

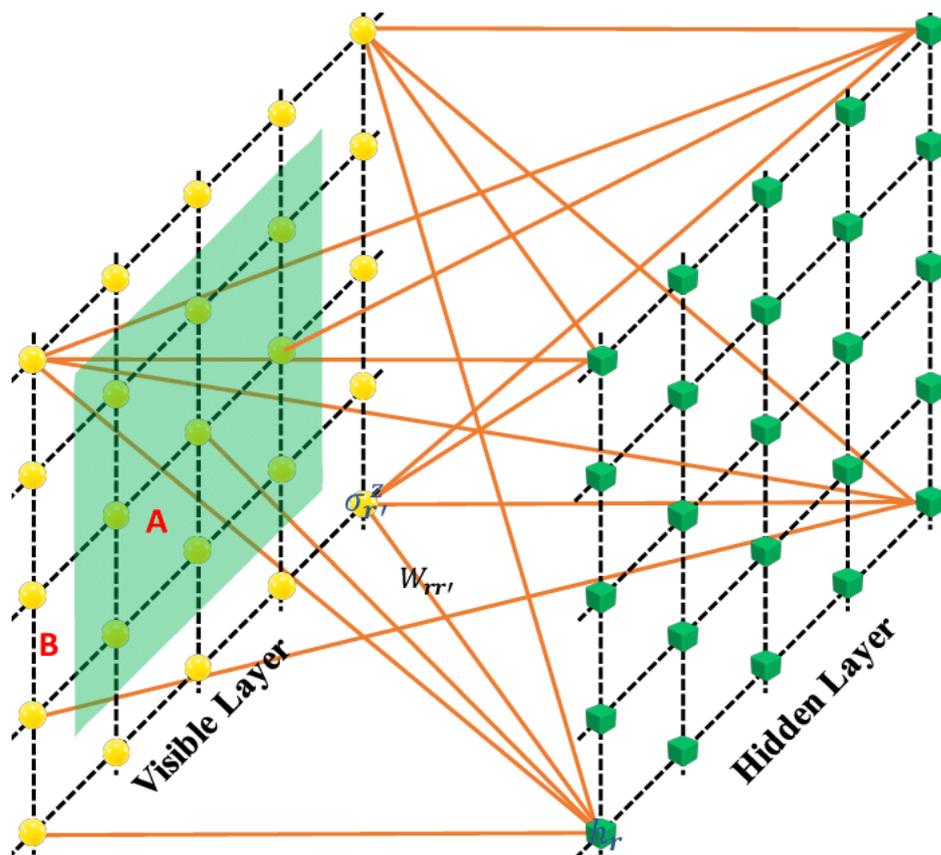
Probing Quantum Gravity Using Neural Quantum States

Student: Júlio Oliveira

Supervisor: Prof. João Manuel Viana Lopes

Co-Supervisor: Prof. João Miguel Penedones

April 2025



Abstract

This essay focuses on the research plan for my PhD, whose goal is to apply methods from artificial intelligence to problems in high-energy physics, with the emphasis on matrix quantum mechanics.

In the introduction (Sec.1), we will provide a brief overview of the importance of matrix quantum mechanics and the challenges associated with simulating these models. Then, we will introduce neural quantum states as a possible solution to the difficulties of simulating a many body quantum mechanics. Finally, we will compare the representability of neural quantum states with the more traditional tensor network ansatz.

The following sections will cover foundational concepts in machine learning, including an overview of neural networks (Sec.2) and their training methods (Sec.3). In Sec.4, we will review existing research on neural quantum states and explore their potential applications to matrix quantum mechanics and other areas of high-energy physics, such as S -matrix bootstrap.

Finally, in Sec.5, we will discuss open questions in the field of neural quantum states and outline a research plan aimed at studying specific models in matrix quantum mechanics.

Contents

1	Introduction	2
1.1	Matrix Quantum Mechanics and Quantum Gravity	2
1.2	The Hilbert Space is Huge	3
1.3	Neural Quantum State	4
2	Neural Networks	5
3	Training	6
4	Applications	8
4.1	Ground-State Search - Condensed Matter Models	8
4.2	Excited States	10
4.3	Ground-State Search High Energy Models	11
4.4	Beyond NQS - Other Applications of Artificial Intelligence in High Energy Physics	12
5	Objectives,Challenges & Research Plan	14
5.1	Objectives	14
5.2	Challenges	14
5.3	Research Plan	15
6	Conclusion	17

1 Introduction

1.1 Matrix Quantum Mechanics and Quantum Gravity

Understanding gravity at the quantum level remains an open question in fundamental physics. The holographic principle, Refs. [1, 2], and examples from AdS/CFT, Refs. [3, 4, 5], suggest that the geometric nature of spacetime dynamics may be an emergent approximate description of some system with an infinite number of strongly coupled degrees of freedom. In this PhD, we will be interested in a class of models called matrix models, usually called **Matrix Quantum Mechanics** (MQM). The simplest example of a MQM model would be the bosonic matrix model, whose Hamiltonian is

$$H_{\text{Bosonic}}(\mathbf{X}_I, \mathbf{X}_J) = \frac{1}{2} \text{Tr} \left(\mathbf{P}_I^2 + \mathbf{P}_J^2 + \mathbf{X}_I^2 + \mathbf{X}_J^2 - \frac{g^2}{2} [\mathbf{X}_I, \mathbf{X}_J]^2 \right), \quad (1.1)$$

where $\mathbf{X}_{I/J} = \sum_{\alpha} \tau_{\alpha} x_{I/J}^{\alpha}$, $\mathbf{P}_{I/J} = \sum_{\alpha} \tau_{\alpha} p_{I/J}^{\alpha}$, τ_{α} are the generators of $SU(N)$ such that $\text{Tr}(\tau_{\alpha} \tau_{\beta}) = \delta_{\alpha\beta}$. The ‘‘matrix’’ in matrix quantum mechanics comes from the possibility of organizing the Hamiltonian by writing its degrees of freedom, $x_{I/J}^{\alpha}$, as matrices. Although, these models do not have any spacetime, they can provide a non-perturbative formulation of some quantum field theories and string theories¹.

The study of MQM dates back to early stages of string theory and quantum gravity. A notable example is the Banks-Fischler-Shenker-Susskind (BFSS) model (Ref. [7, 8]), which describes M-theory in the infinite momentum frame. Furthermore, the spectrum of the BFSS model is continuous with no gap above the ground state. The finite energy states are scattering states. It is possible to introduce a mass deformation into BFSS preserving supersymmetry; that model is called the Berenstein-Maldacena-Nastase (BMN) model (Ref. [9]) and it is described by the following Hamiltonian

$$H_{\text{BMN}} = \text{Tr} \left(\frac{1}{2} \sum_{I=1}^9 P_I^2 - \frac{g^2}{4} [X^I, X^J]^2 - \frac{g}{2} \Theta^T \gamma^I [X^I, \Theta] \right) + \frac{1}{2} \text{Tr} \left(\frac{1}{3^2} \sum_{i=1}^3 (X^i)^2 + \frac{1}{6^2} \sum_{p=4}^9 (X^p)^2 + i \frac{1}{4} \Theta^T \gamma^{123} \Theta + i \frac{2g}{3} \epsilon_{ijk} X^i X^j X^k \right),$$

where X^I are the bosonic $N \times N$ matrices and Θ_{α} , with $\{\alpha = 1, \dots, 16\}$ are the Majorana fermions $N \times N$ matrices.

