NEURAL NETWORKS FOR QUANTUM SYSTEMS

Supervised by: Michel FERRERO

Chayma FARAJI - Serena TANNOUS

ÉCOLE POLYTECHNIQUE
IP PARIS

TABLE DES MATIÈRES

ACKNOWLEDGEMENTS

We wish to sincerely thank **Prof. Michel Ferrero** and **Prof. Arnaud Couairon** as instructors of the Numerical Physics course. Their dedication in creating a stimulating learning environment and providing the necessary theoretical and computational foundation was invaluable to our study.

We are also immensely grateful to **Prof. Michel Ferrero** for his continuous guidance, availability, and constructive feedback as our project supervisor. His expertise and dedicated support were essential to the successful development of this work.

Finally, this project has been an invaluable experience for us, allowing us to deepen our understanding of numerical physics, explore the quantum many-body problem, and gain practical expertise in applying advanced computational methods to complex quantum systems.

1
INTRODUCTION

1.1 Context and motivation

The quantum many-body problem stands at the heart of modern theoretical physics, yet it remains notoriously difficult to solve. Its complexity originates from the structure of the many-body wave function : for a system of N interacting quantum particles, the number of parameters required to describe a generic state grows exponentially with N. As a consequence, even storing the full wave function becomes impossible for large systems, making traditional numerical simulations rapidly intractable.[1]

Over the years, a variety of computational strategies have been developed to tame this exponential growth. Quantum **Monte Carlo (QMC)** methods, for example, provide powerful stochastic techniques capable of exploring high-dimensional configuration spaces with considerable efficiency. Tensor-network approaches, such as **Matrix Product States (MPS)** or more advanced higher-dimensional generalizations, offer another avenue by exploiting entanglement-area laws to compress the quantum state into a more manageable form. Despite their successes, each of these methods has intrinsic **limitations**. QMC often suffers from the notorious **sign problem**, which dramatically restricts the class of models and parameter regimes it can address. Tensor-network methods, while extremely effective in one dimension, tend to scale poorly in higher dimensions or in systems with volume-law entanglement, where the required bond dimension becomes prohibitively large.[1]

Thus, for many systems of interest, such as frustrated magnets or strongly correlated fermions, valid computational methods are scarce. A general approach to compressing the wave function without exponential loss of information remains a significant open problem.

In parallel, the dramatic success of **Artificial Neural Networks** in modeling complex, high-dimensional probability distributions, evident in fields like image analysis and natural language processing, offered a novel conceptual solution. This breakthrough inspired the pioneering work of Giuseppe Carleo and Matthias Troyer, who formally introduced the concept of Neural-Network Quantum States (NQS) in 2017 [1]. Their seminal paper demonstrated that standard neural architectures, particularly **the Restricted Boltzmann Machine (RBM)**, can serve as a powerful and flexible variational ansatz for quantum many-body systems. [2]

Indeed, in the NQS framework, the wave function coefficients are directly parametrized by the network's weights. This approach leverages the expressive capacity of Deep Learning to encode complex quantum correlations.

1.2 Objectives of the project

The primary objective of this project is to implement and assess the Neural-Network Quantum State (NQS) approach to solve the many-body problem for the one-dimensional (1D) Transverse-Field Ising model.

The specific goals are as follows :

- **Neural Wave Function Ansatz :** Representing the unknown quantum ground state using an Artificial Neural Network, specifically utilizing the Restricted Boltzmann Machine (RBM) architecture to capture quantum correlations.
- **Variational Optimization :** Minimizing the expectation value of the Hamiltonian energy with respect to the network weights. This involves implementing the Variational Monte Carlo (VMC) method combined with advanced optimization techniques, specifically comparing Standard Gradient Descent (GD) against **Stochastic Reconfiguration (SR)**.
- **Validation and Accuracy :** Benchmarking the NQS accuracy against exact solutions for small systems ($N = 10$) to validate the implementation. The aim is to demonstrate high-precision convergence for the critical point ($J = H$).
- **Scalability to Large Systems :** Extending the simulation to macroscopic system sizes (up to $N = 50, 100...$) where exact diagonalization is intractable. We aim to validate these results using intrinsic metrics such as the energy variance (**V-score**), thereby showcasing the potential of NQS to overcome the exponential complexity of quantum many-body systems.

2 THEORETICAL BACKGROUND

2.1 The Ising Model

The Ising model is one of the most studied systems in statistical and condensed-matter physics. Originally introduced to describe ferromagnetism, it has since become a paradigmatic model for understanding phase transitions, critical phenomena, and the behavior of interacting binary variables in a wide range of systems such as alloys, neural networks, or liquid-gas transitions [3].

In its classical one-dimensional form, the model consists of a chain of N spins $s_i = \pm 1$, with nearest-neighbor interactions described by the Hamiltonian

$$H_{\text{Ising}} = -J \sum_i s_i s_{i+1}, \tag{1}$$

where J is the coupling constant. A positive $J > 0$ favors ferromagnetic alignment of neighboring spins, while $J < 0$ corresponds to antiferromagnetic coupling.

This Hamiltonian corresponds to the **classical** Ising model, where all spin variables commute and configurations are simply classical bitstrings. An example of a classical spin configuration is shown in Fig. 2.

In this work, however, we focus on the **Transverse-Field Ising Model (TFIM)**, a quantum extension in which the system is driven by a magnetic field applied along the transverse (x) direction. The resulting Hamiltonian reads :

$$\mathcal{H}_{\text{TFIM}} = -H \sum_i \hat{\sigma}_i^x - J \sum_i \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z, \tag{2}$$

where g is the strength of the transverse magnetic field and $\hat{\sigma}^{x,y,z}$ denote the Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The TFIM introduces quantum fluctuations through the non-commuting terms $[\sigma_i^x, \sigma_i^z] \neq 0$, leading to a quantum phase transition at zero temperature. The competition between the two terms defines two distinct quantum phases :

- **Ferromagnetic Phase ($H < J$) :** Dominated by the spin-spin coupling, the ground state is ordered, exhibiting a finite magnetization along the z-axis.
- **Paramagnetic Phase ($H > J$) :** Dominated by the transverse field, the ground state is disordered, with spins aligned primarily along the x-axis.

The transition between these two phases occurs at the **Quantum Critical Point** where $H = J$. The strong non-local entanglement present at this critical point is precisely why the TFIM serves as a rigorous testing ground for variational algorithms like the RBM.

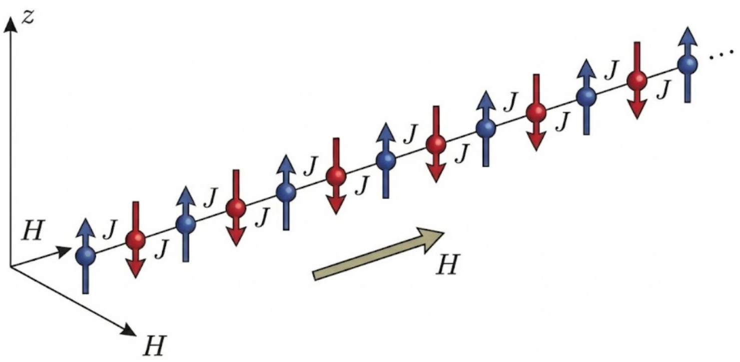


Figure 1 – Illustration of a classical spin configuration in the 1D Ising model [3].

