexibility is combined with problem-specific physical insight. Neural quantum states should therefore be viewed not as a replacement for traditional variational ansätze, but as a framework that can extend them in a highly expressive form.

Acknowledgments

I am grateful to the many people from whom I learned and with whom I discussed all neural quantum state related aspects. I want to in particular thank (in alphabetical order) Julian Blasek, Annika Böhler, Francesco Conoscenti, Fabian Döschl, Hannah Lange, Roger Melko, Balint Soczo, and Anka van de Walle, with whom I gained a deeper understanding of many of the things presented in these notes. I gratefully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC-2111-390814868, by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme ERC Starting Grant QuaQuaMA (Grant No. 101217531), and by LMUexcellent, funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder.

References

[1] U. Schollwöck, Rev. Mod. Phys. **77**, 259 (2005)

[2] G. Carleo and M. Troyer, Science **355**, 602 (2017)

[3] G. Cybenko, Math. Control Signals Syst. **2**, 303 (1989)

[4] K. Hornik, M. Stinchcombe, and H. White, Neural Netw. **2**, 359 (1989)

[5] N.L. Roux and Y. Bengio, Neural Comput. **20**, 1631 (2008)

[6] M. Medvidović and J.R. Moreno, Eur. Phys. J. Plus **139** (2024)

[7] H. Lange, A.V. de Walle, A. Abedinnia, and A. Bohrdt, Quantum Sci. Technol. **9**, 040501 (2024)

[8] M. Schmitt and M. Heyl, arXiv 2506.03124 (2025)

[9] D.-L. Deng, X. Li, and S. Das Sarma, Phys. Rev. X **7** (2017)

[10] Z. Denis, A. Sinibaldi, and G. Carleo, Phys. Rev. Lett. **134** (2025)

[11] A.Y. Kitaev, Ann. Phys. **303**, 2 (2003)

[12] M.Y. Pei and S.R. Clark, J. Phys. A: Math. Theor. **54**, 405304 (2021)

[13] S.R. Clark, J. Phys. A: Math. Theor. **51**, 135301 (2018)

[14] R. Kaubruegger, L. Pastori, and J.C. Budich, Phys. Rev. B **97**, 195136 (2018)

[15] I. Glasser, N. Pancotti, M. August, I.D. Rodriguez, and J.I. Cirac, Phys. Rev. X **8** (2018)

[16] P. Chen, B. Yan, and S.X. Cui, Phys. Rev. B **111** (2025)

[17] Y. Huang and J.E. Moore, Phys. Rev. Lett. **127**, 170601 (2021)

[18] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, Phys. Rev. B **97**, 085104 (2018)

[19] O. Sharir, A. Shashua, and G. Carleo, Phys. Rev. B **106**, 205136 (2022)

[20] D. Wu, R. Rossi, F. Vicentini, and G. Carleo, Phys. Rev. Res. **5** (2023)

[21] F. Becca and S. Sorella: *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press, 2017)

[22] H. Lange: *New approaches to strongly correlated electron systems* (PhD Thesis, LMU München, 2026)

[23] J. Robledo Moreno, G. Carleo, A. Georges, and J. Stokes, Proc. Natl. Acad. Sci. USA **119** (2022)

[24] F. Döschl and A. Bohrdt, arXiv:2508.14152 (2025)

[25] Z. Liu and B.K. Clark, Phys. Rev. B **110**, 115124 (2024)

[26] Y. Gu, Z. Han, W. Li, Z. Xiao, T. Xiang, M. Qin, L. Wang, and D. Lv, arXiv (2026)

[27] Y. Nomura, A.S. Darmawan, Y. Yamaji, and M. Imada, Phys. Rev. B **96**, 205152 (2017)

[28] S. Humeniuk, Y. Wan, and L. Wang, SciPost Phys. **14**, 171 (2023)

[29] D. Luo and B.K. Clark, Phys. Rev. Lett. **122**, 226401 (2019)

[30] A. Chen, Z.-Q. Wan, A. Sengupta, A. Georges, and C. Roth, arXiv (2025)

[31] H. Lange, A. Böhler, C. Roth, and A. Bohrdt, Phys. Rev. Lett. **135**, 136504 (2025)

[32] H. Lange, A. Chen, A. Georges, F. Grusdt, A. Bohrdt, and C. Roth, arXiv:2602.10091

[33] H. Lange, J. Tirpitz, and A. Bohrdt, in preparation (2026)

[34] C. Gauvin-Ndiaye, J. Tindall, J.R. Moreno, and A. Georges,
Phys. Rev. Lett. **134**, 076502 (2025)

[35] D. Pfau, J.S. Spencer, A.G. d.G. Matthews, and W.M.C. Foulkes,
Phys. Rev. Res. **2**, 033429 (2020)

[36] J. Hermann, Z. Schätzle, and F. Noé, Nat. Chem. **12**, 891 (2020)

[37] J. Kim, G. Pescia, B. Fore, J. Nys, G. Carleo, S. Gandolfi, M. Hjorth-Jensen, and
A. Lovato, Commun. Phys. **7**, 148 (2024)