A closely related theory, the Ishibashi-Kawai-Kitazawa-Tsuchiya (IKKT), Ref. [10], model is a statistical model, whose partition function is an integral over matrices. The IKKT has been receiving increasing attention, because current evidence (Ref. [11, 12]) strongly suggests that in the large- N limit, this model can describe a 3+1 dimensional expanding universe (much like the one we live in).

The dualities between matrix models and quantum field theories are conjectured to happen at large- N , where N is the matrix size. The necessity to develop novel methods appears because the complexity of the matrix models grows with N . This has led to the development of advanced numerical methods,

¹ For a more comprehensive guide, see Ref. [6], where Maldacena explains how a matrix model with interactive bosons and Majorana fermions can give rise to the effective description of a charged black hole.

namely, Monte Carlo simulations (Refs. [13, 14, 15, 16]), bootstrap (Refs. [17, 18]), machine learning (Refs. [19, 20, 21]), quantum computing (Refs. [22, 20, 23]) and tensor networks (Ref. [24]).

1.2 The Hilbert Space is Huge

Since the early days of quantum mechanics, it has been clear that approximations are essential for addressing complex problems. In most cases, solving the Schrödinger equation for systems with more than a few interacting particles is not feasible. In certain regimes one can circumvent these difficulties with approximations such as perturbation theory, mean-field theory, variational methods, etc. Unfortunately, these methods often fail to capture all the relevant physics or break down when the interactions are strong.

For those reasons, numerical methods have become essential in tackling problems that cannot be solved analytically, especially in quantum mechanics. Numerical approaches such as exact diagonalization, Monte Carlo simulations, and density-matrix renormalization group (DMRG) have enabled physicists to explore complex systems with many interacting particles. However, even with these numerical tools, the complexity of many-body problems remains a challenge. The primary limitation of the numerical methods is the exponential scaling of the Hilbert space, frequently called exponential wall. For example, consider N particles with d continuous degree of freedom

$$|\psi\rangle = \int dx_1^1 \dots x_1^d, \dots, x_N^1 \dots x_N^d C(x_1^1 \dots x_1^d, \dots, x_N^1 \dots x_N^d) |x_1^1 \dots x_1^d, \dots, x_N^1 \dots x_N^d\rangle. \quad (1.2)$$

To naively store the wave function in a computer, we would need to discretize each degree of freedom. If we use Δ points in the discretization, the memory required would scale as

$$\text{Memory} \propto \Delta^{dN}. \quad (1.3)$$

Let's consider a system of 20 ($N = 20$) electrons 3D space ($d = 3$) with a spacial discretization of 2 points ($\Delta = 2$). The memory required to store this configuration would be approximately 2^{60} bytes or 18 exabytes, far more than any supercomputer currently has. Not to mention that a discretization of 2 points is frustratingly low.

Variational methods break the exponential wall by encoding the wave function into a set of parameters, θ , whose size scale polynomial with N and d ,

$$|\psi\rangle = \int dx_1^1 \dots x_1^d, \dots, x_N^1 \dots x_N^d \psi_\theta(x_1^1 \dots x_1^d, \dots, x_N^1 \dots x_N^d) |x_1^1 \dots x_1^d, \dots, x_N^1 \dots x_N^d\rangle, \quad (1.4)$$

where ψ_θ is called an ansatz. The task is then to find the set of parameters θ that most accurately describes the desired quantum state, such as the ground states of an Hamiltonian.

1.3 Neural Quantum State

Since 2018, artificial intelligence has been the hottest topic in computer science with proven applications in real world problems like self-driving and large language models. The central piece of this area is the **neural network**. From a mathematical point of view, a neural network is a parametrized map, ψ_{θ} , between the feature space (\mathcal{H}) and the prediction space (\mathcal{P}),

$$\begin{aligned} \mathcal{H} &\rightarrow \mathcal{P} \\ h \in \mathcal{H} &\rightarrow p \in \mathcal{P} = \psi_{\theta}(h). \end{aligned}$$

The symbols used here are intended to be highly suggestive. The neural network serves as an ansatz for the wave function, mapping a configuration of the system (the feature) to a probability amplitude (the prediction). This ansatz is referred to as the **Neural Quantum State (NQS)**. Since an ansatz is only as good as its expressive powers, it is crucial to compare the NQS to other ansätze. This comparison was done by Sharir et al. Ref. [25], who concluded that any tensor network ansatz can be expressed as a neural network. However, the reverse is not true, demonstrating that the NQS possesses superior expressive power when compared to a tensor network ansatz (see Fig. 1.1).

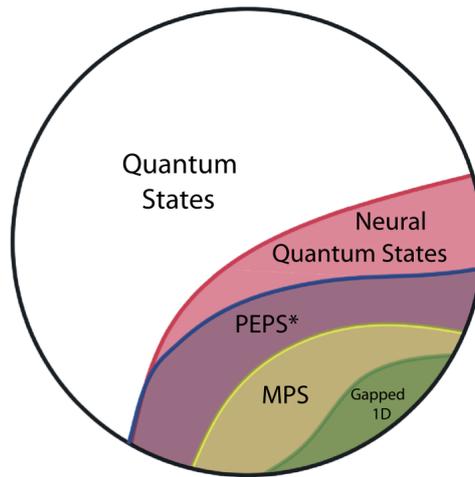


Fig. 1.1: Expressivity power of different variational ansätze compared to the NQS . This plot relates to a qudit system of N degrees of freedom and a polynomial number of parameters in N . (Fig. from [25])