2.2 Quantum Many-Body Systems

The fundamental challenge in quantum many-body physics lies in the structure of the Hilbert space. For a system composed of N spin-1/2 particles (such as the Ising model), the Hilbert space $\mathscr{H}$ is the tensor product of the individual spin spaces :

$$\mathscr{H} = \bigotimes_{i=1}^{N} \mathbb{C}^2. \tag{3}$$

The dimension of this space is $D_{\mathscr{H}} = 2^N$. A generic quantum state $|\Psi\rangle$ can be expanded in the computational basis $\mathcal{S} = (s_1, \ldots, s_N)$ as :

$$|\Psi\rangle = \sum_{\mathcal{S}} \Psi(\mathcal{S}) \, |\mathcal{S}\rangle, \tag{4}$$

where $\Psi(\mathcal{S}) \in \mathbb{C}$ represent the complex amplitudes associated with each configuration.

To fully describe such a state, one must store 2^N complex coefficients. This exponential scaling is known as the **curse of dimensionality**. For small systems (e.g., $N = 20$), the vector size is manageable ($\approx 10^6$ entries). However, for $N = 100$, the dimension $2^{100} \approx 10^{30}$ exceeds the storage capacity of any conceivable supercomputer.

The goal of numerical many-body physics is to find the ground state $|\Psi_0\rangle$ satisfying the Schrödinger equation :

$$\hat{H}|\Psi_0\rangle = E_0|\Psi_0\rangle, \tag{5}$$

where E_0 is the lowest energy eigenvalue. Since exact diagonalization is impossible for large N, we must rely on **variational representations**. The central idea is to approximate the high-dimensional vector $\Psi(\mathcal{S})$ using a function parameterized by a manageable set of parameters $\mathcal{W}$:

$$\Psi(\mathcal{S}) \approx \Psi_{\mathcal{W}}(\mathcal{S}). \tag{6}$$

If the number of parameters $|\mathcal{W}|$ scales polynomially with N (e.g., $O(N)$ or $O(N^2)$), the representation becomes computationally efficient. The success of this approach depends entirely on the expressive power of the ansatz $\Psi_{\mathcal{W}}$, that is, its ability to capture the non-trivial quantum correlations (entanglement) present in the physical ground state.

2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational models inspired by the biological structure of the brain, designed to approximate complex, high-dimensional functions. In mathematical terms, an ANN is a non-linear map

$$f_{\mathcal{W}} : \mathbb{R}^n \to \mathbb{R}^m, \tag{7}$$

parameterized by a set of weights and biases $\mathcal{W}$.

The fundamental motivation for using ANNs in quantum physics stems from the **Universal Approximation Theorem**. This theorem states that a feed-forward neural network with a single hidden layer and a sufficient number of neurons can approximate any continuous function on a compact subset of $\mathbb{R}^n$ to arbitrary precision.

In the context of the Neural-Network Quantum States (NQS) introduced by Carleo and Troyer [1], we treat the many-body wavefunction as a function that maps a spin configuration to a complex amplitude :

$$\Psi_{\mathcal{W}} : \{-1, 1\}^N \to \mathbb{C}. \tag{8}$$

Here, the inputs to the network are the spin values $(s_1, \ldots, s_N)$, and the output is the wavefunction amplitude $\Psi(\mathcal{S})$.

By interpreting the wavefunction as a "black box" machine that assigns a probability amplitude to every configuration, we leverage the ability of neural networks to learn patterns in high-dimensional data. Specifically, for the Ising model, the network learns to identify the spin correlations (such as ferromagnetic ordering) that lower the energy of the system.

3
RESTRICTED BOLTZMANN MACHINE (RBM)

3.1 Architecture

In this project, we approximate the ground state of the one-dimensional Ising model using a *Neural-Network Quantum State* (NQS) based on a Restricted Boltzmann Machine (RBM). This variational quantum-wavefunction ansatz was introduced by Carleo and Troyer [1] and has since become a standard tool in variational quantum Monte Carlo (VMC).

RBM Wavefunction Ansatz : We consider an Ising spin configuration

$$S = (s_1, s_2, \ldots, s_N), \qquad s_j = \pm 1.$$

The RBM represents the corresponding wavefunction amplitude as

$$|\Psi\rangle = \sum_S \Psi(S) |S\rangle \tag{9}$$

Such that :

$$\Psi(S) = \exp\left(\sum_{j=1}^{N} a_j s_j\right) \prod_{i=1}^{M} 2\cosh\left(b_i + \sum_{j=1}^{N} W_{ij} s_j\right), \tag{10}$$

where :

- — a_j are visible biases,
- — b_i are hidden biases,
- — W_{ij} are coupling weights between visible and hidden spins,
- — M is the number of hidden units.

This network contains no intra-layer connections, making hidden spins conditionally independent and allowing analytic tracing over hidden variables.

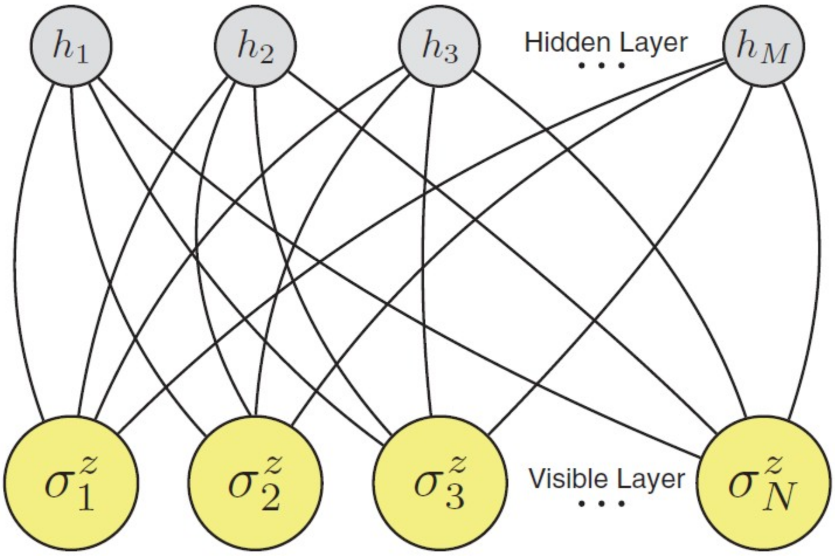


Figure 2 – Artificial neural network encoding a many-body quantum state of N spins. A restricted Boltzmann machine architecture that features a set of N visible artificial neurons (yellow dots) and a set of M hidden neurons (gray dots) is shown.[1]

3.2 RBM as a Wave Function Ansatz

The RBM amplitude before summing out the hidden layer is :

$$\Psi(S;W) = \sum_{\{h_i=\pm1\}} \exp\left(\sum_j a_j s_j + \sum_i b_i h_i + \sum_{i,j} W_{ij} h_i s_j\right). \tag{11}$$

Define for each hidden unit

$$X_i = b_i + \sum_j W_{ij} s_j.$$

Then

$$\Psi(S;W) = \exp\left(\sum_j a_j s_j\right) \prod_{i=1}^{M} \left[\sum_{h_i=\pm1} e^{h_i X_i}\right].$$

The remaining sum is elementary :

$$\sum_{h_i=\pm 1} e^{h_i X_i} = e^{X_i} + e^{-X_i} = 2\cosh(X_i).$$

Thus the RBM wavefunction takes the compact form of Eq. (10).

