



UNIVERSITAT_{DE}
BARCELONA

MASTER THESIS
MASTER IN QUANTUM SCIENCE AND TECHNOLOGY



Machine learning for self-bound quantum systems

Student: Jesús Mosteiro García
Supervisors: Arnau Rios Huguet
Javier Rozalén Sarmiento
Barcelona, July 7th 2024.

Machine learning for self-bound quantum systems

Jesús Mosteiro García

Supervised by: Arnau Rios Huguet, Javier Rozalén Sarmiento

Institute of Cosmos Sciences, University of Barcelona

July 7th, 2024

In this work we compute the ground-state properties of one-dimensional systems composed of fully-polarized fermions interacting through an attractive Gaussian potential. If the interactions are able to overcome kinetic energy contributions, these particles can become bound without the presence of an external potential. We use Neural Quantum States, a technique which exploits deep neural networks as the ansatz for a Variational Monte Carlo. We adapt a previously presented network architecture, that was designed to enforce fermionic antisymmetry, to also include a mean-centering transformation. This fixes the system at the origin of space, overcoming numerical instabilities. Transfer learning from ansätze trained on systems trapped with an external potential is used to facilitate reaching a global energy minimum. We predict analytically and confirm computationally that, under certain conditions, these systems are fully characterized by their scattering length, giving rise to what we call “1D fermionic halo systems”. Our results suggest that the ground state energy decreases linearly with the amount of fermions and has an inverse quadratic dependence on the scattering length.

Keywords: Machine Learning, Neural Quantum States, Self-bound Quantum Systems, Deep Neural Networks, Variational Monte Carlo

Acknowledgements

My thanks go out to Arnau, Javier, James, Joan, and Alejandro for their invaluable help throughout this project. Without their support, it would not have been possible. I would also like to thank my Mom, Dad, and Sister, as well as Adri, Jan, Román, and David. For making life a little easier,

Thank you.

Jesús Mosteiro García: jmostega8@alumnes.ub.edu jesus.mosteiro@udc.es

Contents

1	Introduction	1
2	Theoretical Background	2
2.1	Problem Hamiltonian	2
2.2	Artificial Neural Networks	4
2.3	Neural Quantum States	5
2.4	Monte Carlo integration	5
3	Implementation	6
3.1	Ansatz architecture	6
3.2	Loss function	8
3.3	The RAdamW optimizer	9
3.4	Transfer learning	10
3.5	Mean-centering transformation	11
4	Results	13
4.1	Energy	13
4.2	Density profiles	15
4.3	One-body density matrices	16
4.4	Pair correlation functions	17
4.5	The importance of transfer learning	18
5	Conclusions and outlook	19
	Bibliography	20
A	Relative coordinate derivation	22

1 Introduction

Many problems in quantum many-body physics cannot be approached analytically. The complexity of numerical solutions often scales with the dimensionality of the Hilbert space involved, which in turn scales exponentially with the size of the system [Vid04]. This has become a barrier in the development of computational approaches to many-body problems. As such, methods that provide approximate solutions while lowering the required computational complexity are of great interest in contemporary research.

In recent years, research in the field of machine learning has revolutionized the way we tackle problems as part of numerous other disciplines, such as healthcare, education, marketing, sociology, or finance. While many of these mathematical models have existed for decades, it is only recently that we have acquired the computational resources to be able to apply them to real use cases. The flexibility of these models has also allowed us to make use of them to approach questions in physics [CCC⁺19].

In particular, it is the possibility of optimizing outputs based on a metric of goodness, without requiring understanding how the data is being processed, that makes neural networks so versatile for our purposes. They make for ideal ansätze that are able to adapt to a wide range of situations [Hor91], filling gaps in knowledge by modeling functions that we know properties of, but cannot formulate analytically.

In 2017, Giuseppe Carleo and Matthias Troyer introduced a technique known as Neural-network Quantum States (NQS) [CT17], which makes use of a neural network as the ansatz for a Variational Monte Carlo (VMC) optimization to find the ground state of a quantum many-body system. Since then, this method has shown very promising results [LVdWAB24], at times yielding more accurate predictions than the state of the art. Not only that, but its great scalability makes it viable for working with many-body simulations that other methods cannot handle.

Through my thesis I set out to continue the research lines of our group regarding Neural-network Quantum States [KDRF⁺23, Azz23, Kee22, RSKR24], taking the lessons learned and applying them to self-bound systems. In such systems, particles are not bound by an external potential, but instead by interactions with each other. This poses difficulties for numerical methods, as there is no preference for an absolute position for the system to reach an energy minimum. When attempting to approach the problem via machine learning, numerical stability is compromised. We explore a possible solution to this conundrum, as well the best approaches to work with systems with an arbitrary amount of particles, and the use of transfer learning as part of the optimization workflow.

As part of this work, we develop an implementation of the described techniques. The code is available as open-source software [MGK24]. Simplicity in development and iteration, as well as ease for integration with existing machine learning libraries, are prioritized over runtime efficiency. For this reason, the programming language of choice is Python: a high-level, object-oriented, interpreted language with tools available to work with models such as this one. In addition, we also aim to analyze the performance of our solution.

This document begins with Section 2 by introducing concepts that will be relevant throughout, and establishing the used methodology. Section 3 explains in detail how these ideas are implemented as part of our work. Section 4 delves into the results obtained. Finally, the conclusions that can be drawn from this thesis are presented in Section 5.

2 Theoretical Background

2.1 Problem Hamiltonian

In this work we treat a set of identical fermions that interact with each other via an interparticle Gaussian interaction in one spatial dimension. The particles are assumed to be fully polarized, and thus their spin can be ignored [RS80]. We focus on the case in which the Gaussian interaction provides enough attraction for the fermions to become self-bound. Due to antisymmetry, there is a certain threshold on the magnitude of the interactions between fermions for this behavior to arise, even in one dimension. First, we formulate a trapped Hamiltonian $\hat{H}_t$ considering an additional harmonic potential. Its purpose is related to our proposed training workflow, which makes use of several stages of transfer learning and is explained in detail as part of Section 3.4. The resulting Hamiltonian reads as follows:

$$\hat{H}_t = -\frac{\hbar^2}{2m} \sum_{i=1}^A \nabla_i^2 + \frac{1}{2} m \omega^2 \sum_{i=1}^A x_i^2 + \sum_{i<j} \frac{V}{\sigma \sqrt{2\pi}} e^{-\frac{(x_i - x_j)^2}{2\sigma^2}}, \quad (1)$$

with m the mass of each particle, ω the frequency of the oscillator, V the interaction strength, and σ the interaction range.

For convenience, employ Harmonic Oscillator (HO) units. The interaction constants are replaced by their dimensionless form following $\sigma_0 = \sigma/a_{\text{HO}}, V_0 = V/(a_{\text{HO}}\hbar\omega)$; with $a_{\text{HO}} = \sqrt{\frac{\hbar}{m\omega}}$ the natural length of the oscillator. Then,

$$\hat{H}_t = -\frac{1}{2} \sum_{i=1}^A \nabla_i^2 + \frac{1}{2} \sum_{i=1}^A x_i^2 + \sum_{i<j} \frac{V_0}{\sigma_0 \sqrt{2\pi}} e^{-\frac{(x_i - x_j)^2}{2\sigma_0^2}}. \quad (2)$$

When removing the potential term from the Hamiltonian, our natural units no longer depend on the oscillator. Instead, $V_0 = V/\hbar$, and σ requires no adjustment:

$$\hat{H} = -\frac{1}{2} \sum_{i=1}^A \nabla_i^2 + \sum_{i<j} \frac{V_0}{\sigma \sqrt{2\pi}} e^{-\frac{(x_i - x_j)^2}{2\sigma^2}}. \quad (3)$$

The Gaussian interaction is chosen due to its simplicity for integration, and the relevance of short-range interactions in the context of nuclear physics [RS80, SGM19, BDZ08]. The two constants parametrizing this interaction are its strength V_0 , and distance σ . For a self-bound system, we consider an attractive interaction in which $V_0 < 0$. In the limit when $\sigma \rightarrow 0$, this normalized Gaussian potential is equivalent to a contact interaction. Having an expression that is easy to work with (both numerically and analytically) opens the possibility to accurately benchmark via methods that would otherwise struggle computing many-body simulations [RFPJD20, SV98]. It is worth noting that this model for the interaction behaves in the zero-limit $\sigma \rightarrow 0$ as $V(x) \approx \delta(x)$. In contrast, previous literature makes use of odd-parity interactions, for which this limit results in behavior following $V(x) \approx \overleftarrow{\partial}_x \delta(x) \overrightarrow{\partial}_x$. For spinless fermions, this is known to be the correct limit [Še86], and it cannot be reproduced with a Gaussian potential such as the one used in this work. For this reason, our results should be taken as an analysis of a toy model, and we focus on developing methodological knowledge that to be extrapolated into more practical models.

To help us understand the problem, as well as to obtain some benchmark predictions, we now briefly look at the analysis of a simplified case. We limit ourselves to a two particle system with a square well interaction, as opposed to the Gaussian discussed so far.