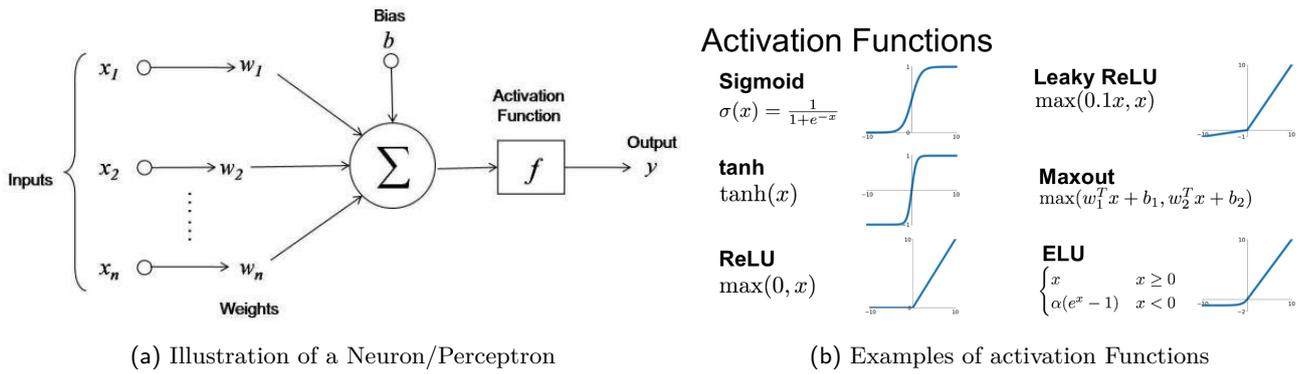
Furthermore, NQS aren't affected by the sign problem when fermions are introduced, like Monte Carlo. NQS does not render tensor networks (TN) or density matrix renormalization group (DMRG) obsolete. In certain cases, TN can be contracted in a way that eliminates the need for stochastic estimation of expectation values. Additionally, DMRG does not rely on gradient-based optimization techniques to determine its parameters.

2 Neural Networks

The fundamental building block of a neural network is the neuron/perceptron (Fig. 2.1a) . A neuron is a function that takes a vector input, \mathbf{x} , and outputs a number, y , given by

$$\mathbf{x} \rightarrow y = f(\mathbf{x}) = \varphi(\mathbf{x} \cdot \boldsymbol{\omega} + b), \quad (2.1)$$

where $\boldsymbol{\omega}$, b are the learnable parameters, called weights and bias, respectively, and φ is the activation function (Fig. 2.1b).



Similar to a LEGO set, one can combine neurons to create composite functions known as **neural networks** (NN). The "vanilla" NN, first introduced by Minsky and Papertis in 1969, is the multilayer perceptron (MLP). An MLP is composed of the one input layer, l^0 , a number of hidden layers, $l^0 < n < N$, and an output layer l^N , see Fig. 2.1. The MLP performs the following operations, it takes the output of the $(i - 1)$ th layers and applies the i th layer to it,

$$\mathbf{y}^{i-1} \rightarrow l^i(\mathbf{y}^{i-1}) = \mathbf{y}^i = \varphi_i(\mathbf{W}^i \cdot \mathbf{x} + \mathbf{b}^i), \quad (2.2)$$

where $\mathbf{y}^0 = \mathbf{x}$ and \mathbf{y}^i is the output of the i th layer; \mathbf{W}^i is the weight matrix of i th layer; \mathbf{b}^i is the bias vector of the i th layer and φ_i is the activation function of the i th layer². For a MLP with 3 hidden layers the output is $\mathbf{y}^0 = l^4(l^3(l^2(l^1(l^0(\mathbf{x})))))$. Neural networks in which each layer takes only the outputs of the previous layers as inputs are referred to as **feedforward neural networks** (FNN).

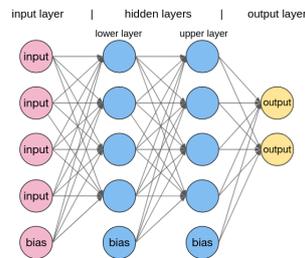


Fig. 2.1: Illustration of a Neuron/Perceptron

² If we were to use the MLP for an NQS, the learnable parameter are $\boldsymbol{\theta} = (\mathbf{W}^0, \dots, \mathbf{W}^N, \mathbf{b}^0, \dots, \mathbf{b}^N)$

3 Training

In section 2, we concluded that one can combine neurons to build different NQS. In this section, we will focus on the training process and discuss how to find the set of parameters that accurately represent the desired quantum state.

In typical industrial applications of machine learning, the parameter search is performed by optimizing a specific objective, commonly referred to as the **loss function**. For example, with a set of data y_{Data}^i organized in terms of a set of features (f_1^i, \dots, f_N^i) and a NN, N_θ , we can search for the best parameters that approximate $y_{\text{Data}}^i(f_1, \dots, f_N)$ by

$$\min_{\theta} L = \min_{\theta} \sum_{i \in \text{Data}} \left(y_{\text{Data}}^i - N_{\theta} \left(f_1^i, \dots, f_N^i \right) \right)^2. \quad (3.1)$$

Here, the NN serves as the extrapolator for values of y for the (f_1, \dots, f_N) that we do not have the y_{Data}^i . However, in physical applications, we typically lack knowledge of the wavefunction at different basis elements, meaning we cannot use the neural network as an extrapolator. Fortunately, for certain problems, there is a workaround. One such problem is finding the ground state of a quantum system. Formally, the ground state is the eigenstate of the Hamiltonian with the lowest eigenvalue, or in other words,

$$|\psi_0\rangle = \arg \min_{\psi} \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle}. \quad (3.2)$$

In Ref. [26], Carleo et. al realized that when ψ is a NQS parametrized by a set of parameters θ , then finding the ground state is equivalent to finding the parameters that minimize

$$L = \frac{\langle \psi_{\theta} | \hat{H} | \psi_{\theta} \rangle}{\langle \psi_{\theta} | \psi_{\theta} \rangle}. \quad (3.3)$$

Thus, provided we can compute the loss function, which we can using Monte Carlo, we are able to take advantage of the already developed machine learning technology to solve the ground state problem.

Why is this approach so exciting?

Besides the increased expressivity of the NQS, it is still unclear how machine learning tools can assist in solving this problem. When examining the minimization task at hand, one major challenge arises: computing derivatives both with respect to the inputs of the wavefunction and the parameters. Before explaining how to circumvent this issues, we should mention that computing derivatives is a crucial step both in the evaluation of L , since $\hat{P} \equiv \frac{\partial}{\partial x}$ appears in every continuous space hamiltonian, and optimization procedure, where one always needs to compute $\nabla_{\theta} L$. In computational physics 101, we learned that we can approximate the derivative of a function via discretization,

$$\frac{df}{dx} \approx \frac{f\left(x + \frac{\Delta}{2}\right) - f\left(x - \frac{\Delta}{2}\right)}{\Delta}. \quad (3.4)$$

This approach is feasible for functions with low-dimensional inputs, otherwise, the error becomes unmanageable. Thus, if we were to compute derivatives via discretization, we would be limited to an NQS with few parameters and problems involving a small number of particles, undermining the very motivation for using NQS in the first place. While it would also be possible to compute the derivatives by hand, as the complexity of the NQS increases, this becomes time-consuming and highly susceptible to human error. Instead, the machine learning community has developed an exact way of computing derivatives, called **automatic differentiation** (AD), Ref. [27]. Without delving into too much detail, it applies the chain rule of partial derivatives along with basic differentiation rules (e.g., the derivative of a sum) to compute the derivatives. In the typical code one would write the function, ψ , using elementary operation³ like those on the left of Fig. 3.1. The automatic differentiation (AD) routine applies the chain rule along with the differentiation rules for each elementary operation, resulting in a function that represents the derivative of ψ ($d\psi(1,0) = \frac{d\psi}{dx}$ and $d\psi(0,1) = \frac{d\psi}{dy}$). To exemplify, consider the following ansatz for the wave function

$$\psi(x, y) = \omega x + b + \sigma(\omega y), \quad (3.5)$$

where σ is the sigmoid function, see Fig. 3.1.