However, in order to avoid numerical **overflow**, we evaluate $\log\Psi(S)$ instead of $\Psi(S)$. The RBM wavefunction contains products of exponential and hyperbolic cosine terms, which grow extremely rapidly and can easily exceed floating-point numerical limits. This leads to infinities or undefined values. Using $\log\Psi(S)$ keeps all computations within a stable numerical range :

$$\log\Psi(S) = \sum_j a_j s_j + \sum_{i=1}^{M} \log\left[2\cosh\left(b_i + \sum_j W_{ij} s_j\right)\right].$$

4 TRAINING AND OPTIMIZATION : VARIATIONAL MONTE CARLO (VMC)

4.1 Variational principle

The foundation of this approach is the Quantum Variational Principle. It states that for any normalized trial wavefunction $\Psi_{\mathcal{M}}$ corresponding to a physical system with Hamiltonian $\mathcal{H}$, the expectation value of the energy $E(\mathcal{W})$ is always greater than or equal to the true ground state energy E_0 :

$$E(\mathcal{W}) = \frac{\langle\Psi|\mathcal{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} \geq E_0 \tag{12}$$

The ground state energy of the system is thus found by minimizing the expectation value of the Hamiltonian $\mathcal{H}$ with respect to the variational parameters $\mathcal{W} = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ of the RBM wavefunction $\Psi_{\mathcal{M}}(\mathcal{W})$. This minimization is achieved efficiently using the **Variational Monte Carlo (VMC)** method, which operates as a quantum reinforcement learning scheme.

4.2 VMC - Energy Estimation :

The exponential growth of the Hilbert space dimension (2^N for N spins) makes exact summation intractable. We therefore employ Monte Carlo integration, but this requires reformulating the energy expectation value into a suitable form.

Starting from the variational energy that we write in the basis of configurations $|\mathcal{S}\rangle$:

$$E(\mathcal{W}) = \frac{\langle\Psi|\hat{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \frac{\sum_{\mathcal{S},\mathcal{S}'} \Psi^*(\mathcal{S})\langle\mathcal{S}|\hat{H}|\mathcal{S}'\rangle\Psi(\mathcal{S}')}{\sum_{\mathcal{S}} |\Psi(\mathcal{S})|^2} \tag{13}$$

Then :

$$E(\mathcal{W}) = \frac{\sum_{\mathcal{S}} |\Psi(\mathcal{S})|^2 \cdot \left[\frac{\sum_{\mathcal{S}'} \langle\mathcal{S}|\hat{H}|\mathcal{S}'\rangle\Psi(\mathcal{S}')}{\Psi(\mathcal{S})}\right]}{\sum_{\mathcal{S}} |\Psi(\mathcal{S})|^2} \tag{14}$$

This defines the **local energy** :

$$E_{\text{loc}}(\mathcal{S}) = \frac{\langle\mathcal{S}|\hat{H}|\Psi\rangle}{\Psi(\mathcal{S})} = \frac{\sum_{\mathcal{S}'} \langle\mathcal{S}|\hat{H}|\mathcal{S}'\rangle\Psi(\mathcal{S}')}{\Psi(\mathcal{S})} \tag{15}$$

The energy can now be expressed as an expectation value over the quantum probability distribution :

$$E(\mathcal{W}) = \sum_{\mathcal{S}} P(\mathcal{S}) \cdot E_{\text{loc}}(\mathcal{S}) \approx \frac{1}{N_{\text{samples}}} \sum_{k=1}^{N_{\text{samples}}} E_{\text{loc}}(\mathcal{S}_k) \tag{16}$$

where :

- $P(\mathcal{S}) = \frac{|\Psi_{\mathcal{W}}(\mathcal{S})|^2}{\sum_{\mathcal{S}'} |\Psi_{\mathcal{W}}(\mathcal{S}')|^2}$ is the quantum probability distribution
- $E_{\text{loc}}(\mathcal{S}) = \frac{\langle\mathcal{S}|\hat{H}|\Psi\rangle}{\Psi(\mathcal{S})}$ is the local energy, representing the energy "felt" by configuration $\mathcal{S}$ in the quantum state $|\Psi\rangle$

4.3 Steps of VMC

Phase 1 : Sampling using Metropolis Hastings The Monte Carlo sampling in VMC requires generating a sequence of spin configurations $\{\mathcal{S}_k\}$ distributed according to the target probability distribution :

$$P(\mathcal{S}) = \frac{|\Psi(\mathcal{S})|^2}{\sum_{\mathcal{S}'} |\Psi(\mathcal{S}')|^2} \propto |\Psi(\mathcal{S})|^2$$

The standard Metropolis-Hastings acceptance ratio $A(\mathcal{S} \to \mathcal{S}')$ for a proposed move from configuration $\mathcal{S}$ to $\mathcal{S}'$ is defined as :

$$A(\mathcal{S} \to \mathcal{S}') = \min\left(1, \quad R\right) \tag{17}$$

where R is the ratio of probabilities and transition rates :

$$R = \frac{P(\mathcal{S}') \cdot T(\mathcal{S}' \to \mathcal{S})}{P(\mathcal{S}) \cdot T(\mathcal{S} \to \mathcal{S}')}$$

In the VMC implementation, the trial move is typically a single spin flip at a randomly chosen site i.

- The transition probability $T(\mathcal{S} \to \mathcal{S}')$ to propose a move from $\mathcal{S}$ to $\mathcal{S}'$ is the same as the transition probability $T(\mathcal{S}' \to \mathcal{S})$ to propose the reverse move. This is because both transitions only involve a single, specific spin flip, and the probability of selecting that spin is $1/N$, regardless of the direction.
- Therefore, the proposal rates cancel out : $T(\mathcal{S}' \to \mathcal{S}) = T(\mathcal{S} \to \mathcal{S}')$.

The ratio R simplifies to the ratio of the target probabilities :

$$R = \frac{P(\mathcal{S}')}{P(\mathcal{S})}$$

Since the target distribution $P(\mathcal{S})$ is proportional to the square of the wavefunction amplitude $|\Psi(\mathcal{S})|^2$, we can substitute this into the ratio R. The normalization constants cancel out :

$$R = \frac{|\Psi(\mathcal{S}')|^2}{|\Psi(\mathcal{S})|^2} = \left(\frac{|\Psi(\mathcal{S}')|}{|\Psi(\mathcal{S})|}\right)^2$$

Using the logarithm of the wavefunction the final form is obtained :

$$R = \exp\left(2\left[\log|\Psi(\mathcal{S}')| - \log|\Psi(\mathcal{S})|\right]\right)$$

Phase 2 : Gradient Calculation :

- To minimize $E(\mathcal{W})$, we use first **the Stochastic Gradient Descent (SGD)** method, iteratively updating the parameters in the direction opposite to the gradient : $\mathcal{W}_{\text{new}} = \mathcal{W}_{\text{old}} - \eta\nabla_{\mathcal{W}}E(\mathcal{W})$, where η is the learning rate.