The Hamiltonian in natural units and with respect to the relative coordinate $r = \frac{x_1 - x_2}{\sqrt{2}}$, following Busch et al. [BERW98], in this case is

$$\hat{H}_{\text{sqr}} = -\frac{\nabla_r^2}{2} + \Theta_\sigma(r), \quad (4)$$

where $\Theta_R(x_1 - x_2)$ represents our square well potential

$$\Theta_\sigma(x) = \begin{cases} V_0 & \text{if } |x| < \sigma \\ 0 & \text{otherwise} \end{cases}. \quad (5)$$

From the Schrödinger equation for the relative wavefunction $\psi_n(r)$ with energy E_n , we find equations for two different regions of space:

$$\begin{aligned} (\nabla_r^2 + 2E_n - 2V_0)\psi_n(r) &= 0, & |r| < \sigma; \\ (\nabla_r^2 + 2E_n)\psi_n(r) &= 0, & |r| > \sigma. \end{aligned} \quad (6)$$

We are interested in bound states, for which the attractive interaction overcomes the kinetic contribution and $E_n < 0, V_0 < 0$. We introduce the wavevectors $k_n^2 = -2E_n$ and $\kappa^2 = -2V_0$, finding

$$\begin{aligned} (\nabla_r^2 - k_n^2 + \kappa^2)\psi_n(r) &= 0, & |r| < \sigma &\implies \psi_n(r) \propto \sin(\bar{k}_n r); \\ (\nabla_r^2 - k_n^2)\psi_n(r) &= 0, & |r| > \sigma &\implies \psi_n(r) \propto e^{-k_n r}; \end{aligned} \quad (7)$$

for the interior and exterior regions, respectively. The interior solution is determined by the antisymmetry requirement, where $\bar{k}_n^2 = \kappa^2 - k_n^2$ is the momentum of the harmonic. To find the binding momenta k_n , knowing that the logarithmic derivatives of the solutions must coincide at the boundary between regions,

$$\left. \frac{\nabla_r \psi_n(r)}{\psi_n(r)} \right|_{\sigma_-} = \left. \frac{\nabla_r \psi_n(r)}{\psi_n(r)} \right|_{\sigma_+} \implies \frac{\bar{k}_n \cos(\bar{k}_n \sigma)}{\sin(\bar{k}_n \sigma)} = \frac{-k_n e^{-k_n \sigma}}{e^{-k_n \sigma}} \implies \frac{\bar{k}_n}{\tan(\bar{k}_n \sigma)} = -k_n, \quad (8)$$

which gives the following recursive equation:

$$k_n = -\frac{\sqrt{\kappa^2 - k_n^2}}{\tan(\sigma \sqrt{\kappa^2 - k_n^2})}. \quad (9)$$

For this equation to be valid, the tangent must be negative. This sets a minimum threshold for a bound state to exist. Assuming $k_0 \ll \kappa$, the condition is then

$$\sigma \sqrt{-2V_0} > \frac{\pi}{2} \implies V_{\text{min}} = \frac{\pi^2}{8\sigma^2}. \quad (10)$$

This is not the case for non-fermionic 1D systems, as it is a consequence of the antisymmetric solution in the inner region. We define the scattering length a as the scale of the exponential decay of the wavefunction. For the ground state, $a = k_0^{-1}$. This value depends only on V_0 and σ . In cases where $a \gg \sigma$, a fully characterizes the system, universally dominating its behavior irregardless of the chosen combination of $\{V_0, \sigma\}$. This is reminiscent of nuclear halo systems, leading us to denominate the cases where this holds as “1D fermionic halo systems”. The binding energy can then be approximated as

$$E_0 \approx -\frac{1}{2a^2}. \quad (11)$$

2.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a mathematical construct that is capable of portraying a wide variety of functions by combining a high amount of simple operations through extensive parametrization [PG17]. This is achieved by making use of a graph structure in which data flows across a series of nodes called neurons. Each neuron has a numerical value associated to it. This value may depend directly on one of the inputs to the NN (in which case we talk about an input neuron), or is otherwise determined by combining the values of other neurons in the network through operations involving parameters that control the behavior of the model when tweaked. The outputs of the network are provided by a series of so-called output neurons. Depending on how the graph is structured, we refer to different kinds of Neural Network architectures.

One of the simplest kinds of networks are multilayer perceptrons, first introduced in 1958 [Ros58]. This architecture is organized in a series of layers: an input layer (consisting fully of input neurons), a series of hidden layers, and an output layer (consisting fully of output neurons). The layers are ordered sequentially from input to output, and the values of each neuron in a layer depend only on the ones from the previous layer. These values are computed into a weighted sum along with a constant factor called the bias of the neuron. The output of this sum then goes through the activation function of the neuron φ , with the purpose of adding nonlinearity to the result. The computation performed at neuron i in layer l is written as

$$x_{l,i} = \varphi \left(b_{l,i} + \sum_j^{N_{l-1}} w_{l,ij} x_{l-1,j} \right), \quad (12)$$

with φ the activation function, $b_{l,i}$ the bias of the neuron, N_l the number of neurons in layer l , and $w_{l,ij}$ the weight that is applied onto the output of neuron j from the previous layer. Figure 1 illustrates the layered graph architecture of such a perceptron, with input neurons in blue, an output neuron in red, and a white hidden layer. The weights between neurons $w_{l,ij}$ are indicated on the corresponding graph edges.

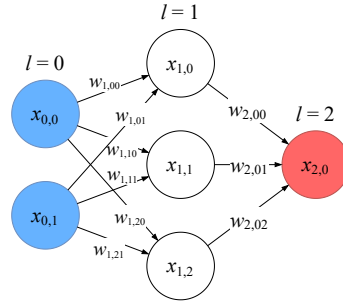


Figure 1: Example perceptron architecture with one hidden layer (in white). In blue are the two input neurons. The output is in red, for our purposes, a probability amplitude. The weights associated to each neuron connection are indicated on the graph edges.

In order for ANNs to be of any use, their parameters must be tweaked to model the function we are trying to approximate. This is achieved via an optimization process usually referred to as “learning”. A key element to this process is being able to quantify the performance of a set of parameters. For this purpose, we define what is called a loss function $\mathcal{L}$, which correlates with the magnitude of the error. Depending on how this loss function is defined, there are two main schemes for training neural networks: supervised

and unsupervised learning. The key difference is that supervised learning defines the loss with respect to a series of known “good” outputs (often labeled manually by experts), while unsupervised learning computes losses without relying on previously solved cases. The VMC methodology used here falls under the umbrella of unsupervised learning.

Comparing the loss yielded by different sets of parameters could become a very demanding task, and attempting to find a minimum this way would be daunting. For this reason, NN optimization is performed via a technique called gradient descent [Ama67, Lin70]. By differentiating the loss function with respect to the parameters of the network, it is possible to update these parameters stepping in a direction that causes the loss to become smaller. Computationally, this is achieved through automatic differentiation [PGC⁺17]. The algorithm used in this work for this purpose is RAdamW, explained in Section 3.3.

2.3 Neural Quantum States

The NQS technique makes use of ANNs as variational ansätze to model quantum states. The network acts as an approximate wavefunction taking as inputs the one-dimensional particle positions x_i . Its output is that of the estimated wavefunction, from which the probability amplitude can be determined. The weights and biases of the network are the tunable parameters that allow the wavefunction to change form as it is trained. In order to guide the optimization of our network, we make use of the Rayleigh-Ritz theorem [Rit09]. From it, we know that given a general parametric ansatz state $|\Psi_\theta\rangle$, with a set of parameters θ ,

$$E_{\text{gs}} \leq \frac{\langle \Psi_\theta | \hat{H} | \Psi_\theta \rangle}{\langle \Psi_\theta | \Psi_\theta \rangle}, \quad (13)$$

where E_{gs} is the ground state energy associated to $\hat{H}$. This inequality holds for any $|\Psi_\theta\rangle$ fulfilling the correct boundary conditions, as well as any other physical requirements to be considered a valid state.

Thus, by setting the loss function to be the expectation of the energy with respect to $\hat{H}$, the NN ansatz can be trained to learn the ground state of the system. Compared to other methods often used to achieve the same result, the versatility of NNs shows promising results in terms of scalability. This comes as a direct consequence of their efficient way of representing approximate wavefunctions. For exact diagonalization, for instance, even the best algorithms known to date take exponential resources with respect to the size of the system [WF08]. This makes them unfeasible for simulating large systems. The NQS method is a much more viable numerical medium through which to execute many-body simulations, as it dodges the exponential growth of the Hilbert space dimension by compressing the state in terms of network parameters [LVdWAB24]. NNs can be seen as universal approximators [Hor91], which also makes them a great resource in cases in which we are unable to narrow down the form of the expected wavefunction. NQS-based techniques have shown successes not only in finding ground [KPSG⁺21] and excited [CCRN18] states, but also in finite temperature [NYN21] and open [HC19] systems, simulating dynamics [SH20], and quantum state tomography [KMH⁺22].

2.4 Monte Carlo integration

In order to estimate the energy expectation of Eq. 13 numerically, we make use of Monte Carlo integration. To do so, we sample the trial wavefunction pseudo-randomly according to the distribution given by its squared probability amplitude $|\Psi_\theta|^2$. Markov Chain Monte

Carlo (MCMC) methods are a suitable, efficient solution for this purpose [JQ22]. They work by constructing a Markov chain for which the equilibrium distribution matches the desired distribution. In particular, the Metropolis-Hastings algorithm [MRR⁺53, Has70] is a form of MCMC that generates candidate samples from a proposal distribution. We use multiple chains, each generated by a different walker and starting at a random position in the high-dimensional space. Evolving them through the Metropolis-Hastings algorithm gets their final iteration samples to more closely match the target distribution, with more steps (or sweeps) improving the approximation. Of course, increasing the sample size by making use of more walkers also increases the accuracy of the estimated integral. With this, we are able to provide a numerical estimation for the energy expectation in Eq. 13 given a Hamiltonian $\hat{H}$ and a trial wavefunction Ψ_θ :

$$E_{\text{gs}} \leq E_\theta = \mathbb{E}_{X \sim |\Psi_\theta|^2} [\Psi_\theta(X)^{-1} \hat{H} \Psi_\theta(X)]. \quad (14)$$

The inequality holds within statistical uncertainties. As the form of Ψ_θ changes when its parameters θ are updated, during training we interweave sweeping the samplers and optimizing the NN. If the two processes are balanced accordingly, this allows the walkers to adapt to the distribution even as the target morphs toward the ground state [Azz23]. To keep training times manageable, we use a limited amount of walkers (4096) and sweeps (20) during optimization. However, more precise sampling is used to obtain a better estimation over the already trained NQS during a final pass.