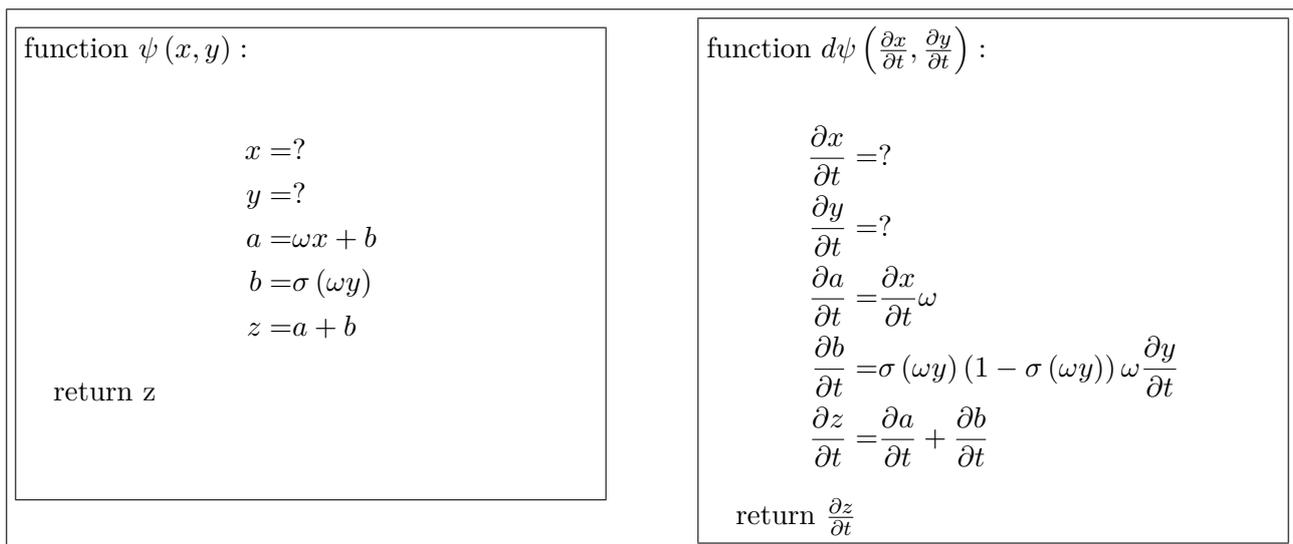


Fig. 3.1: Auto differentiation via forward accumulation

This particular routine belongs to the class of **forward accumulation**. More efficient algorithms of automatic differentiation (AD) fall under the class of **reverse accumulation**. Naturally, implementing this method efficiently is not trivial. Fortunately, the machine learning community has invested significant resources into developing these algorithms, and they are now available in open-source libraries such as JAX, PyTorch, and TensorFlow.

³ This are operation for which we know how to perform the differentiation explicitly

4 Applications

In this section, we will review the current state of the art for NQS and their application to both condensed matter and high-energy physics.

4.1 Ground-State Search - Condensed Matter Models

The first application of NQS in condensed matter were by Carleo et. al., Ref. [26]. In this paper, the authors used the following NQS, called Restricted Boltzmann Machine,

$$\Psi(S; W) = \sum_{\{h_i\}} e^{\sum_j a_j \sigma_j^z + \sum_i b_i h_i + \sum_{ij} W_{ij} h_i \sigma_j^z}, \quad (4.1)$$

where $h_i \in \{-1, 1\}$ are the so-called hidden spins, S is the input state and $W = \{a_j, b_j, W_{ij}\}$ are the learnable parameters.

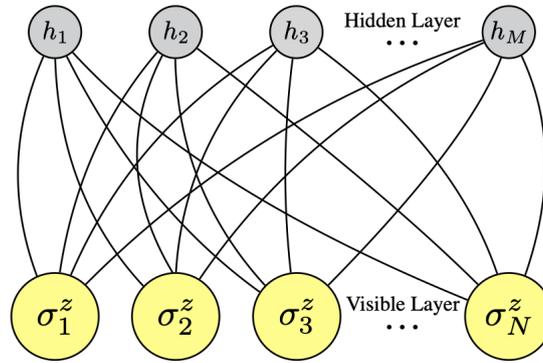


Fig. 4.1: Restricted Boltzmann Machine ansatz (Figure taken from Ref. [26]), here the visible layers represent the physical spins and hidden layers represent the hidden spins, which are responsible for encoding the correlation between the physical spins on the chain.

Using the ansatz in Fig. 4.1, the authors concluded that, for the transverse-field Ising model, the accuracy of the NQS was comparable to state-of-the-art matrix product state (MPS) methods. For the 2D antiferromagnetic Heisenberg model, they observed systematic improvements in the accuracy when compared to MPS⁴. This was back in 2017, of course the complexity of the neural networks has grown as well as the system they have been used (Ref. [28, 29]). In this PhD, we will focus on continuous systems. The NQS gained significant attention after its success in solving the many-electron Schrödinger equation [30, 31]. More recently in Refs. [32, 33], the authors re-introduced the **backflow ansatz** to find the ground state of the Fermi gas whose Hamiltonian is

$$H_{\text{fermi}} = -\frac{1}{2r_s} \sum_i \nabla_{\mathbf{r}_i}^2 + \frac{1}{r_s} \sum_{i < j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \text{const},$$

⁴ The authors also introduced a novel approach that uses machine learning to perform Unitary evolution of a quantum state.

where r_s is the Wigner-Seitz radius, and the constant arises from the interaction between the electrons and the positive background created by the nuclei. Although this is an all-to-all interactive model, the authors were able to find the ground state for 128 electrons. The main challenge in this case was finding a sufficiently expressive NQS that incorporates the Pauli exclusion principle,

$$\begin{aligned}\psi_\theta(\mathbf{R}) &= -\psi_\theta(\mathbf{P}_{ij}\mathbf{R}) \Leftrightarrow \\ \psi_\theta(\mathbf{r}_0, \dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \mathbf{r}_N) &= -\psi_\theta(\mathbf{r}_0, \dots, \mathbf{r}_j, \dots, \mathbf{r}_i, \mathbf{r}_N).\end{aligned}$$

This is where the **backflow transformations** comes in. The physical motivation is as follows: Suppose that at $\tau = -\infty$ (imaginary time), a system, whose ground state is $|\Phi_0\rangle$, is described by \hat{H}_0 . Then, a perturbation is introduced adiabatically such that at $\tau = \infty$ the system is described by $\hat{H} = \hat{H}_0 + \hat{V}$. According to evolution,

$$\Psi_\tau(\mathbf{X}) = \langle X | e^{-\tau\hat{H}} | \Psi_0 \rangle. \quad (4.2)$$