For $\lambda \in \mathcal{W}$ E is derived as follows :

$$\frac{\partial E}{\partial \lambda} = \frac{\partial}{\partial \lambda}\left[\frac{\langle\Psi|\hat{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle}\right] \qquad = \frac{\langle\partial_\lambda\Psi|\hat{H}|\Psi\rangle + \langle\Psi|\hat{H}|\partial_\lambda\Psi\rangle}{\langle\Psi|\Psi\rangle} - \frac{\langle\Psi|\hat{H}|\Psi\rangle\partial_\lambda\langle\Psi|\Psi\rangle}{\langle\Psi|\Psi\rangle^2} \tag{18}$$

Using the Hermitian property of $\hat{H}$, $\langle\partial_\lambda\Psi|\hat{H}|\Psi\rangle = \langle\Psi|\hat{H}|\partial_\lambda\Psi\rangle^*$ we obtain :

$$\frac{\partial E}{\partial \lambda} = 2\left[\frac{\langle\partial_\lambda\Psi|\hat{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} - \frac{\langle\Psi|\hat{H}|\Psi\rangle\langle\partial_\lambda\Psi|\Psi\rangle}{\langle\Psi|\Psi\rangle^2}\right] \tag{19}$$

The logarithmic derivative for a parameter $\lambda \in \mathcal{W}$ and a configuration $\mathcal{S}$ is defined as :

$$O_\lambda(\mathcal{S}) = \frac{\partial \log \Psi(\mathcal{S})}{\partial \lambda} = \frac{\partial_\lambda \Psi(S)}{\Psi(S)}$$

So :

$$\langle\partial_\lambda\Psi|\hat{H}|\Psi\rangle = \langle\Psi|O_\lambda\hat{H}|\Psi\rangle$$

And :

$$\langle\partial_\lambda\Psi|\Psi\rangle = \langle\Psi|O_\lambda|\Psi\rangle$$

we can express the derivative as :

$$\frac{\partial E}{\partial \lambda} = 2[\langle O_\lambda \hat{H} \rangle - \langle O_\lambda \rangle \langle \hat{H} \rangle] \tag{20}$$

We put $\langle \hat{H} \rangle = \langle E_{loc} \rangle = E$
Since :

$$\langle O_\lambda \hat{H} \rangle = \frac{\sum_S |\Psi(S)|^2 \, O_\lambda(S) \, \frac{(\hat{H}\Psi)(S)}{\Psi(S)}}{\sum_S |\Psi(S)|^2} = \frac{\sum_S |\Psi(S)|^2 \, O_\lambda(S) \, E_{\mathrm{loc}}(S)}{\sum_S |\Psi(S)|^2} \tag{21}$$

We obtain :

$$\frac{\partial E}{\partial \lambda} = 2 \left[\langle O_\lambda E_{\mathrm{loc}} \rangle - E \langle O_\lambda \rangle \right]$$

Recognizing that $E = \langle E_{\mathrm{loc}} \rangle$, the gradient can be written as a covariance :

$$\nabla_\lambda E = 2 \cdot \mathrm{Cov}(O_\lambda, E_{\mathrm{loc}}) = 2 \left[\langle O_\lambda E_{\mathrm{loc}} \rangle - \langle O_\lambda \rangle \langle E_{\mathrm{loc}} \rangle \right] \tag{22}$$

we still have to compute O_λ
The components of $O_\lambda(\mathcal{S})$ are :

$$O_\lambda(\mathcal{S}) = \begin{cases} O_{a_k}(\mathcal{S}) = s_k \\ O_{b_k}(\mathcal{S}) = \tanh\left(b_k + \sum_j W_{kj} s_j \right) \\ O_{W_{kl}}(\mathcal{S}) = s_l \tanh\left(b_k + \sum_j W_{kj} s_j \right) \end{cases}$$

Using vector notation, where functions (such as tanh) are applied element-wise, the logarithmic derivatives become :

$$\begin{cases} O_{\mathbf{a}}(\mathcal{S}) = \mathcal{S} \\ O_{\mathbf{b}}(\mathcal{S}) = \tanh(\mathbf{b} + \mathbf{W}\mathcal{S}) \\ O_{\mathbf{W}}(\mathcal{S}) = \tanh(\mathbf{b} + \mathbf{W}\mathcal{S}) \, \mathcal{S}^T \end{cases}$$

— In addition to standard Gradient Descent, we also implement the **Stochastic Reconfiguration (SR)** method, introduced by *Sorella*, which provides a more stable and geometry-aware optimization strategy for Variational Monte Carlo.

The central idea of SR is : Instead of performing updates in the Euclidean parameter space, SR modifies the descent direction by projecting it onto the tangent space of normalized wavefunctions. [4]

Starting from the gradient of the energy

$$g_\lambda = \frac{\partial E}{\partial \lambda} = 2\left[\langle O_\lambda E_{\text{loc}}\rangle - \langle O_\lambda\rangle\langle E_{\text{loc}}\rangle\right],$$

SR defines the parameter update as the solution of the linear system

$$\sum_\mu S_{\lambda\mu}\,\Delta\mathcal{W}_\mu = -\eta\, g_\lambda, \quad \text{with } \eta \text{ the SR learning rate.} \tag{23}$$

The matrix $S_{\lambda\mu}$ is the *quantum Fisher information matrix* (also called the covariance matrix of the logarithmic derivatives), defined as :

$$S_{\lambda\mu} = \langle O_\lambda O_\mu\rangle - \langle O_\lambda\rangle\langle O_\mu\rangle. \tag{24}$$

In practice, the updates for the parameters are obtained by solving

$$\Delta\mathcal{W} = -\eta\, S^{-1} g.$$

SR therefore provides a natural-gradient-like update that takes into account correlations between parameters and the curvature of the variational manifold. This generally leads to faster convergence and improved stability compared to SGD, sensitivity to the choice of learning rate. [4]

Phase 3 : Parameter Update In summary, while Gradient Descent updates parameters as

$$\mathcal{W}_{\text{new}} = \mathcal{W}_{\text{old}} - \eta\, g,$$

Stochastic Reconfiguration instead performs the geometry-aware update

$$\mathcal{W}_{\text{new}} = \mathcal{W}_{\text{old}} - \eta\, S^{-1} g.$$

This allows us to compare both optimization strategies in our implementation of VMC with an RBM ansatz for the Ising model.

This iterative cycle (sampling, gradient calculation, and update) is repeated for `n_epochs`, driving the system towards the ground state, where $\nabla_{\mathcal{W}} E \approx 0$ and $E(\mathcal{W}) \approx E_0$.

5 NUMERICAL IMPLEMENTATION

Our simulations and analyses are built upon four fundamental Python modules, each addressing a critical component of the Variational Monte Carlo (VMC) approach to the 1D Transverse Field Ising Model (TFIM) (Section 4). This section details the specific numerical techniques employed within these modules.

CONFIGURATION MANAGEMENT (ISING.PY)

The `Ising.py` module defines the basic spin configuration and includes a classical Metropolis sampler, primarily for initialization and ensuring periodic boundary conditions.

In our model, the physical state is represented by a 1D array of spins $s \in \{-1, 1\}^N$. We have to consider the boundary conditions, for that, the calculation of the energy change ΔE in the classical `metropolis_step` uses the modulo operator (`% L`) to enforce **periodic boundary conditions** (PBCs), ensuring the spin s_N interacts with s_1.

RBM

Based on the theoretical part (Section 4), we implemented two main classes of RBM :

— RBM with Gradient Descent (`RBM_GD.py`) : implements the core RBM variational ansatz $\Psi_{\mathcal{W}}(\sigma)$ and optimizes its parameters $\mathcal{W}$ using standard Gradient Descent.