3 Implementation

3.1 Ansatz architecture

The NQS ansatz used for this work is a slightly modified version of the one presented by J.W.T. Keeble et al. in 2023 [KDRF⁺23]. It is inspired by FermiNet [PSMF20], which made use of Generalized Slater Matrices (GSMs) to ensure fermionic antisymmetry is upheld by the ansatz. This solution combines GSMs with equivariant layers, log-envelope functions, and a summed signed-log determinant function. An additional coordinate transformation which fixes the center of mass at the origin has been added to the proposed ansatz, which is discussed in Section 3.5. The resulting architecture is depicted in Figure 2. This network has A inputs representing the one-dimensional positions of our A fermions in space. Its outputs are the sign of the trial wavefunction and the logarithm of its absolute value.

We show the equivariant layers as part of the network in gray. These are used to enforce permutation equivariance across the network, which in turn ensures the antisymmetry expected in fermionic systems. Within each equivariant layer, an equivariant feature $\mu^{(L)}$ is computed from all of the inputs. This new value is appended to form a pair with each of the inputs, resulting in an $A \times 2$ tensor to be transformed by the layer. Our equivariant feature of choice, for a $A \times 1$ input tensor x to the layer is

$$\mu^{(L)} = \frac{1}{A} \sum_{i=1}^A x_i^2, \quad (15)$$

to ensure backflow correlations are maintained even when the mean value of these inputs is fixed at 0. These correlations are essential to incorporate many-body correlations into the wavefunction [HM19, KDRF⁺23]. The output of each layer is an $A \times H$ tensor, where H is the number of hidden neurons in the next layer. For a row i of this tensor, $h_i^{(L)}$, we

perform the following computation

$$h_i^{(L)} = \tanh \left(\left(x \mid \mu^{(L)} \right) W^{(L)} + b^{(L)} \right), \quad (16)$$

where $W^{(L)}$ and $b^{(L)}$ are the weights and biases of layer L , respectively, and $(a \mid b)$ represents a matrix constructed from concatenating the columns a and b . The inputs to the layer combined with the equivariant feature tensor are involved in a linear transformation with these weights and biases, and the result goes through a non-linear activation function, in this case the hyperbolic tangent. While the input to the first layer only has one column, subsequent layers have a number of columns H , equal to their number of neurons. The equivariant feature is computed for each of these columns, resulting in H $\mu^{(L)}$ columns too. The result of all of this is that a permutation of the inputs to the layer causes an equivalent permutation of its output rows.

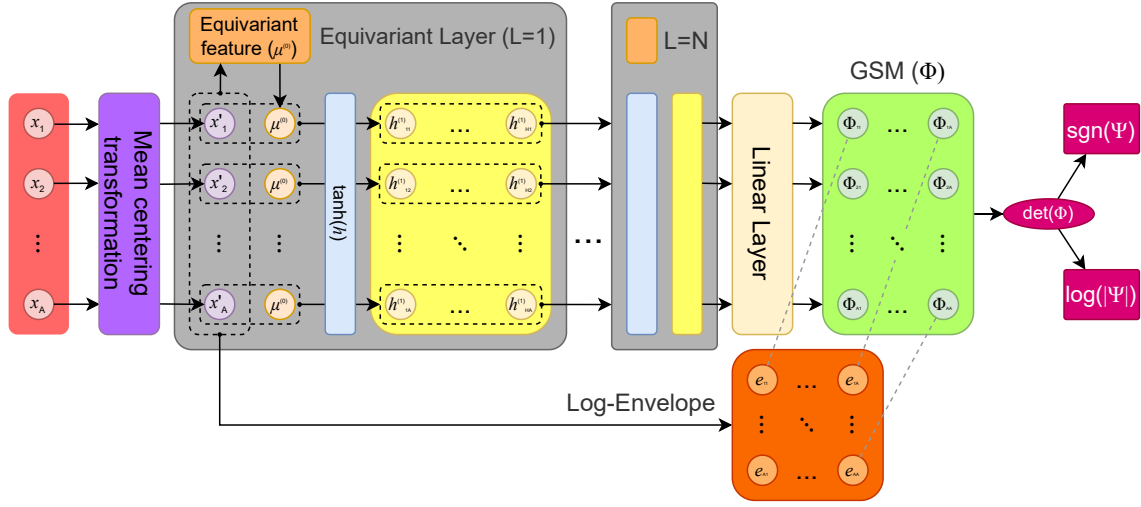


Figure 2: The NQS ansatz architecture used in this work, with N equivariant layers and H neurons per hidden layer. The inputs to the network are the one-dimensional positions of the A particles in the system. First, they undergo a mean centering transformation to model translational invariance. Then, they are processed by a series of equivariant layers (in gray), that ensure the network maintains equivariance through its forward pass. The output of the last equivariant layer is transformed by a linear layer into an $A \times A$ matrix, which is multiplied element-wise by the log-envelope matrix (in orange) to yield a GSM Φ (in green). Finally, the determinant of this matrix is computed via an LU decomposition, yielding the sign and logarithm of the absolute value of the wavefunction Ψ .

The last equivariant layer is followed by a fully connected layer that yields an $A \times A$ matrix M . To impose the asymptotic boundary conditions that state the wavefunction must decay at infinity, this matrix is multiplied element-wise with an envelope e . To improve numerical stability, this envelope is computed in the logarithmic domain:

$$\ln(e_{ij}) = -\left(x_i W_j^{(e)}\right)^2, \quad (17)$$

for a particle i and a natural orbital j , with $W^{(e)}$ the weights corresponding to each orbital. The product of the envelope matrix and the linear layer output results in the GSM Φ .

Finally, the determinant of the GSM, the Generalized Slated Determinant (GSD), is computed via LU decomposition to obtain an antisymmetric wavefunction. Again, for

numerical stability, this computation is performed in the logarithmic domain. The sign and logarithm of the absolute value of the wavefunction are then served as the outputs of the network.

This architecture leads to a total amount of tunable parameters that depends on the amount of particles A , the number of hidden neurons per equivariant layer H (assuming the same number is used for each), and the number of equivariant layers N ,

$$|\{\theta\}| = N(A + 2H + H^2) + HA. \quad (18)$$

It is important to keep the scale of the parameter set in mind as we work with these ansätze, as it plays an important role in the optimization procedure. The networks we use to evaluate this method have $N = 2$ layers with $H = 64$ neurons each. For $A = 6$ particles, this results in a total of 8844 parameters. In terms of space complexity (which determines memory usage), the network scales quadratically both with respect to the amount of particles in the system due to the GSM matrix, and the number of hidden neurons per layer for the connections between them:

$$\mathcal{O}(A^2 + NH^2). \quad (19)$$

3.2 Loss function

In this variational scheme, the NQS acts as the trial wavefunction. To learn the ground state of a target system via the Rayleigh-Ritz method, we aim to minimize the energy expectation of its Hamiltonian with respect to this trial wavefunction. As such, we set the loss function to be the energy expectation, which involves the Metropolis-Hasting sampling explained in Section 2.4,

$$\mathcal{L}_\theta = \frac{\langle \Psi_\theta | \hat{H} | \Psi_\theta \rangle}{\langle \Psi_\theta | \Psi_\theta \rangle} = \mathbb{E}_{X \sim |\Psi_\theta|^2} [\Psi_\theta(X)^{-1} \hat{H} \Psi_\theta(X)], \quad (20)$$

this expression is computed in the logarithmic domain from the output of the network

$$\begin{aligned} \Psi_\theta(X)^{-1} \hat{H} \Psi_\theta(X) = E_L(X) = & \frac{1}{2} \sum_{i=1}^A \left[x_i^2 - \frac{\partial^2 \ln |\Psi_\theta|}{\partial x_i^2} \Big|_X - \left(\frac{\partial \ln |\Psi_\theta|}{\partial x_i} \right)^2 \Big|_X \right] \\ & + \frac{V_0}{\sqrt{2\pi}\sigma} \sum_{i < j} e^{-\frac{(x_i - x_j)^2}{2\sigma^2}}. \end{aligned} \quad (21)$$

$E_L(X)$ are the local energies given a set of samples X . To improve numerical stability, we apply ℓ_1 norm clipping [PSMF20] to these values. This procedure obtains a median energy $\langle E_L \rangle$ with an associated deviation σ_{ℓ_1} . We restrict the energies to lie within the interval $\langle E_L \rangle \pm c_{\ell_1} \sigma_{\ell_1}$, with c_{ℓ_1} a clipping factor that controls how lenient the processing of the values is (in this work, $c_{\ell_1} = 5$). Values outside this range are clipped to be equal to the nearest allowed value.

Once an acceptable set of local energies has been computed, the gradient of this expectation with respect to the parameters comes from

$$\nabla_\theta E(\theta) = 2 \mathbb{E}_{X \sim |\Psi_\theta|^2} \left[(E_L(X) - E(\theta)) \nabla_\theta \ln |\Psi_\theta(X)| \right], \quad (22)$$

which in practice can be computed by applying automatic differentiation (AD) [PGC⁺17] to the auxiliary loss

$$\mathcal{L}_\theta^{\text{aux}} = 2\mathbb{E}_{X \sim \perp(|\Psi_\theta|^2)} \left[\perp (E_L(X) - \mathbb{E}_{X \sim |\Psi_\theta|^2} [E_L(X)]) \times \ln |\Psi_\theta(X)| \right]. \quad (23)$$

Here, $\perp$ indicates the detach operation, which acts as an identity function over its input values while setting their derivative to 0 when accounted for in AD ($\frac{\partial \perp(x)}{\partial y} = 0 \ \forall x, y$). The derivative of this auxiliary function with respect to θ is the same as that of the energy expectation, and it is numerically more stable to compute the gradient this way as it does not require higher order derivatives.