At sufficiently large τ , $\Psi_\tau(\mathbf{X})$ should be proportional to $\Psi_0(\mathbf{X})$. So as long as the Ψ_0 is a valid quantum state and the matrix elements of the time evolution operator remain finite, we can apply the mean value theorem to

$$\begin{aligned}\Psi(\mathbf{X}) &= \int_{\Omega} d\mathbf{X}' G_\tau(\mathbf{X}, \mathbf{X}') \Psi_0(\mathbf{X}') \\ &= \text{Vol}(\Omega) G_\tau(\mathbf{X}, \mathbf{Y}(\mathbf{X})) \Psi_0(\mathbf{Y}(\mathbf{X}))\end{aligned}$$

where $G_\tau(\mathbf{X}, \mathbf{X}') = \langle X | e^{-\tau\hat{H}} | X' \rangle$ and $\mathbf{Y}(\mathbf{X})$ is such that

$$G_\tau(\mathbf{X}, \mathbf{Y}(\mathbf{X})) \Psi_0(\mathbf{Y}(\mathbf{X})) = \int_{\Omega} d\mathbf{X}' \frac{G_\tau(\mathbf{X}, \mathbf{X}') \Psi_0(\mathbf{X}')}{\text{Vol}(\Omega)}. \quad (4.3)$$

By definition $G_\tau(\mathbf{X}, \mathbf{Y}(\mathbf{X}))$ must be positive so we can write the backflow ansatz as

$$\Psi(\mathbf{X}) = e^{J(\mathbf{X}, \mathbf{Y}(\mathbf{X}))} \Psi_0(\mathbf{Y}(\mathbf{X})). \quad (4.4)$$

Of course, the goal of all this is to impose the Pauli principle, so

$$\Psi(\mathbf{X}) = -\Psi(\mathbf{P}_{ij}\mathbf{X})$$

The solution with no interactions, Ψ_0 , already satisfies $\Psi_0(\mathbf{X}) = -\Psi_0(\mathbf{P}_{ij}\mathbf{X})$. Therefore if

$$\begin{cases} J(\mathbf{P}_{ij}\mathbf{X}, \mathbf{Y}(\mathbf{P}_{ij}\mathbf{X})) = J(\mathbf{X}, \mathbf{Y}(\mathbf{X})) & (1) \\ \mathbf{Y}(\mathbf{P}_{ij}\mathbf{X}) = \mathbf{P}_{ij}\mathbf{Y}(\mathbf{X}) & (2) \end{cases} \quad (4.5)$$

then $\Psi(\mathbf{X}) = -\Psi(\mathbf{P}_{ij}\mathbf{X})$. We are already familiar with Condition (1), which simply states that J

remains invariant under pairwise permutations. Condition (2) is more subtle: it requires that when two indices are permuted in X , the same two indices must also be permuted in $\mathbf{Y}(X)$. This property is known as **permutation equivariance**. In the backflow ansatz, J and \mathbf{Y} will be neural networks that obey condition (1) and (2) respectively. Examples of neural networks that exhibit permutation equivariance include **graph neural networks** (GNNs), **message-passing neural networks** (MPNNs), and **transformers**.

4.2 Excited States

Early approaches to find excited states relied on multiple neural networks (Ref. [34]) or computation of large determinants (Ref. [35]). Recently, in Ref. [36], Li et.al proposed a new NQS that takes the configuration of the system and outputs the the first N excited states of the system.

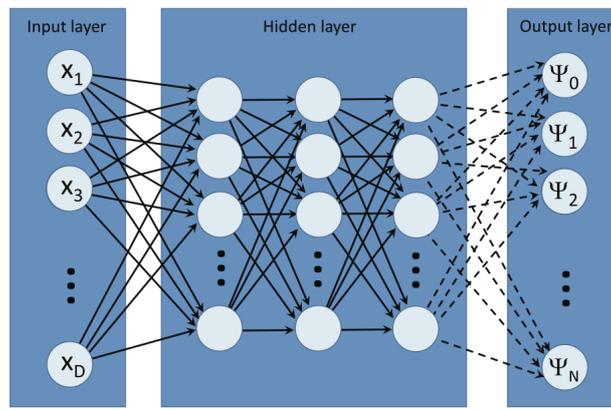


Fig. 4.2: Multi-state architecture taken from Ref. [36], where x_1, \dots, x_D are the quantum numbers, ψ_0, \dots, ψ_N are the first N eigenstates of the Hamiltonian and the dotted lines are the normalizing operation on the last layer.

The NQS in Fig. 4.2 is a MLP with 3 hidden layers and a normalizing layer⁵ whose outputs are

$$\psi_n = \sum_j a_{nj} \exp \left[\sum_{i=1}^D v_{ji}^2 x_i^2 \right] z_j, \quad (4.6)$$

where (a_{nj}, v_{ji}) are learnable coefficients and z_j are the outputs of the last hidden layer in the MLP. Obviously, the loss function we want to minimize is not the same as in the ground-state search. In Ref. [36], the authors propose the following loss function

$$L[\{\psi_n\}] = \sum_{n=0}^N \frac{\langle \psi_n | \hat{H} | \psi_n \rangle}{\langle \psi_n | \psi_n \rangle} + \beta \sum_{n=1}^N \sum_{m=0}^{n-1} \langle \psi_n | \psi_m \rangle^2 + \gamma \sum_n \left(\sum_j a_{nj}^2 - 1 \right)^2 + \frac{\Lambda}{2} \sum_{ji} \omega_{ij}^2, \quad (4.7)$$

where (β, γ, Λ) are hyper-parameters, and ω_{ij} are the learnable parameters within the MLP. The first issue with this loss function is that one can minimize the overlap between different states (the second

⁵ The authors introduced this layer, because they used a Monte Carlo method to compute the loss and its gradients that requires that $|\psi_n|^2$ is always normalizable for any n .

term) simply by making ψ_n small. The second, yet not so obvious remark, is that the first two terms in the loss function remain invariant under rotations of the set $\{\psi_n\}$. This implies that any linear combination of the Hamiltonian's eigenstates also minimizes these terms. One might argue that the third term could constrain the solutions, ensuring that only the true eigenstates of the Hamiltonian remain valid. However, this claim is difficult to prove. Moreover, we have been unable to reproduce the results reported in Ref. [36] or find other groups that have successfully done so.