— RBM with Stochastic Reconfiguration (`RBM_SR.py`) :It introduces the Stochastic Reconfiguration (SR) method to accelerate convergence by incorporating the geometry of the variational manifold.

A curcial point to mention here is that : Since the correlation matrix S can be ill-conditioned or noisy due to finite Monte Carlo sampling, we regularize it using a small diagonal shift :

$$S^{(\mathrm{reg})}_{\lambda\mu} = S_{\lambda\mu} + \epsilon\,\delta_{\lambda\mu}, \tag{25}$$

with typically $\epsilon \sim 10^{-3}$–10^{-2}. This is crucial for smooth and stable training convergence.

EXACT DIAGONALIZATION FOR VALIDATION (EXACT_ENERGY.PY)

This module is crucial for establishing the ground state energy, E_0, which serves as the ground truth to **benchmark the accuracy** of our VMC results.

The key challenge in Exact Diagonalization (ED) is the exponential scaling of the Hilbert space, where the Hamiltonian matrix size is $D \times D$ with $D = 2^N$. To manage this complexity, we employ two crucial numerical strategies :

1. The Hamiltonian is constructed and stored using the **Compressed Sparse Row (CSR) format** (`scipy.sparse.csr_matrix`). This format stores only the non-zero matrix elements, which minimizes **memory usage** for sparse matrices.
2. Instead of calculating the full spectrum of eigenvalues, which is computationally prohibitive, we use the efficient **Arnoldi iteration** method, specifically `scipy.sparse.linalg.eigsh`. This method is optimized to find only the lowest eigenvalue, E_0, which corresponds to the ground state energy.

6 ANALYSIS

The analysis of our VMC simulations focuses on two primary objectives : first, assessing the performance and stability of the optimization strategies (GD vs. SR) based on the hyperparameters, and second quantifying the expressive power of the RBM ansatz with different lengths of Ising chain. The system is studied at the quantum critical point ($J = 1, H = 1$) to probe the network's ability to capture strong quantum correlations. (As we justified in section 2.1)

6.1 Preliminary Observation

We define the primary metric for accuracy as the **Relative Error**, ϵ_{rel}, with respect to the exact ground state energy E_0 :

$$\epsilon_{\text{rel}} = \left| \frac{E_{\text{VMC}} - E_0}{E_0} \right|$$

We also define the hidden unit density $\alpha = M/N$, where M is the number of hidden units and N is the number of visible spins.

Our initial simulations immediately highlight the dependence of the final accuracy on the choice of hyperparameters and the optimization method.

As a first observation, illustrated in the figures below, we present the influence of the hidden unit count (M) and the learning rate (LR) on the averaged final relative error, using the **Gradient Descent** optimization approach.

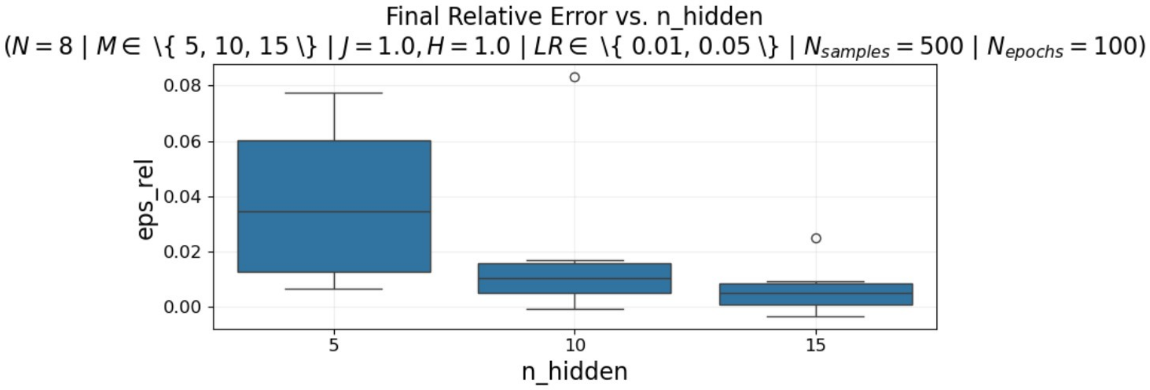


Figure 3 – Evolution of the mean final relative error ϵ_{rel} as a function of the number of hidden units M using Gradient Descent (GD).

This visualization reveals the coupled impact of these hyperparameters. Indeed, increasing the number of hidden neurons (M) generally **lowers the achievable error** floor by allowing the model to capture more complex correlations (Figure 3).

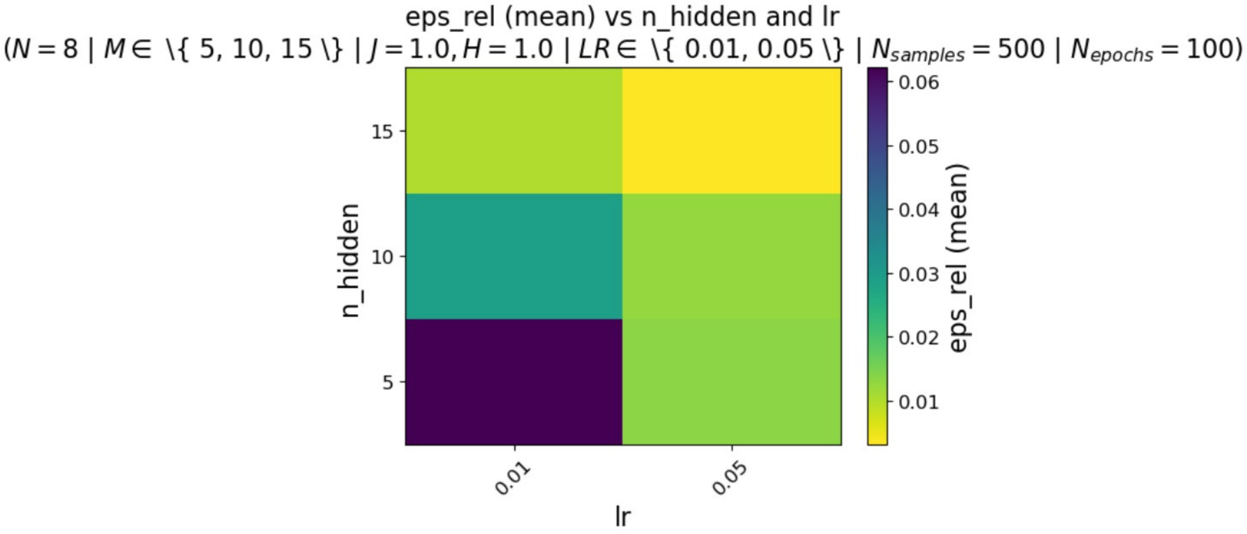


Figure 4 – Heatmap showing the mean final relative error ϵ_{rel} (averaged over repeats) as a function of the number of hidden units M and the learning rate LR using Gradient Descent (GD). The color intensity encodes the error magnitude.

Concerning the learning rate (LR), it remains a **deciding factor** for reaching this theoretical minimum (Figure 4). Indeed, an optimal LR range exists ; values outside this range lead either to slow convergence or to instability, limiting performance regardless of the network's capacity.

Subsequently, we performed a preliminary comparison of the two optimization models. As illustrated in Figure 5, the Stochastic Reconfiguration (SR) approach consistently outperforms standard Gradient Descent (GD) for the fixed parameters of our system. SR reaches the ground state with a significantly lower final relative error and requires fewer epochs for convergence.