3.3 The RAdamW optimizer

The Rectified Adaptive Moment estimation with decoupled Weight decay optimization algorithm (RAdamW) is a variant of the Adam optimizer [KB14], originally published in 2014. Adam is a first-order gradient-based optimization algorithm for stochastic objective functions. It computes parameter-wise adaptive learning rates from estimates of first and second moments of the gradients. It has become very popular among researchers for its efficiency, scalability, and simplicity of use in many settings.

A common problem with algorithms based on gradient descent is determining the size of the steps to take when advancing in the direction indicated by the gradient. The hyperparameter that controls how coarse the jumps in model parameters are is known as the learning rate. While a small learning rate may delay convergence by a large number of epochs, setting its value to something too large can cause the training to become unstable and incapable of reaching a global minimum [PG17]. To address this concern, Adam adapts the learning rates for different parameters independently based on the moments of the gradients. The first moment at iteration t , $\hat{m}_t$, corresponds to a weighted average of the gradient g_t

$$\hat{m}_t = \frac{\beta_1 m_{t-1} + (1 - \beta_1) g_t}{1 - \beta_1^t}, \quad (24)$$

while the second moment $\hat{v}_t$ corresponds to its variance

$$\hat{v}_t = \frac{\beta_2 v_{t-1} + (1 - \beta_2) g_t^2}{1 - \beta_2^t}. \quad (25)$$

The hyperparameters β_1 and β_2 control how smoothed the moment updates are, and are generally set to values close to 1. The denominators in these expressions correct against bias when t is small and β is initialized with small values. These moment estimates are then used to update the parameters of the network with a global learning rate α :

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}. \quad (26)$$

A technique often used in tandem with optimizers such as Adam is ℓ_2 regularization, which attempts to increase generalization and punish overfitting by adding a term to the loss accounting for the ℓ_2 norm of the model weights. In 2017, a modification of the Adam algorithm was proposed, aimed at replacing the mechanism for regularization to reduce overfitting tendencies. Known as AdamW [LH⁺17], it decoupled the weight decay from the optimization steps taken with respect to the loss function. This change allows for better generalization performance. Another improvement to Adam was presented in 2019,

RAdam [LJH⁺19]. It set out to rectify inconsistencies observed at the early optimization stages by warming up the optimizer with a lower learning rate and no momentum term. According to their findings, these adjustments decrease variance between runs, improve training stability, and achieve convergence in fewer epochs. The implementation used in this work is the one offered as part of PyTorch within the `RAdam` class. We configure RAdam here to use decoupled weight decay, combining the two variants to capitalize on both their improved generalization and stability. By introducing this optimizer into NQS training, we aim to address the inconsistencies caused by the stochastic VMC approach.

3.4 Transfer learning

In this project, the training of a single NQS ansatz is split in three sequential phases:

1. **Pre-training:** We first train the many-body wavefunction to follow that of the non-interacting system. This is achieved by training the GSM on Harmonic Oscillator eigenstates in a supervised learning setting.
2. **Trapped system:** We perform an energy minimization over the Hamiltonian with a harmonic trap, Eq. 2.
3. **Removal of the harmonic trap:** We continue energy minimization, but the harmonic trap is disabled as in Eq. 3, and a mean-centering transformation is enabled. This centers the system at the origin.

After each phase, the network parameters are preserved and used as the initial state for the subsequent optimization phase. The idea behind this process is to guide the shaping of the wavefunction toward the final result by obtaining states that are physical and somewhat similar to it. Starting an optimization phase with parameters that are closer to the minimum allows us to achieve convergence in a lower amount of epochs. In the case of potentially non-convex loss functions, it also makes it likelier that each phase of training begins within the valley the global minimum lies in. This is a form of transfer learning, which has been shown to offer significant benefits in speeding up the training of NQSs [Azz23].

A notable difference for the pre-training phase is that, since we work with a non-interacting system, it is simple to obtain the analytical solutions for each particle and use them as part of a supervised learning scheme. Given the Harmonic Oscillator eigenstates

$$\varphi_n(x) = \mathcal{N}_n e^{-\frac{x^2}{2}} H_n(x) \quad (27)$$

(with H_n the n^{th} Hermite polynomial and $\mathcal{N}_n = (n!2^n\sqrt{\pi})^{-\frac{1}{2}}$, a normalization constant) [RS80], we can train the elements of the GSM ϕ_{ij} directly against them by setting the pre-training loss to

$$\mathcal{L}_\theta^{\text{pre}} = \int d\{x\} |\Psi_\theta(\{x\})|^2 \times \sum_{ij} (\phi_{ij} - \varphi_i(x_j))^2 = \mathbb{E}_{X \sim |\Psi_\theta|^2} \left[\sum_{ij} (\phi_{ij} - \varphi_i(x_j))^2 \right], \quad (28)$$

implemented by taking the Monte Carlo estimation of the integral. This loss amounts to the sum of the squared distances of the GSM elements against the Harmonic Oscillator eigenstates corresponding to the matching orbitals. It is easy to see that minimizing these differences causes the NQS to learn the non-interacting many-body wavefunction.

Additionally, we use transfer learning in the performed experiments to approach training over multiple systems with the same number of particles but different interaction parameters [Azz23]. At the end of the final training phase, the network can be repurposed by swapping the interaction term and starting another optimization pass directly from the final phase. When attempting to solve multiple similar systems in bulk this technique speeds up training, potentially saving considerable amounts of time and energy. For the case of self-bound systems, it is important to note that the state must remain bound for transfers to be effective. This means that in practice, when trying to transfer learning between cases with different interactions, stronger interaction potentials and longer interaction distances should run first. In case a certain set of interaction parameters is not capable of binding the particles, ordering transfers this way ensures they are efficient. This way, other cases that would be able to bind do not suffer from a transfer that would cause them to be stuck at a local minimum.

3.5 Mean-centering transformation

In self-bound systems such as the one studied in this work, the lack of an external potential causes no position in space to be preferable over any other for the center of mass of the system. Thus, an NQS attempting to minimize the energy over such a state would attempt to stretch its wavefunction excessively along space. This can cause numerical issues, as the neural networks struggle to approximate the distribution, the samplers cannot span an infinite hypervolume, and floating point errors may start to become noticeable. This behavior also clashes with the asymptotic boundary conditions of our ansatz architecture, which state that the wavefunction must decay at infinity as enforced by the envelope function.

This poses a problem when attempting to implement NQS-based solutions for self-bound systems. Figure 3 shows the evolution of the energy expectation value, the center of mass, and the average VMC sample position throughout a sample NQS optimization, illustrating the error. We see that past a certain number of epochs the center of mass drifts away from the origin, and the sampler fails to follow the distribution. This leads to massive divergences in the energy as numerical stability is lost. This comes as a result of naive NQs not managing to model these kinds of states. When transferring learning from a trapped state, the wavefunction becomes increasingly elongated along space, with its center of mass drifting off the origin until optimization runs into numerical instabilities, and the samplers fail to track the density distribution.

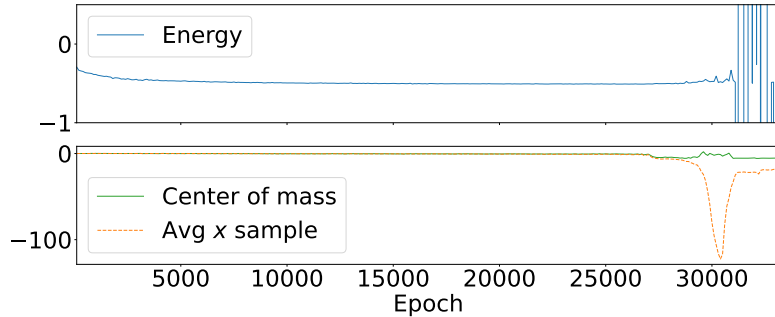


Figure 3: Top panel: evolution of the energy expectation as a function of training epochs. Bottom panel: evolution of the NQS center of mass (green), and the average walker sample (orange, dashed) across training epochs. Optimization runs into numerical instabilities due to the inability to model translational invariance.