4.3 Ground-State Search High Energy Models

The papers that use NQS to study of Matrix models are Refs. [21, 19, 20]. They use generative flow architectures, such as **normalizing flow** (NF), Ref. [37], and **masked autoregressive flow** (MAF). The big advantage of using generative flow architectures is their ability to efficiently sample from complex probability distributions, which in turn enhances the training process. Let Z be a random variable with a known probability density function (PDF), p_Z . If $\mathbf{Y} = g(Z)$ is an invertible function, then, using the change of variable formula, the PDF for Y is given by

$$p_Y(y) = p_Z(f(y)) |\det Dg(f(y))|^{-1}, \quad (4.8)$$

where f is the inverse of g and Dg is the Jacobian of g . As one might expect, f will be the neural network. Among other things, Eq. 4.8 tells us that if we can efficiently sample from p_Z (e.g. a normal distribution) and we know the transformation, f , such that $p_Y(y)$ matches a target distribution (e.g. $|\psi|^2$), then one can sample points from p_Z , apply f to those points and get a sample that follows the target distribution, p_Y (see Fig. 4.3).⁶

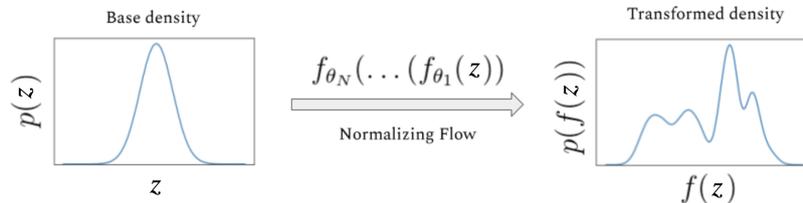


Fig. 4.3: Schematic normalizing flow where f is the composition of other elementary function f_{θ_i} .

In the application of NF, Refs. [19, 21], used NF as an ansatz for the square root of the wavefunction. Since the output of an NF is always a positive real number, this approach only works when there is a priori knowledge about the wavefunction. A clever workaround, following Ref. [19], is to express the contribution from fermionic modes as a phase factor that multiplies the bosonic part of the wavefunction,

$$|\Psi(X)\rangle = \sqrt{f(X)} |M(X)\rangle, \quad (4.9)$$

⁶ The MAF architecture follows the same principle, but with an elaborate way of parametrizing f such that the $Dg(f(y))$ is a triangular matrix, thus making the computation of $\det Dg(f(y))$ linear with the dimension of y .

where $f(X)$ is the normalizing flow and $|M(X)\rangle$ is a pure fermionic state⁷.

4.4 Beyond NQS - Other Applications of Artificial Intelligence in High Energy Physics

Artificial intelligence can also offer alternative solution to other problems in high-energy physics. One example is S -Matrix bootstrap. The S -matrix is defined as the amplitude of going from a initial state, $|i\rangle$, to a final state, $|f\rangle$, in a scattering process. For a $2 - 2$ scattering,

$$\langle p_3, p_4 | \mathbf{S} | p_1, p_2 \rangle = S(s, t, u) (2\pi)^d \delta^d(p_3 + p_4 - p_1 - p_2), \quad (4.10)$$

where s, t and u are the Mandelstam variables. The $S(s, t, u)$ cannot be any function it must be analytical in s, t, u ; crossing symmetric $S(s, t, u) = S(s, u, t) = S(u, t, s)$ and unitarity $\mathbf{S}^\dagger \mathbf{S} = \mathbb{I}$. In $1 + 1\text{D}$ (1 spatial and 1 temporal dimension) the conditions are

$$\begin{cases} S(s^*) = S^*(s) & \text{Analyticity} \\ S(s) = S(t = 4m^2 - s) & \text{Crossing} \\ |S(s)| \leq 1, s > 4m^2 & \text{Unitarity} \end{cases} \quad (4.11)$$

The bootstrap philosophy is to find the allowed values of S which respect the conditions in Eq. 4.11. Traditionally this is done by writing an ansatz for S , usually

$$S(s) = S_0 + \sum_{n=1}^{N_{max}} a_n [r_{s_0}^n(s) + r_{s_0}^n(4 - s)], \quad (4.12)$$

where $r_{s_0}(s) \equiv \frac{\sqrt{4-s_0} - \sqrt{4-s}}{\sqrt{4-s_0} + \sqrt{4-s}}$ and $S_0 \equiv S(s_0)$. Then one finds the parameters, (a_n, S_0) , that maximize a given functional for example, $\mathcal{F}[S] = |S[3]|$, and plot the allowed values of $S(s^*)$ for a given value of S_0 (See Fig. 4.4).

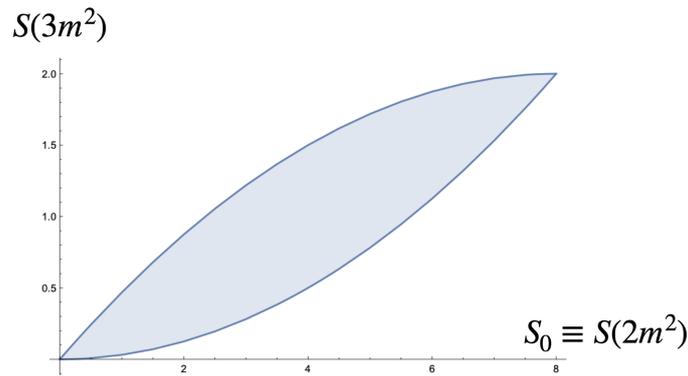


Fig. 4.4: Bootstrap results from Ref. [38] for $s_0 = 2$ and $s^* = 3$, for the allowed region for the values of $S(3m^2)$.

⁷ Schematically $|M(X)\rangle = \sum_i M_i(X) c_i^\dagger |0\rangle$, where $M_i(X)$ are complex coefficients and c_i^\dagger are the fermionic creation operators.

Recently, Gumus et. al. (Ref. [39]), proposed using a neural network as an ansatz for the S -matrix. This approach offers better control over the structural properties of the amplitude and may provide valuable insights into the microscopic details of the amplitudes generated in bootstrap studies. We are likely to explore potential contributions to this problem as well, leveraging the power of machine learning to enhance the understanding and predictions of scattering amplitudes in high-energy physics.

5 Objectives, Challenges & Research Plan

In this section, we will describe the main objectives for this PhD; the challenges in applying NQS to matrix quantum mechanics and devise a research plan for the upcoming 3 years.

5.1 Objectives

This PhD project aims to investigate the emergence of a gravitational description in matrix quantum mechanics (MQM). To achieve these, we propose to develop new methods based on artificial intelligence. This approach circumvents the limitations commonly found in traditional methods, like the sign-problem and exponential/factorial growth of the Hilbert space. Our main objectives are:

- 1) Develop a new efficient machinery for computing quantum states of strongly interactive many-body quantum systems. More concretely, create a neural quantum state that is capable of dealing with both bosonic and fermionic degrees of freedom, while preserving symmetries and respecting the Pauli Principle. As well as, improving the neural quantum state training.
- 2) Use these methods to compute observables in the BFSS and BMN models. Compare the results with the predictions from the dual gravitational description. This will constitute a very non-trivial test of the gauge/gravity duality.
- 3) Study the polarised IKKT matrix model with similar artificial intelligence methods.
- 4) Investigate which properties of MQM (or matrix models) are relevant for the emergence of a gravitational dual, e.g. “Is supersymmetry essential?”
- 5) Create a public database with trained NQS as open-source models, enabling researchers to use them as pre-trained tools for studying other MQM systems.