Therefore, given the **interdependence** observed between the hyperparameters, we must systematically analyze each factor to clarify their individual contributions.

In the following sections, we isolate and study the impact of each hyperparameter alongside the optimization model (GD or SR) to deduce their specific roles in the model's accuracy.

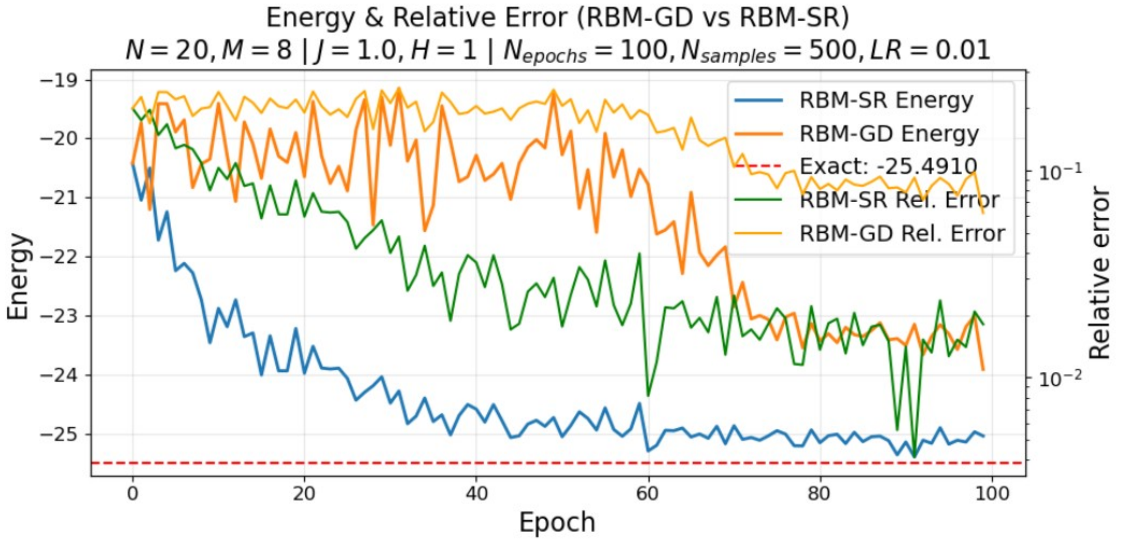


Figure 5 – Comparison of convergence for RBM optimized by Gradient Descent (GD) and Stochastic Reconfiguration (SR) for a system of $N = 20$ spins and $\alpha = 1$. SR converges faster and achieves a lower final relative error.

6.2 Influence of Hyperparameters on Convergence

We now investigate how key hyperparameters affect the convergence speed and the minimum achievable error, keeping the system size fixed at $N = 10$.

• Influence of Learning Rate (η)

The learning rate η controls **the magnitude** of the parameter update step $\Delta\mathcal{W}$. Let's first examine the direct sensitivity of the convergence accuracy to specific learning rate values. **Figure 6** illustrates the mean final relative error ϵ_{rel} for three distinct learning rates $(0.03, 0.05, 0.06)$.

A sharp disparity in performance is immediately visible depending on the optimization method. With **(GD)**, a higher learning rate of $LR = 0.06$ yields the fastest convergence among the tested values, demonstrating the method's robustness.
In stark contrast, for **(SR)**, the stability region is much narrower. While lower learning rates ($LR = 0.03$ and $LR = 0.05$) maintain a low error and stable convergence, a slight increase to $LR = 0.06$ results in a drastic degradation of accuracy. This sudden spike is a signature of **overshooting** : the gradient updates become too large for the Euclidean geometry, causing the optimization trajectory to oscillate wildly or diverge from the energy minimum rather than settling into it. (A)

Given this sensitivity, we extended our analysis to identify the effective stability window across different model sizes. **Figure 7** presents the relative error as a joint function of the learning rate (LR) and the number of hidden units (M).

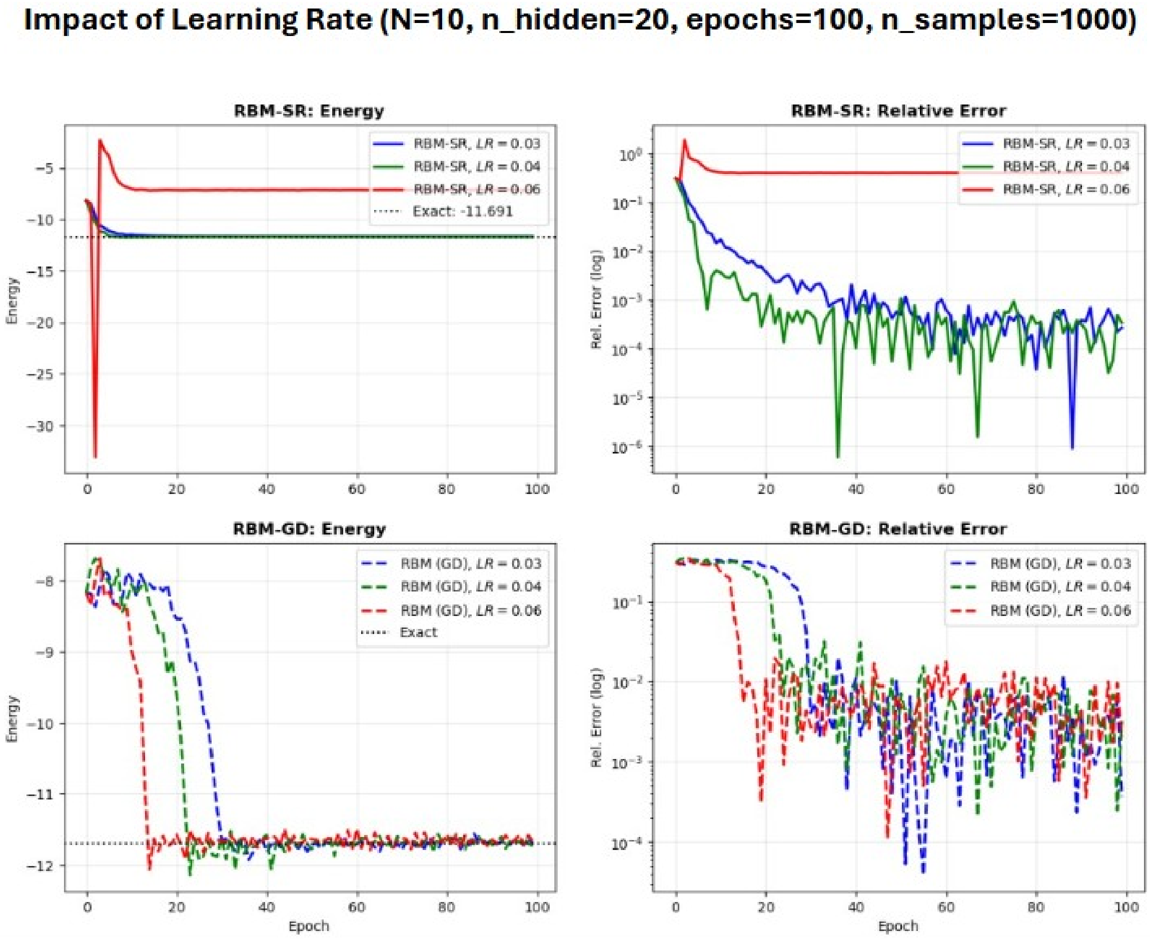


Figure 6 – Comparison of final relative error for specific learning rates. The drastic increase at $LR = 0.06$ highlights the instability threshold and the phenomenon of overshooting.