[38] L. Fu: *Fermi Sets: Universal and interpretable neural architectures for fermions* (2026)

[39] L. Zhang, G. Duan, and D. Luo, arXiv:2605.29683 (2026)

[40] M.B. Hastings, I. González, A.B. Kallin, and R.G. Melko,
Phys. Rev. Lett. **104**, 157201 (2010)

[41] P. Mehta, M. Bukov, C.-H. Wang, A.G. Day, C. Richardson, C.K. Fisher, and D.J. Schwab,
Phys. Rep. **810**, 1 (2019)

[42] N. Paul, Phys. Rev. Lett. **136**, 120403 (2026)

[43] X. Gao and L.-M. Duan, Nat. Commun. **8** (2017)

[44] B. Swingle, Phys. Rev. Lett. **105**, 050502 (2010)

[45] G. Passetti, D. Hofmann, P. Neitemeier, L. Grunwald, M.A. Sentef, and D.M. Kennes,
Phys. Rev. Lett. **131** (2023)

[46] A. Jreissaty, H. Zhang, J.C. Quijano, J. Carrasquilla, and R. Wiersema, Phys. Rev. Res. **8** (2026)

[47] F. Conoscenti, H. Lange, F. Döschl, C. Kuhlenkamp, J. Carrasquilla, and A. Bohrdt, in preparation (2026)

[48] S. Sorella, Phys. Rev. Lett. **80**, 4558 (1998)

[49] A. Chen and M. Heyl, Nat. Phys. **20**, 1476 (2024)

[50] A. Chen: *Large-scale simulation of deep neural quantum states* (PhD Thesis, Universität Augsburg, 2025)

[51] R. Rende, L.L. Viteritti, L. Bardone, F. Becca, and S. Goldt, Commun. Phys. **7**, 260 (2024)

[52] G. Goldshlager, N. Abrahamsen, and L. Lin, arXiv (2024)

[53] Y. Gu, W. Li, H. Lin, B. Zhan, R. Li, Y. Huang, D. He, Y. Wu, T. Xiang, M. Qin, L. Wang, and D. Lv, arXiv (2025)

[54] J.A. Sobral, M. Perle, and M.S. Scheurer, Nat. Commun. **16** (2025)

[55] Z.-Q. Wan, R. Wiersema, and S. Zhang, arXiv:2603.18148 (2026)

[56] R.S. Cortes Santamaria, A.S. Shankar, M. Dalmonte, R. Verdel, and N. Niggemann, SciPost Phys. Core **9** (2026)

[57] S.B. Kožić, V. Zlatić, F. Franchini, and S.M. Giampaolo, arXiv:2512.17893 (2025)

[58] T. Westerhout, N. Astrakhantsev, K.S. Tikhonov, M.I. Katsnelson, and A.A. Bagrov, Nat. Commun. **11**, 1593 (2020)

[59] M. Bukov, M. Schmitt, and M. Dupont, SciPost Phys. **10**, 147 (2021)

[60] A. Szabó and C. Castelnovo, Phys. Rev. Res. **2**, 033075 (2020)

[61] W. Marshall, Proc. Roy. Soc. A **232**, 48 (1955)

[62] H. Lange, F. Döschl, J. Carrasquilla, and A. Bohrdt, Commun. Phys. **7**, 187 (2024)

[63] I. Schurov, A. Kravchenko, M.I. Katsnelson, A.A. Bagrov, and T. Westerhout, arXiv:2508.09870 (2025)

[64] M. Hibat-Allah, M. Ganahl, L.E. Hayward, R.G. Melko, and J. Carrasquilla, Phys. Rev. Res. **2**, 023358 (2020)

[65] S. Morawetz, I.J.S. De Vlugt, J. Carrasquilla, and R.G. Melko, Phys. Rev. A **104**, 012401 (2021)

[66] A. Malyshev, J.M. Arrazola, and A.I. Lvovsky, arXiv (2023)

[67] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Phys. Rev. Lett. **121**, 167204 (2018)

[68] C. Roth and A.H. MacDonald, arXiv:2104.05085 (2021)

[69] R. Kaneko and S. Goto, Phys. Rev. B **112**, 155163 (2025)

[70] S. Czischek, M.S. Moss, M. Radzihovsky, E. Merali, and R.G. Melko, Phys. Rev. B **105** (2022)

[71] M.S. Moss, S. Ebadi, T.T. Wang, G. Semeghini, A. Bohrdt, M.D. Lukin, and R.G. Melko, Phys. Rev. A **109**, 032410 (2024)

[72] H. Lange, G. Bornet, G. Emperauger, C. Chen, T. Lahaye, S. Kienle, A. Browaeys, and A. Bohrdt, Quantum **9**, 1675 (2025)

[73] E.R. Bennewitz, F. Hopfmueller, B. Kulchytskyy, J. Carrasquilla, and P. Ronagh, Nat. Mach. Intell. **4**, 618 (2022)

[74] C.-Y. Park and M.J. Kastoryano, Phys. Rev. Res. **2**, 023232 (2020)

[75] S. Dash, L. Gravina, F. Vicentini, M. Ferrero, and A. Georges, Commun. Phys. **8**, 92 (2025)