Our proposed solution to this problem involves a mean-centering transformation of the input coordinates as part of the NQS ansatz architecture. We look at the transformation in its matrix form S_A , since an efficient implementation applies it via batch matrix multiplication. For a number of particles A , the transformation reads

$$S_A \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_A \end{pmatrix} = \frac{1}{A} \begin{pmatrix} A-1 & -1 & \cdots & -1 \\ -1 & A-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & A-1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_A \end{pmatrix} = \begin{pmatrix} x_1 - \sum_i \frac{x_i}{A} \\ x_2 - \sum_i \frac{x_i}{A} \\ \vdots \\ x_A - \sum_i \frac{x_i}{A} \end{pmatrix}, \quad (29)$$

which results in subtracting the mean many-body position from each of the particle coordinates. This effectively fixes the center of mass at the origin. It is important to note that this transformation is equivariant with respect to particle permutations, which ensures the antisymmetry properties of the network are preserved. Since the mean many-body position is set at zero, it cannot be chosen as the equivariant feature and we use Eq. 15 instead. This preserves backflow correlations, preventing a degradation of the performance of the ansatz [KDRF⁺23]. Working under this transformation is equivalent to using the coordinate system given by

$$R_A \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_A \end{pmatrix} = \begin{pmatrix} \sum_i \frac{x_i}{\sqrt{A}} \\ x_2 - \frac{x_1}{\sqrt{A+1}} - \sum_i \frac{x_i}{\sqrt{A+A}} \\ \vdots \\ x_A - \frac{x_1}{\sqrt{A+1}} - \sum_i \frac{x_i}{\sqrt{A+A}} \end{pmatrix} = \begin{pmatrix} R \\ r_1 \\ \vdots \\ r_{A-1} \end{pmatrix}, \quad (30)$$

composed of a center of mass coordinate R (which we choose to be zero), and $A-1$ relative coordinates $\{r\}$. This equivalence is discussed in Appendix A. Figure 4 shows two approximate wavefunctions generated by our NQSs plotted against these relative coordinates. On the left panel, the two-body case shows an exponential decay on each side, separated by a node crossing the origin due to antisymmetry. It is a close match to our benchmark, produced from a similar solution as the one discussed in Section 2.1, but for a Gaussian potential (in orange, dashed). The right panel shows the equivalent graph for a three particle system, in which three symmetry axis cross the origin, each corresponding to the relative positions of a pair of particles being zero.

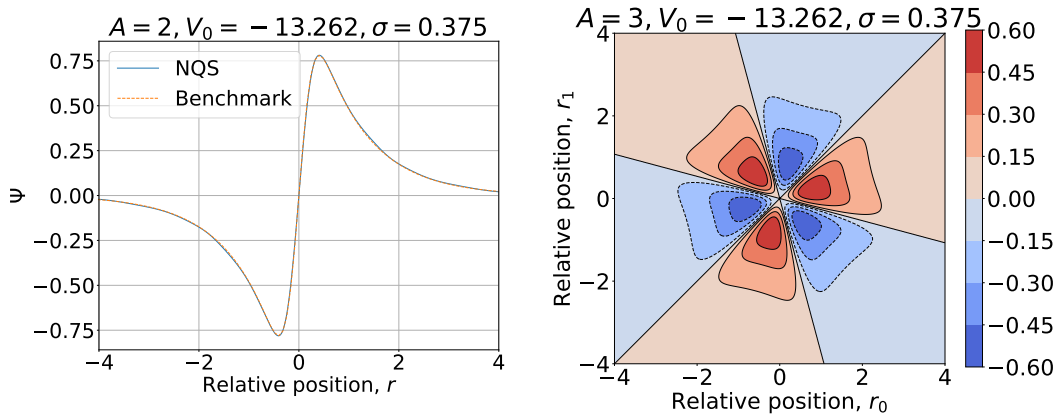


Figure 4: Left panel: Trained NQS wavefunction (in blue) and analytical benchmark (in orange, dashed) for a two-particle system with respect to the relative coordinate. The NQS solution has been normalized so that $\int \Psi_\theta(x)^2 dx = 1$. Right panel: NQS wavefunction for a three-body system with respect to the two relative coordinates. The three axis of symmetry (with density zero) correspond to positions at which pairs of particles occupy the same position.

This solution overcomes the clash with the ansatz asymptotic boundary conditions, since the transformation is applied before the log-envelope matrix is computed. Regardless, the envelope function remains relevant with respect to the local coordinates, ensuring that the wavefunction decays when interparticle distances tend to infinity. An important aspect to keep in mind is that this projection is transparent to every other component of the implementation. The coordinate system is kept the same, and even though the dimensionality of the coordinate space we work with becomes $A - 1$, the length of the NQS input vector is still A . This allows us to enable the transformation on an existing network and transfer learning without any additional changes. The sampler also need not account for the change, as it sees a distribution that extends along space and follows it accordingly. Incorporating the transformation as part of the NQS black box means its interface, with absolute coordinates as the input, remains unchanged.

4 Results

4.1 Energy

For a state to remain self-bound under the conditions studied, the contribution from the attractive interaction between fermions must overcome that of the kinetic energy. Thus, we expect the energies of these states to be negative. Figure 5 shows the ground state energies computed for three different interactions with the same scattering length. The horizontal axis represents a growing number of fermions, up to $A = 9$. For visual clarity, sample points are joined with dashed lines.

The energy appears to decrease linearly as the number of particles increases. Table 1 shows this energy progression. The difference between each measurement stays about constant for $A > 4$. For lower particle numbers, the scaling deviates slightly from this linear relationship, and so this assumption could be inaccurate to assess the jump from smaller values like $A = 2$ to $A = 3$. We also observe that the resulting energies are the same for the three combinations of $\{V_0, \sigma\}$ leading to the same value of a . This follows the prediction of universality from Section 2.1.

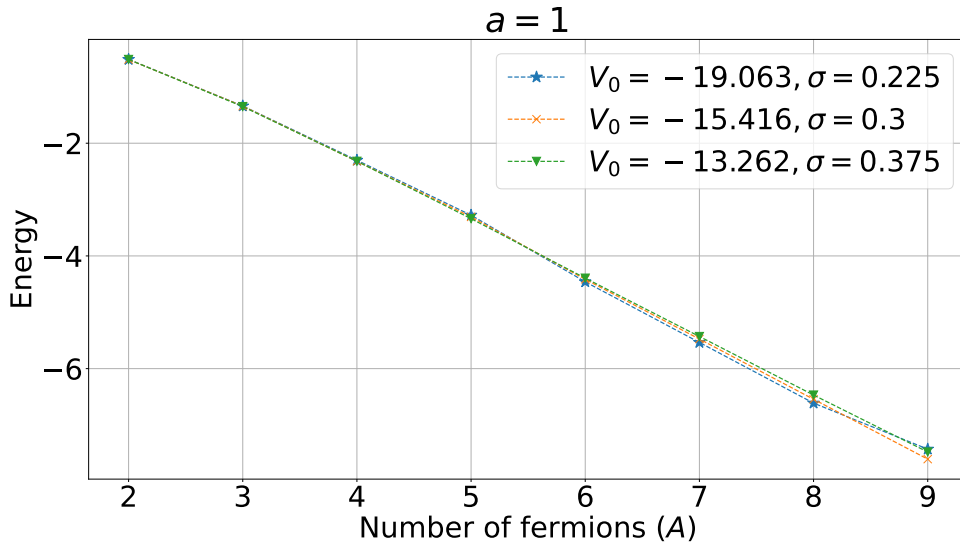


Figure 5: Ground state energies for three different sets of interaction parameters resulting in the same scattering length ($a = 1$). The three give rise to the same set of energies.

A	3	4	5	6	7	8	9
$E_{A-1} - E_A$	0.8365	0.9705	1.0187	1.0551	1.0356	1.0392	1.0111

Table 1: Energy differences as the number of fermions A increases for a fixed scattering length $a = 1$.

We now look at how the energy depends on the scattering length. Figure 6 plots the ground state energies with respect to the amount of fermions in the system, now for four different scattering lengths ranging from 0.5 to 2 in increments of 0.5. Again, samples are joined with dashed lines to illustrate the roughly linear progression.

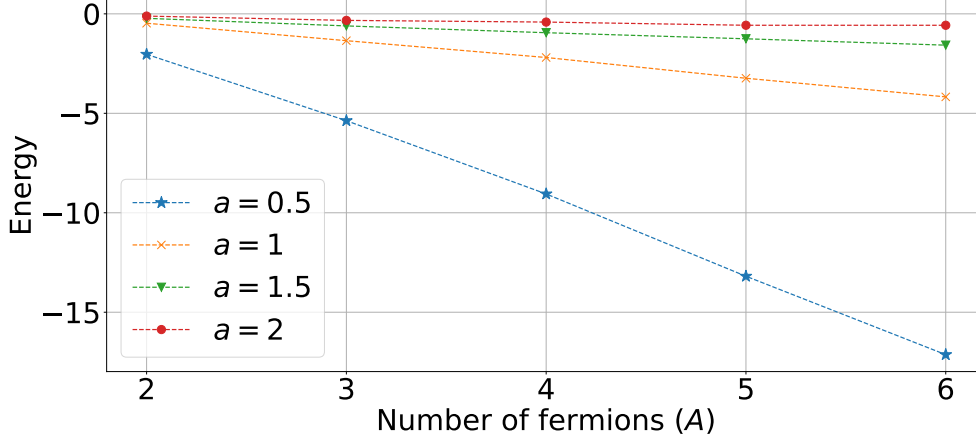


Figure 6: Ground state energies for four different scattering lengths, and an increasing number of particles. The relationship between energy and number of fermions is linear, with lower scattering lengths resulting in a steeper slope.

The slope of each set of energies changes as does a , still showing a close to linear progression. Lower scattering lengths correspond to steeper energy progressions, with a lower energy value per particle. The change in incline is more pronounced for smaller values of a .