The figure 7 reveals an optimal window, approximately ($0.04 \leq LR \leq 0.05$) where the error is consistently minimized. For very small LR, convergence is too slow ; for large LR, the instability observed dominates.

In addition, in **figure 6**, for SR, we notice for the learning rate ($LR = 0.06$) that the energy after an **overshooting**, stabilizes, indicating convergence. However, the final energy value remains strictly higher than the exact ground state energy E_0. This gap suggests that the optimization process has been trapped in a **local minimum** of the variational energy landscape. Indeed, here the learning rate is small, for that the parameter updates are insufficient to escape this shallow basin of attraction.

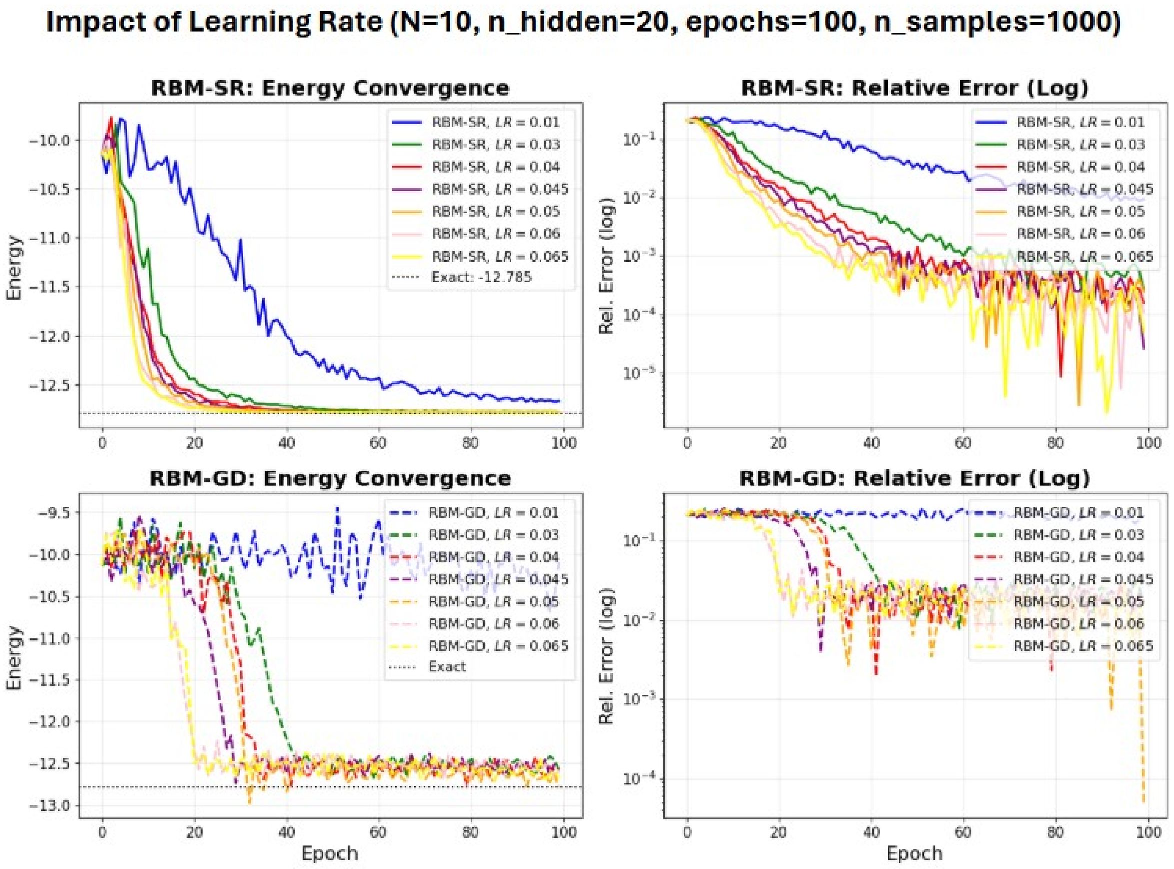


Figure 7 – Comparison of mean final relative error ϵ_{rel} for specific learning rates.

- ## Influence of Sample Size (N_{samples})

The accuracy of the Variational Monte Carlo (VMC) method fundamentally relies on the estimation of quantum expectations via Metropolis-Hastings sampling. The number of samples per epoch, N_{samples}, directly controls the **statistical noise (variance)** of the gradient estimate.

Indeed, for **low** N_{samples} The gradient estimate is noisy, leading to erratic updates. The optimization trajectory fluctuates significantly and may fail to settle into the precise ground state minimum. However, for **High** N_{samples} **:** the variance decreases as $1/\sqrt{N_{\text{samples}}}$, providing a cleaner gradient signal. This allows for finer adjustments to the parameters and a lower final relative error.