This PhD will explore how ideas from artificial intelligence, particularly neural quantum states, can be used to improve methods for computing ground and excited states in many-body quantum systems. Then, we’ll use those methods to study several matrix models with the goal of deepening our understanding of quantum gravity.

5.2 Challenges

Although Refs. [21, 19, 20] pioneered the idea of applying NQS to study Matrix Models, but most of the heavy lifting has [33] yet to be done. In the following paragraphs, we will explain some of these challenges and the strategies we will use to address them.

▷ Imposing Symmetries & Pauli Principle

With an infinite number of parameters, minimizing the Hamiltonian’s expectation value ensures the NQS converges to the true ground state, preserving all symmetries. In practice, a finite number of parameters only approximates these symmetries. Adding a penalty term to the loss function can inefficiently enforce those symmetries. However it requires multiple evaluations and hyper-parameter tuning, thus a slower training. A more efficient approach is to design a NQS that inherently respects

the desired symmetries. This has been implemented for a discrete symmetry group, C_6 , in Ref. [28] and for the Pauli Principle in Ref. [33]. In MQM, models are invariant under $SU(N)$ and $SO(M)$, where M is the number of bosons. A major challenge is to build a NQS that is invariant under those symmetries while also enforcing the Pauli principle for the fermionic sector.

▷ **Efficient Training**

Training a NQS requires Monte-Carlo sampling, thus introducing different time scales for sample thermalization and NQS evolution. To improve efficiency, samples from the previous step serve as the starting point for Monte-Carlo. If the NQS changes significantly, more Monte-Carlo iterations are required to re-thermalize the samples. And when NQS is not changing significantly the convergence is slow. Efficient training must balance sampling and optimization. Alongside Monte-Carlo expertise, we will explore novel NN like autoregressive flows (Refs. [19, 20, 21]) to enhance sampling.

▷ **Fermionic Sector**

In MQM, the Hilbert space grows exponentially with the number of fermions and N . A major challenge is to find a memory-efficient representation of the fermionic sector. Traditionally approaches truncate the Hilbert space, Ref. [19]. Recent ones, adapt the vision transformer architecture (Ref. [40]) to encode the wavefunction of spin models, such as the Heisenberg chain.

▷ **Degeneracy & Excited States**

Some MQM models have degenerate ground states. NQS struggles to find degenerate ground states or multiple excited states. Recent work, Ref. [36] suggests a multistate NQS, a NQS that outputs several wavefunctions, together with a custom loss can address this. Another approach uses independent NQS coupled by a loss function that enforces both orthogonality and minimal energy.

5.3 Research Plan

The strategy is to introduce each challenge through a new model, each with its own scientific merit. Regarding the measuring stage of each model, it only requires querying the NQS which is significantly easier than the training.

▷ **Task 1) Bosonic Matrix Model (Month 1-8)**

- This model is only composed of bosons. Our objective is to design a NQS that is invariant under $SO(M)$ and $SU(N)$ utilizing the blackflow transformation, Ref. [33].

Contingency Plan (CP): If this approach proves impractical, we will explore the use of group-convolutional NN, Ref. [28], as an alternative. If both fail, we can still introduce penalty terms in the loss to force the symmetries.

- Additionally, we will improve the Monte-Carlo sampling either by using the knowledge from other projects or by implementing new techniques such as autoregressive flows [19, 20, 21].

▷ **Task 2) Minimal-BMN (Month 9-14)**

-The minimal-BMN is the simplest supersymmetric MQM model. The primary challenge is constructing a representation for the fermionic sector. To address this, we will either truncate the Hilbert space of many-body fermions [19] or implement vision transformers, Ref. [40]. Further memory optimizations are possible, as the Hamiltonian can be written in blocks.

CP: Explore tensor networks, Ref. [24].

▷ **Task 3) Mini-BMN (Month 15-22)**

- The mini-BMN is the simplest supersymmetric MQM model with degenerate ground states. The primary challenge is enforcing the Pauli principle within the NQS. We aim to achieve this by using a message-passing NN, Ref. [33].

CP: Add penalty terms to the loss that forces the Pauli principle.

- The mini-BMN introduces the problem of a degenerate ground state. We plan to tackle this by either using multi-state NQS with custom loss, Ref. [36]. Using the same strategy we also try to extend the current formalism to the excited states.

CP: Use multiple NQS with a loss that imposes orthogonality and energetic constraints.

- Study several observables (e.g. the size $\langle \text{Tr} X^2 \rangle$) in the ground states as a function of the coupling and N.

▷ **Task 4) BMN & BFSS (Month 23-32)**

- The BMN and BFSS models consist of 9 bosonic and 16 fermionic matrices. After successfully studying the simpler models and solving the above-stated challenges, only minor code optimizations will be needed to extend our approach to the BMN and BFSS models. Thus completing Objective 1.

- Develop/adapt a neural network for measuring scattering amplitudes in BFSS, Ref. [41].

- Investigate the relevance of microscopic properties in the emergence of a gravitational description in the BMN and BFSS models (Objective 4). Possibly building a dictionary of all the properties in the Matrix Models and their consequences in the gravitational dual.

CP: Explore new algorithms in quantum computing Refs. [23, 22, 20] .

- Test existing conjectures, Ref. [42, 43], by studying the ground states of the BMN model as a function of the coupling and N (Objective 2).

▷ **Task 5) IKKT (Month 9-24)**

- Use similar techniques to Task 1 (with fermions), namely autoregressive flows, to compute distribution of physical observables. (Objective 3). Since the challenges are similar to the BMN model, knowledge can be transferred from the other tasks to this one.

▷ **Write Thesis (Month 32-36)**

6 Conclusion

Neural Quantum States present a promising path for addressing the long-standing challenges in Matrix Quantum Mechanics, ultimately providing insight into the world of quantum gravity. By leveraging deep learning techniques, we can potentially overcome the exponential complexity of Hilbert space and develop more efficient methods for simulating many-body quantum systems. Moving forward, the research plan aims to create NQS architectures that incorporate fundamental physical principles such as symmetry constraints and the Pauli exclusion principle. Additionally, improving the scalability of these methods is crucial for their application to large- N limit of matrix models. Ultimately, this research will contribute to the broader effort of applying artificial intelligence to fundamental physics, paving the way for more accurate simulations of quantum many-body systems as well provide insight into fundamental problems in quantum field theory.