To better understand the relationship between these ground states energies and the scattering length, we measure the energies our method predicts for a finer mesh of values of a , between 0.2 and 2.0. This range is of most interest to study, as energies and potentials become overly strong as a approaches 0 and scattering lengths above 2 can hardly lead to bound states. The results are plotted in Figure 7. In blue are the measured estimates for our ansatz when $A = 3$. The samples in purple are similar measurements for $A = 2$. The dotted green line corresponds to the prediction from Eq. 11. The dashed orange line is proportional to this prediction. Its constant coefficient comes from the relationship between the two and three particle energies for a small scattering length of $a = 0.5$:

$$\frac{1}{2} \frac{E_{A=3}}{E_{A=2}} = \frac{1}{2} \frac{5.3702}{2.0335} \approx \frac{4}{3}. \quad (31)$$

We show that the measured energies closely match the inverse quadratic relationship we had predicted. The exact values differ slightly from these functions, both as a result of the error originating from the NQS approximation and due to the difference between the square and Gaussian wells. From these experiments, we draw two relationships that can help us understand the ground state energy of these systems. A linear relationship to the

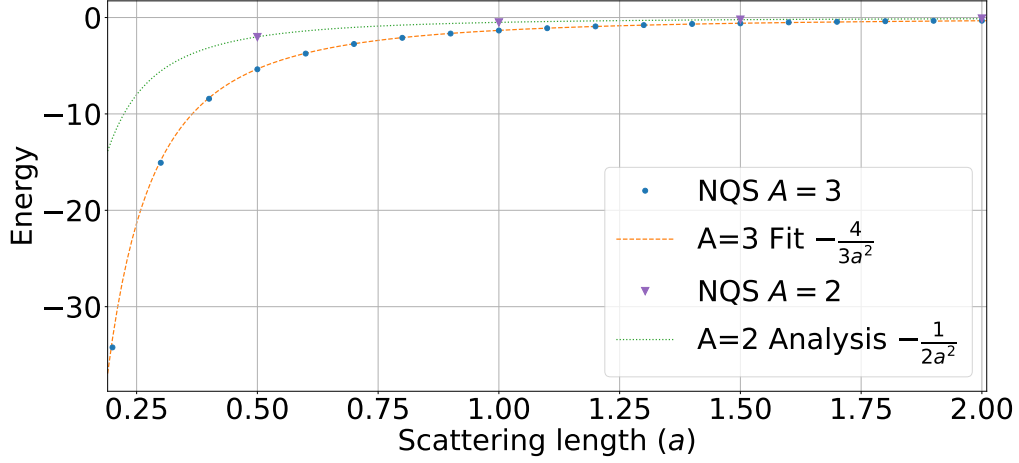


Figure 7: NQS estimated ground state energies for systems with $A = 2$ (purple triangles) and $A = 3$ (blue circles), plotted against a varying a . The prediction from the $A = 2$ analytical approximation is pictured (green, dotted) along a proportional fit for $A = 3$ (orange, dashed).

amount of fermions in the system (which appears to become more accurate as the number of particles grows), and an inverse quadratic relationship to the scattering length:

$$E_{\text{gs}} \propto \frac{A}{a^2}. \quad (32)$$

4.2 Density profiles

We switch our focus to the density profiles $n(x)$ of the systems with $A = 2$ to 6 particles. Figure 8 shows the results for three different scattering lengths, ranging from 0.5 to 1.5 in increases of 0.5 from the top to the bottom row. For a small σ , the wavefunctions and thus the density profiles are nearly identical for the same a and different combinations of $\{V_0, \sigma\}$. The number of particles increases from left to right. The vertical axis corresponds to the

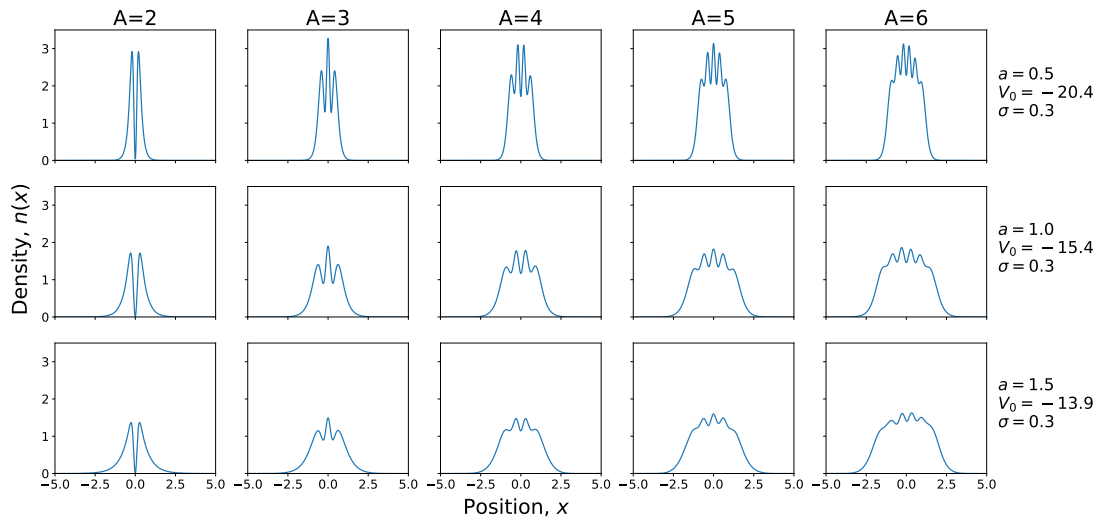


Figure 8: Density profiles $n(x)$ as a function of absolute position x , for different combinations of a, A . From left to right the number of fermions increases from 2 to 6. From top to bottom the scattering length increases from 0.5 to 1.5.

density, while the horizontal axis is the particle position. The combinations of interaction parameters used to achieve this are shown on the right.

Starting with the two-body case, we see two density peaks symmetric with respect to the origin, with a valley of zero density separating them. With our center of mass fixed at 0, the pair of particles must mirror each other around the origin. The valley at the origin comes from the impossibility of observing both fermions at the same location. We also see an exponential decay on each side, with a dictating the scale of the decay. This matches the expectations set by our analysis in Section 2.1. When the number of particles increases, an equal number of A peaks appear in the density, on top of an overall Gaussian-like behavior. This fermionic behavior is more pronounced when the scattering length is smaller, and the density smoothes out as a grows. While the position of the peaks barely changes, they are more defined for lower scattering lengths. Lower scattering lengths, which translate to stronger interactions, also lead to narrower distributions. These results indicate that the system is bound to a smaller region of space the lower the amount of fermions and the stronger the interactions between them.

4.3 One-body density matrices

To get a more complete picture of nonlocal structures, we look at the one-body density matrix. To compute $\rho(x_1, x'_1)$, we use the ghost-particle method as in [McM65]. Here, we are computing the center-of-mass free density matrix [Yab86]. Usually, the coordinates of the density matrix correspond to some spatially fixed origin. In this case, the system shows no preference for a well defined center of mass and in practice translational symmetry is broken when the bound state forms. Thus, measuring with respect to absolute coordinates would show a mixture of states with different centers of mass, equivalent to the data being convolved with a uniform distribution in space. To compute this matrix stochastically, we adjust the sampling by applying S_A onto our generated configurations. This procedure does not interfere with the scale of the measurements, as $|R_A| = 1$ (similar to the Jacobi coordinates used by Yabana).

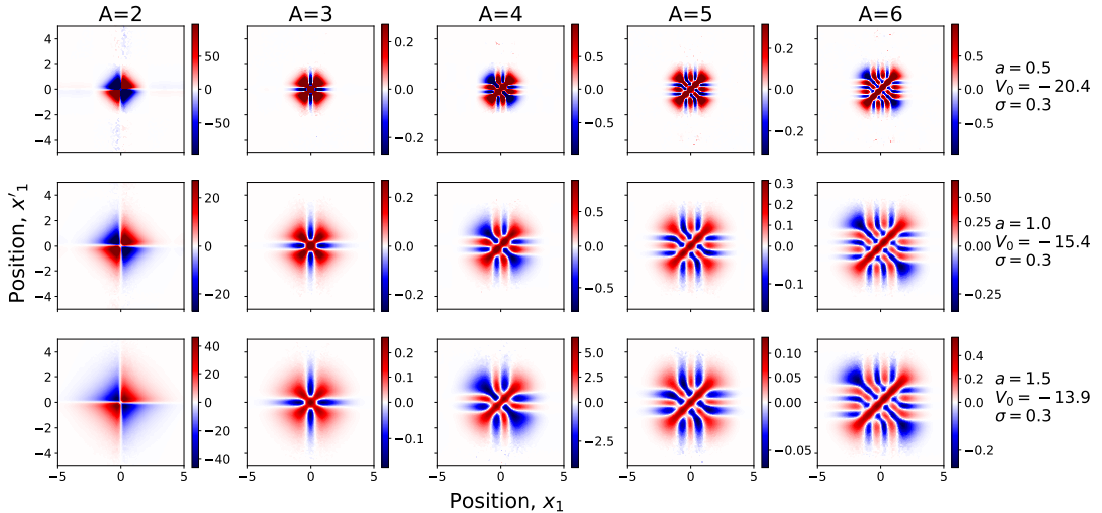


Figure 9: Density plots for the one body density matrices of systems with different combinations of a, A . From left to right the number of fermions increases from 2 to 6. From top to bottom the scattering length increases from 0.5 to 1.5.

The results are shown in Figure 9, again with the number of particles increasing from left to right and the three scattering lengths being laid out from top to bottom. The horizontal and vertical axis correspond to the particle position x_1 and ghost particle position x'_1 , respectively. We use a diverging color scale, normalized to fit each panel, to better illustrate the oscillating structures that emerge from changes in sign. The stochastically computed data has been treated with ℓ_1 norm clipping [PSMF20] as to aid visualization.

We can see how a narrow band of positive values passes along the diagonal, where $\rho(x_1, x'_1 = x_1) \equiv n(x) > 0$. Once again, it is apparent that higher scattering lengths lead to more disperse systems. This matches the intuition that as $a \rightarrow \infty$, the distribution becomes uniform across space. Here, we also observe oscillations parallel to the antidiagonal that increase in number with the amount of fermions in the system. They reflect the nodal structure of the single-particle states. These ripples become more packed together and increase in magnitude the stronger the interaction.

4.4 Pair correlation functions

Finally, we look at the pair correlation function $g(x_1, x_2)$. Our results are presented in Figure 10 for systems, again, between 2 and 6 particles increasing from left to right. The same three scattering lengths are used, increasing from top to bottom. The horizontal and vertical axis correspond to the pair of particle positions, x_1 and x_2 . In this case the color scale is sequential, and it has been normalized to the maximum of each panel.

In the two particle case, there is a one-to-one correlation between the positions of the two bodies. With the center of mass fixed at the origin, it is expected that when a particle is found at $x_1 = y$, its identical pair is found at $x_2 = -y$. This leads to a one-dimensional sliver of density along the antidiagonal. As the number of particles increases, it is possible for different configurations of particles to balance the center of mass at the origin, and so the densities spread out along the diagonal. This balancing causes two additional off-diagonal bands of zero density in the three-body cases, where x_3 would be forced to the same position as either x_1 or x_2 . Density matrices of order $A - 1$ would show similar bands

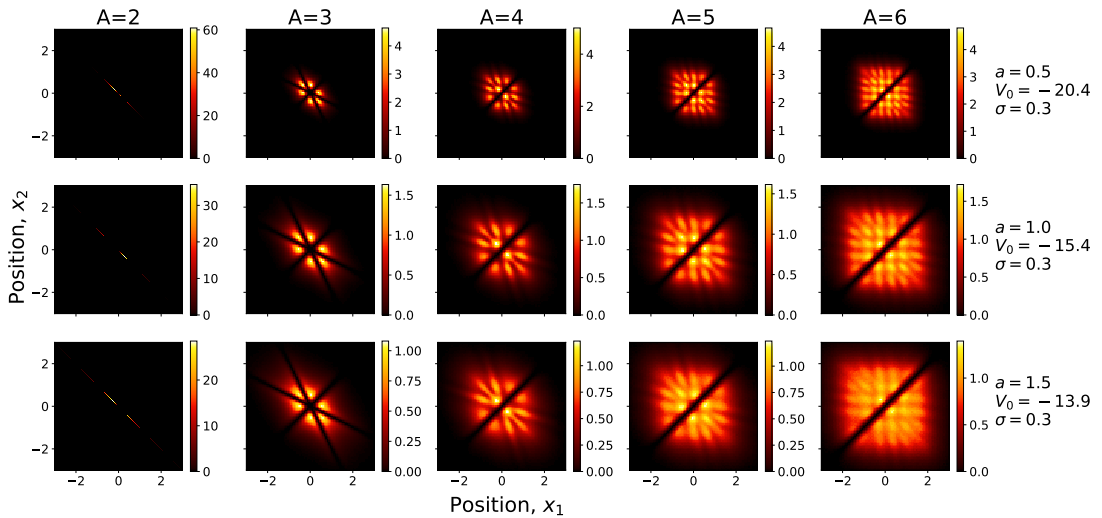


Figure 10: Density plots for the pair correlation functions of systems with different combinations of a , A . From left to right the number of fermions increases from 2 to 6. From top to bottom the scattering length increases from 0.5 to 1.5.

for systems of size A . The higher the scattering length, the wider the distribution becomes. Lower scattering lengths show more pronounced density peaks. As expected for fermionic systems, we see a band of zero density along the diagonal that emerges from antisymmetry, indicating that two particles may not occupy the same position.

4.5 The importance of transfer learning

In Section 3.4, we talked about how transfer learning is used to guide training by approaching the problem in different phases, each approximating the wavefunction with increasing accuracy. Here, we briefly compare the chosen approach with one in which optimization starts directly at the removal of the harmonic trap with random network weights.

The convergence plot for both procedures is pictured in Figure 11. The vertical axis represents the loss function $\mathcal{L}$, while the horizontal axis corresponds to training epochs. The total amount of epochs is 10000 for both approaches, but this amount is distributed differently between phases. The blue line represents the single phase run, whose loss always corresponds to the energy minimization of the Hamiltonian in Eq. 3. The orange line represents transfer learning. Discontinuities are present at the points in which transfers happen, 1000 epochs (in green) and 2000 epochs (in purple). This is a result of the loss function switching from Eq. 28, to minimizing the energy over the Hamiltonian in Eq. 2, to finally matching single phase learning and minimizing the expectation of Eq. 3.

It is possible to see how the transfer learning convergence takes over the single phase scheme, achieving a better energy estimate using the same amount of epochs in a more efficient manner. Our results show that transfer learning expedites the optimization process, leading to faster and more consistent convergence. By starting each training phase with an approximate solution rather than random parameters, we can be more certain that we are minimizing toward the global minimum valley. These improvements translate to increases in both time and energy efficiency.

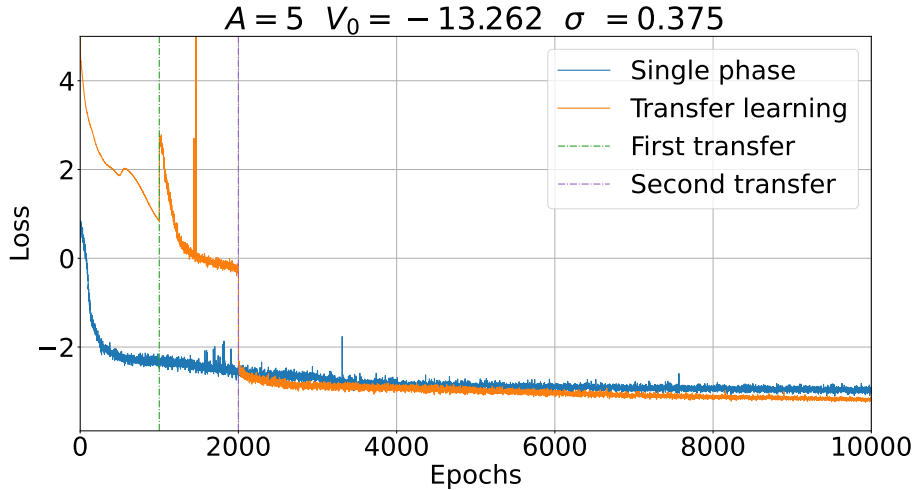


Figure 11: Convergence plot comparing two runs with the same system parameters. One of them is performed in a single phase starting with random weights directly without a harmonic trap (in blue), while the other makes use of transfer learning (in orange). Two transfers are performed: one at 1000 epochs (in green, from orbital fitting), and another at 2000 epochs (in purple, from a trapped system). The total number of epochs is 10000 in both cases. Transfer learning outperforms single phase training.

5 Conclusions and outlook

Throughout this thesis, we have analyzed the ground state properties of one-dimensional systems composed of fully-polarized fermions that are bound by attractive Gaussian interactions. We approach the problem by making use of the Neural Quantum States technique. This method allows for the approximation of the ground state wavefunction of a system as part of a Variational Monte Carlo scheme. The NQS representation makes our solution scale in a robust way with respect to the size of the system. The exponential complexity originating from the growth of the Hilbert space is avoided by encoding information into Neural Network parameters. The flexibility of Neural Networks as a universal approximator enables us to use them as an ansatz without extensive prior knowledge about the structure of the wavefunctions we are searching for.

We adapt an existing ansatz architecture, capable of modeling fermionic behavior, to include a coordinate transformation that fixes the center of mass of the system at the origin. This procedure allows us to analyze the system ignoring spurious contributions from center of mass motion. Without it, the ansatz would not be able to model self-bound states, which break the translational symmetry of the Hamiltonian by becoming bound within a limited region of space around an arbitrary center of mass. Minimizing the energy expectation using this transformation is equivalent to working under a relative coordinate system. We also introduce to this problem a variant of the Adam optimizer, selected to improve training stability and generalization. The training process is performed in consecutive phases that exploit transfer learning to achieve more efficient convergence. Overall, our solution greatly improves numerical stability, overcoming the difficulties characteristic of the problem. This implementation has been open-sourced as part of a public repository [MGK24].

Our results closely match the analytical benchmark we set for the two-body case, producing nearly identical wavefunctions with respect to the relative particle coordinate that lead to the same energies, density profiles, and density matrices. We also compute results for higher numbers of particles, for which there are no previous predictions provided by other methods. The NQS measurements show an inverse quadratic dependence of the ground state energy on the scattering length a . For $a \gg \sigma$, a fully characterizes the system, behaving like a “1D fermionic halo system”. We also observe a roughly linear trend in the scaling of the energy with respect to the amount of fermions in the systems up to $A = 9$. This linear relationship becomes increasingly accurate as the number of particles grows. We stochastically compute the density profiles, one-body density matrices, and pair correlation functions for systems up to 6 particles. This allows us to observe the nodal structure of the particle states, with as many fermion-like density peaks as particles in the system. Stronger interactions bind the system to smaller regions of space, and lead to more pronounced density peaks. In the two particle case the pair of fermions is perfectly correlated, balancing to keep the center of mass at the origin. This one-to-one correlation becomes less pronounced as the number of particles increases.

Research following this line of work in the near future could provide further insight into this system, as well as more complex variations. Having access to an approximation of the full wavefunction of the system, more interesting properties could be extracted. Looking at the problem from the perspective of momentum space could also lead to interesting findings. Adapting the approach to account for spin would open an array of possibilities, including using this methodology to solve three-dimensional systems. Finally, tackling the problem via other methods could help verify our results with a higher degree of certainty.