References

- [1] G. 't Hooft. Dimensional reduction in quantum gravity.
- [2] L. Susskind. The world as a hologram. 36(11):6377–6396.
- [3] Juan M. Maldacena. The large n limit of superconformal field theories and supergravity. 38(4):1113–1133.
- [4] S. S. Gubser, I. R. Klebanov, and A. M. Polyakov. Gauge theory correlators from non-critical string theory. 428(1):105–114.
- [5] Edward Witten. Anti de sitter space and holography.
- [6] Juan Maldacena. A simple quantum system that describes a black hole.
- [7] T. Banks, W. Fischler, S. H. Shenker, and L. Susskind. M theory as a matrix model: A conjecture. 55(8):5112–5128.
- [8] Nathan Seiberg. Why is the matrix model correct? 79(19):3577–3580. Publisher: American Physical Society.
- [9] David Berenstein, Juan Maldacena, and Horatiu Nastase. Strings in flat space and pp waves from $\mathcal{N}=4$ super yang mills. 2002(4):013–013.
- [10] N. Ishibashi, H. Kawai, Y. Kitazawa, and A. Tsuchiya. A large- n reduced model as superstring. 498(1):467–491.
- [11] Shota Komatsu, Adrien Martina, Joao Penedones, Antoine Vuignier, and Xiang Zhao. Einstein gravity from a matrix integral – part II.
- [12] Shota Komatsu, Adrien Martina, João Penedones, Antoine Vuignier, and Xiang Zhao. Einstein gravity from a matrix integral – part i.
- [13] Evan Berkowitz, Enrico Rinaldi, Masanori Hanada, Goro Ishiki, Shinji Shimasaki, and Pavlos Vranas. Precision lattice test of the gauge/gravity duality at large- n . 94(9):094501.
- [14] Stratos Pateloudis, Georg Bergner, Masanori Hanada, Enrico Rinaldi, Andreas Schäfer, Pavlos Vranas, Hiromasa Watanabe, and Norbert Bodendorfer. Precision test of gauge/gravity duality in d0-brane matrix model at low temperature. 2023(3):71.
- [15] Konstantinos N. Anagnostopoulos, Masanori Hanada, Jun Nishimura, and Shingo Takeuchi. Monte carlo studies of supersymmetric matrix quantum mechanics with sixteen supercharges at finite temperature. 100(2):021601.
- [16] Georg Bergner, Norbert Bodendorfer, Masanori Hanada, Stratos Pateloudis, Enrico Rinaldi, Andreas Schäfer, Pavlos Vranas, and Hiromasa Watanabe. Confinement/deconfinement transition in the d0-brane matrix model – a signature of m-theory? 2022(5):96.
- [17] Xizhi Han, Sean A. Hartnoll, and Jorrit Kruthoff. Bootstrapping matrix quantum mechanics. 125(4):041601. Publisher: American Physical Society.

- [18] Henry W. Lin. Bootstrap bounds on d0-brane quantum mechanics. 2023(6):38.
- [19] Xizhi Han and Sean A. Hartnoll. Deep quantum geometry of matrices. 10(1):011069.
- [20] Enrico Rinaldi, Xizhi Han, Mohammad Hassan, Yuan Feng, Franco Nori, Michael McGuigan, and Masanori Hanada. Matrix model simulations using quantum computing, deep learning, and lattice monte carlo. 3(1):010324.
- [21] Norbert Bodendorfer, Onur Oktay, Vaibhav Gautam, Masanori Hanada, and Enrico Rinaldi. Variational monte carlo with neural network quantum states for yang-mills matrix model.
- [22] Hrant Gharibyan, Masanori Hanada, Masazumi Honda, and Junyu Liu. Toward simulating superstring/m-theory on a quantum computer. 2021(7):140.
- [23] H. L. Dao. Exploring new variational quantum circuit ansatzes for solving $SU(2)$ matrix models.
- [24] Enrico M. Brehm, Yibin Guo, Karl Jansen, and Enrico Rinaldi. Simulating matrix models with tensor networks.
- [25] Or Sharir, Amnon Shashua, and Giuseppe Carleo. Neural tensor contractions and the expressive power of deep neural quantum states. 106(20):205136.
- [26] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. 355(6325):602–606. Publisher: American Association for the Advancement of Science.
- [27] Andreas Griewank and Andrea Walther. *Evaluating Derivatives*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics.
- [28] Imelda Romero, Jannes Nys, and Giuseppe Carleo. Spectroscopy of two-dimensional interacting lattice electrons using symmetry-aware neural backflow transformations.
- [29] Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. 124(2):020503. Publisher: American Physical Society.
- [30] Kenny Choo, Antonio Mezzacapo, and Giuseppe Carleo. Fermionic neural-network states for ab-initio electronic structure. 11(1):2368. Publisher: Nature Publishing Group.
- [31] Jan Hermann, James Spencer, Kenny Choo, Antonio Mezzacapo, W. M. C. Foulkes, David Pfau, Giuseppe Carleo, and Frank Noé. Ab initio quantum chemistry with neural-network wavefunctions. 7(10):692–709. Publisher: Nature Publishing Group.
- [32] Jane Kim, Gabriel Pescia, Bryce Fore, Jannes Nys, Giuseppe Carleo, Stefano Gandolfi, Morten Hjorth-Jensen, and Alessandro Lovato. Neural-network quantum states for ultra-cold fermi gases.
- [33] Gabriel Pescia, Jannes Nys, Jane Kim, Alessandro Lovato, and Giuseppe Carleo. Message-passing neural quantum states for the homogeneous electron gas.
- [34] Kenny Choo, Giuseppe Carleo, Nicolas Regnault, and Titus Neupert. Symmetries and many-body excited states with neural-network quantum states. 121(16):167204.

- [35] David Pfau, Simon Axelrod, Halvard Sutterud, Ingrid von Glehn, and James S. Spencer. Accurate computation of quantum excited states with neural networks. 385(6711):eadn0137.
- [36] Hong Li, Qilong Zhai, and Jeff Z. Y. Chen. Neural-network-based multistate solver for a static schrödinger equation. 103(3):032405. Publisher: American Physical Society.
- [37] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows.
- [38] Hongbin Chen, A. Liam Fitzpatrick, and Denis Karateev. Bootstrapping 2d ϕ^4 theory with hamiltonian truncation data. 2022(2):146.
- [39] Mehmet Asim Gumus, Damien Leflot, Piotr Tourkine, and Alexander Zhiboedov. The s-matrix bootstrap with neural optimizers i: zero double discontinuity.
- [40] Riccardo Rende, Luciano Loris Viteritti, Federico Becca, Antonello Scardicchio, Alessandro Laio, and Giuseppe Carleo. Foundation neural-network quantum states.
- [41] Sebastian Mizera. Scattering with neural operators.
- [42] Jerzy Kowalski-Glikman. Vacuum states in supersymmetric kaluza-klein theory. 134(3):194–196.
- [43] Hai Lin, Oleg Lunin, and Juan Maldacena. Bubbling AdS space and 1/2 BPS geometries. 2004(10):025–025.