Bibliography

- [Ama67] S. Amari. A theory of adaptive pattern classifiers. *IEEE Trans. Electron. Comp.*, EC-16(3):299–307, 1967.
- [Azz23] A. Azzam. Transfer learning for many-body systems. Master’s thesis, University of Barcelona, 2023.
- [BDZ08] I. Bloch, J. Dalibard, and W. Zwerger. Many-body physics with ultracold gases. *Rev. Mod. Phys.*, 80:885–964, 2008.
- [BERW98] T. Busch, BG. Englert, K. Rzazewski, and M. Wilkens. Two cold atoms in a harmonic trap. *Found. Phys.*, 28(4):549–559, 1998.
- [CCC⁺19] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, L. Tishby, N. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, 2019.
- [CCRN18] K. Choo, G. Carleo, N. Regnault, and T. Neupert. Symmetries and many-body excitations with neural-network quantum states. *Phys. Rev. Lett.*, 121:167204, 2018.
- [CT17] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.
- [Has70] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [HC19] M. J. Hartmann and G. Carleo. Neural-network approach to dissipative quantum many-body dynamics. *Phys. Rev. Lett.*, 122:250502, 2019.
- [HM19] M. Holzmann and S. Moroni. Orbital-dependent backflow wave functions for real-space quantum monte carlo. *Phys. Rev. B*, 99:085121, 2019.
- [Hor91] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- [JQ22] G. L. Jones and Q. Qin. Markov chain monte carlo in practice. *Ann. Rev. Stat. Appl.*, 9:557–578, 2022.
- [KB14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [KDRF⁺23] J. W. T. Keeble, M. Drissi, A. Rojo-Francàs, B. Juliá-Díaz, and A. Rios. Machine learning one-dimensional spinless trapped fermionic systems with neural-network quantum states. *Phys. Rev. A*, 108:063320, 2023.
- [Kee22] J. W. T. Keeble. Neural network solutions to the fermionic Schrödinger equation. Doctoral thesis, University of Surrey, 2022.
- [KMH⁺22] D. Koutný, L. Motka, Z. Hradil, J. Řeháček, and L. L. Sánchez-Soto. Neural-network quantum state tomography. *Phys. Rev. A*, 106:012409, 2022.
- [KPSG⁺21] D. Kochkov, T. Pfaff, A. Sanchez-Gonzalez, P. Battaglia, and B. K. Clark. Learning ground states of quantum hamiltonians with graph networks. *arXiv preprint arXiv:2110.06390*, 2021.
- [LH⁺17] I. Loshchilov, F. Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5, 2017.
- [Lin70] S. Linnainmaa. The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors. Master’s thesis, University of Helsinki, 1970.
- [LJH⁺19] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

- [LVdWAB24] H. Lange, A. Van de Walle, A. Abedinnia, and A. Bohrdt. From architectures to applications: A review of neural quantum states. *arXiv preprint arXiv:2402.09402*, 2024.
- [McM65] W. L. McMillan. Ground state of liquid He^4 . *Phys. Rev.*, 138:A442–A451, 1965.
- [MGK24] J. Mosteiro-García and J. W. T. Keeble. Self bound spinless fermions. *GitHub repository* github.com/5uso/SelfBoundSpinlessFermions, 2024. [Accessed: 07-07-2024].
- [MRR⁺53] N. Metropolis, A. W. Rosenbluth, M. N Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.
- [NYN21] Y. Nomura, N. Yoshioka, and F. Nori. Purifying deep boltzmann machines for thermal quantum states. *Phys. Rev. Lett.*, 127:060601, 2021.
- [PG17] J. Patterson and A. Gibson. *Deep Learning: A Practitioner’s Approach*. O’Reilly Media, Inc., 1st edition, 2017.
- [PGC⁺17] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. *NIPS 2017 Workshop Autodiff Submission*, 2017.
- [PSMF20] D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Phys. Rev. Res.*, 2:033429, 2020.
- [RFPJD20] A. Rojo-Francàs, A. Polls, and B. Juliá-Díaz. Static and dynamic properties of a few spin 1/2 interacting fermions trapped in a harmonic potential. *Mathematics*, 8(7), 2020.
- [Rit09] W. Ritz. Über eine neue methode zur lösung gewisser variationsprobleme der mathematischen physik. *Crelles J.*, 135:1–61, 1909.
- [Ros58] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386, 1958.
- [RS80] P. Ring and P. Schuck. *The Nuclear Many-Body Problem*. Springer Science & Business Media, 1980.
- [RSKR24] J. Rozalén-Sarmiento, J. W. T. Keeble, and A. Rios. Machine learning the deuteron: new architectures and uncertainty quantification. *Eur. Phys. J. Plus*, 139(2):189, 2024.
- [SGM19] T. Sowiński and M. Á. García-March. One-dimensional mixtures of several ultracold atoms: a review. *Rep. Prog. Phys.*, 82(10):104401, 2019.
- [SH20] M. Schmitt and M. Heyl. Quantum many-body dynamics in two dimensions with artificial neural networks. *Phys. Rev. Lett.*, 125:100503, 2020.
- [SV98] Y. Suzuki and K. Varga. *Stochastic Variational Approach to Quantum-Mechanical Few-Body Problems*. Springer Science & Business Media, 1998.
- [Vid04] G. Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.*, 93:040502, 2004.
- [WF08] A. Weiße and H. Fehske. *Computational Many-Particle Physics*. Springer Berlin Heidelberg, 2008.
- [Yab86] K. Yabana. On the center-of-mass free density matrix. *Prog. Theor. Phys.*, 76(2):414–432, 1986.
- [Še86] P. Šeba. Some remarks on the $\delta^{\prime}$ -interaction in one dimension. *Rep. Math. Phys.*, 24(1):111–120, 1986.

A Relative coordinate derivation

Another way of understanding our proposed mean-centering transformation is as a projection to a hyperplane on the A -dimensional space orthogonal to the axis of absolute motion in which we

1. Switch to a coordinate system with a center of mass coordinate and $A - 1$ relative coordinates via a rotation (R_A).
2. Collapse the center of mass coordinate to the origin (C_A).
3. Switch back to absolute coordinates by undoing the rotation (R_A^{-1}).

$$S_A = R_A^{-1} C_A R_A \quad (33)$$

The rotation that must be applied to switch to relative coordinates is one that aligns the axis of absolute motion so that it becomes part of the natural basis $\{\vec{e}\}$. This axis is parallel to the A -dimensional “all-ones” vector $\vec{1}$, and moving along it changes the center of mass without altering the relative coordinates. Defining $u_A = \frac{\vec{1}}{\sqrt{A}}$, and $v_A = \vec{e}_1 - \frac{\vec{1}}{A}$, the rotation can be written as

$$R_A = \begin{pmatrix} u_A & | & v_A \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} u_A & | & v_A \end{pmatrix}^T + I_A - u_A u_A^T - v_A v_A^T, \quad (34)$$

with $\cos \theta = \frac{1}{\sqrt{A}}$ and I_A the $A \times A$ identity matrix. It can be seen as the sum of a rotation of the plane spanned by $\{\vec{1}, \vec{e}_1\}$, and a projection to its complementary $A - 2$ dimensional subspace. Applying this to an input vector, we obtain a center of mass coordinate R and a set of relative coordinates $\{r\}$

$$\begin{aligned} R_A \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_A \end{pmatrix} &= \frac{1}{\sqrt{A}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ -1 & \sqrt{A} - \frac{1}{\sqrt{A+1}} & \cdots & -\frac{1}{\sqrt{A+1}} \\ & \vdots & \ddots & \vdots \\ -1 & -\frac{1}{\sqrt{A+1}} & \cdots & \sqrt{A} - \frac{1}{\sqrt{A+1}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_A \end{pmatrix} \\ &= \begin{pmatrix} \sum_i \frac{x_i}{\sqrt{A}} \\ x_2 - \frac{x_1}{\sqrt{A+1}} - \sum_i \frac{x_i}{\sqrt{A+1}} \\ \vdots \\ x_A - \frac{x_1}{\sqrt{A+1}} - \sum_i \frac{x_i}{\sqrt{A+1}} \end{pmatrix} = \begin{pmatrix} R \\ r_1 \\ \vdots \\ r_{A-1} \end{pmatrix}. \end{aligned} \quad (35)$$

The matrix that collapses the center of mass coordinate can then simply be constructed as $C_A = I_A - \vec{e}_1 \vec{e}_1^T$, which is a projection to the hyperplane spanned by the relative coordinates, orthogonal to the axis of absolute motion and crossing the origin. This transformation removes a degree of freedom from the system, fixing the center of mass at the origin. The behavior of the system must thus depend on the relative particle positions only, facilitating optimization.

It is possible to rewrite our Hamiltonian in terms of this new coordinate system, and separate it into two components following $\hat{H} = \hat{H}_{\text{CM}} + \hat{H}_r$. Taking the Hamiltonian without

a harmonic trap, the center of mass and relative components are

$$\begin{aligned}\hat{H}_{\text{CM}} &= -\frac{1}{2}\nabla_R^2, \\ \hat{H}_r &= -\frac{1}{2}\sum_{i=1}^A \nabla_{r_i}^2 + \frac{V_0}{\sigma\sqrt{2\pi}} \left(\sum_{1 \leq i < j} e^{-\frac{(r_i - r_j)^2}{2\sigma^2}} + \sum_{1 \leq i} e^{-\frac{\left(r_i + \frac{\sum_j r_j}{\sqrt{n+1}}\right)^2}{2\sigma^2}} \right).\end{aligned}\tag{36}$$

Separating the Hamiltonian in this manner shows that our transformation, which fixes the center of mass at a static position, preserves the ground state properties of the system. For the ground state of this system, the kinetic energy associated to the movement of the center of mass must be zero, and so the energy minimization procedure will naturally nullify that component. This gives us some guarantees about the correctness of the transformation. In fact, for $A = 2$, the resulting coordinate system,

$$R_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} x_1 + x_2 \\ x_2 - x_1 \end{pmatrix} = \begin{pmatrix} R \\ r \end{pmatrix},\tag{37}$$

produces a center of mass coordinate R and relative coordinate r equivalent to the ones used in Busch et al. [BERW98].