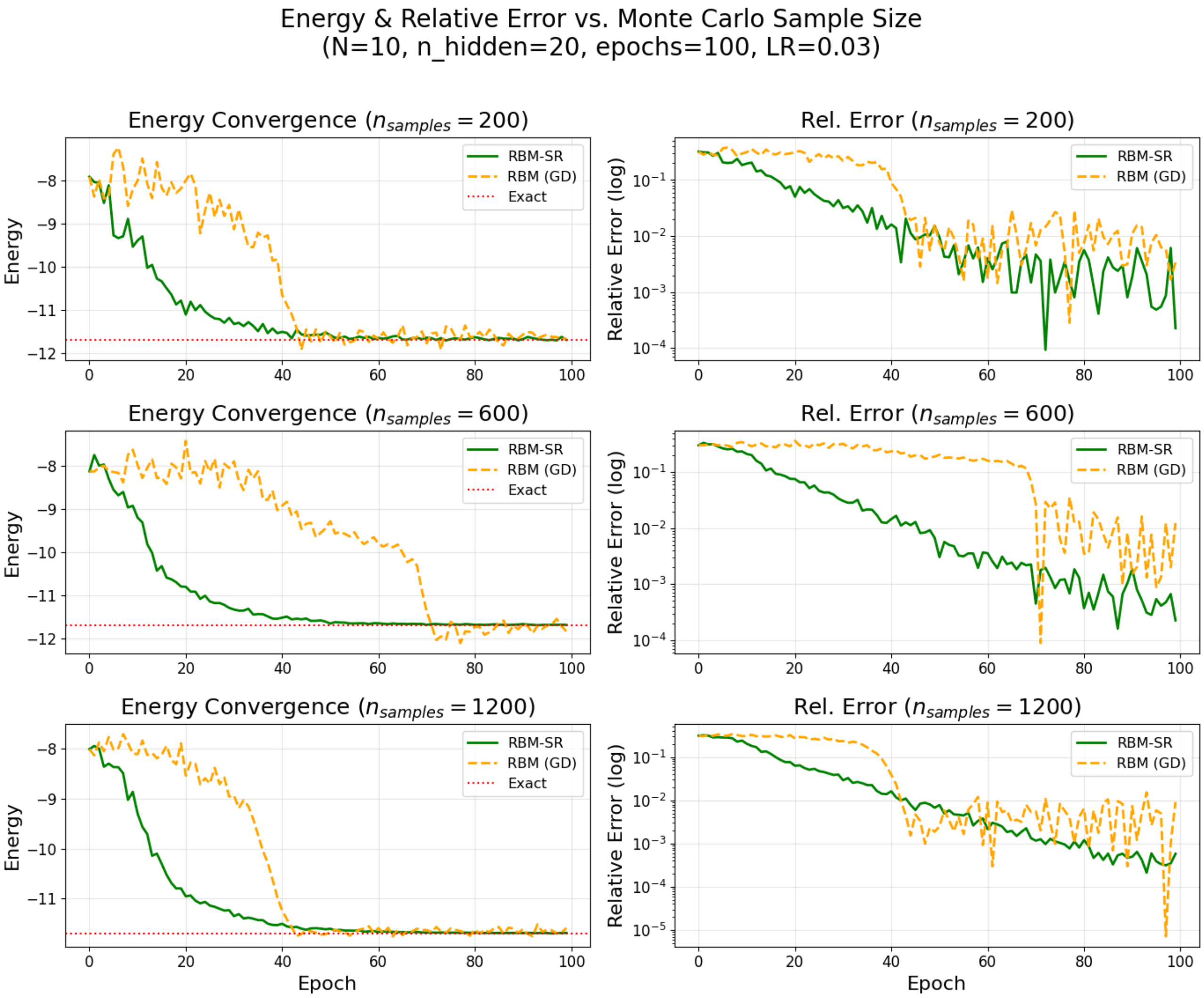


Figure 8 – Impact of sample size (N_{samples}) on the final relative error for GD and SR. While increasing samples reduces noise for both methods, SR maintains superior performance and stability even in high-noise (low sample) regimes.

Now, if we compare the performance of **Gradient Descent (GD)** and **Stochastic Reconfiguration (SR)** across varying sample sizes (**Figure 8**) : SR demonstrates remarkable **robustness** to sampling noise. Even at lower sample counts ($N = 600$), the curvature matrix S (acting as a preconditioner) helps filter out some of the noisy components of the gradient, maintaining a relatively smooth convergence trajectory. (A)

However, we can see that GD is more **sensitive** to sampling noise. With few samples, the "blind" Euclidean updates are easily misled by statistical fluctuations, resulting in a higher error floor and more unstable convergence compared to SR.

Therefore, While increasing N_{samples} universally improves accuracy by suppressing noise, SR's geometric correction offers a distinct advantage in computationally constrained scenarios where sampling is expensive.

- ## Impact of the Hidden Unit Density (α) on RBM Expressibility

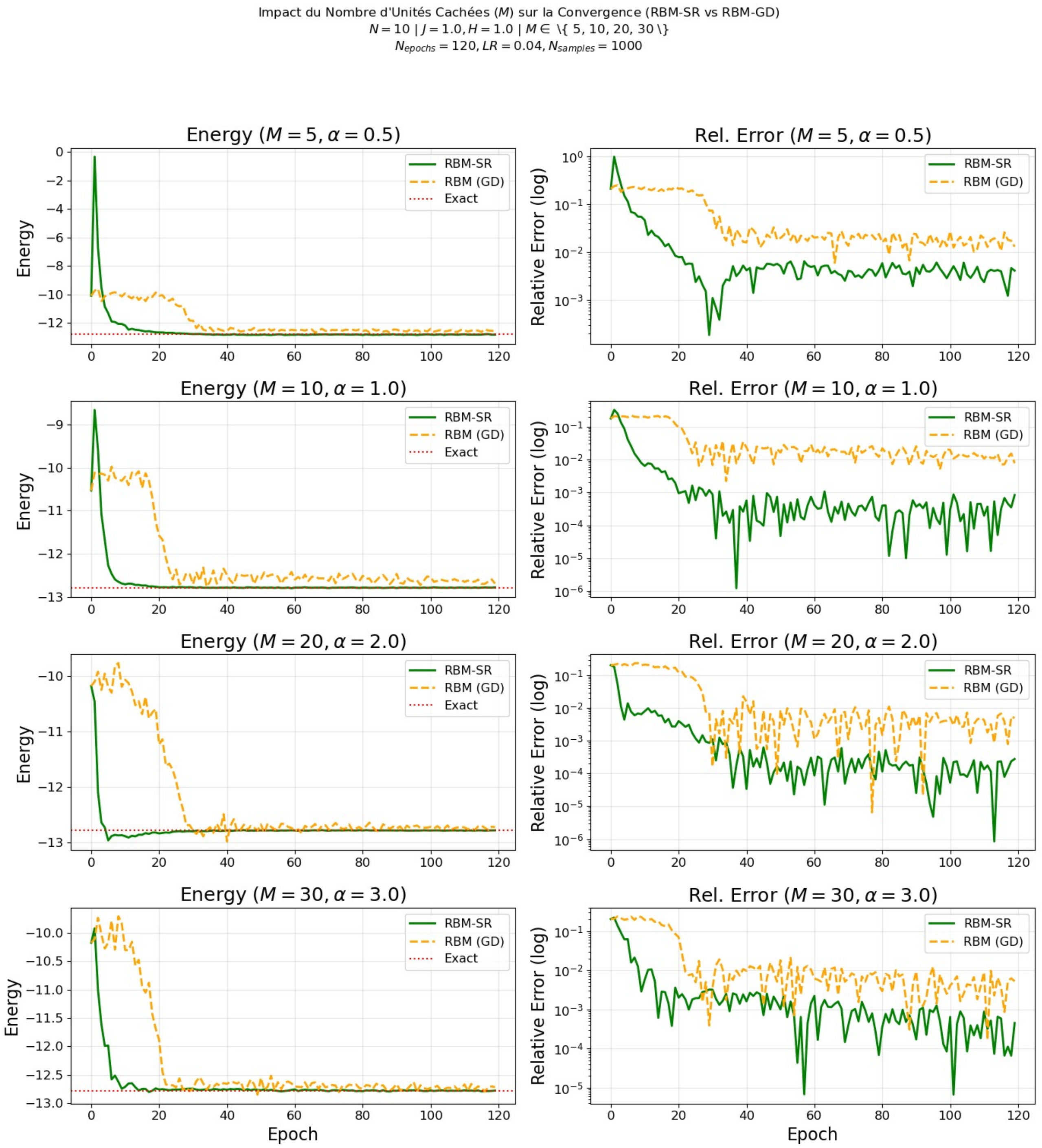
The expressibility of the RBM ansatz is controlled by the ratio $\alpha = M/N$, where M is the number of hidden units. We expect that an increase in α allows the network to represent more complex, entangled quantum states, leading to improved accuracy. Our results, summarized in Figure **9**, confirm this hypothesis. As α increases from 1 to 4, the final relative error decreases systematically. This dependence demonstrates that the RBM acts as a **universal approximator** for quantum states : a larger capacity (higher α) systematically lowers the energy expectation value towards the exact ground state E_0.

The representational power of the RBM ansatz is governed by the number of hidden units M, or more specifically, the density $\alpha = M/N$. For a fixed system size of $N = 10$ visible spins, we analyzed the evolution of the final relative error as a function of α (i.e., varying M).

Based on **figure 9** increasing α allows the RBM to encode higher-order correlations between the visible spins :

For **low Density ($\alpha < 1$)** The model is under-parameterized. Regardless of the optimization method, the RBM lacks the sufficient degrees of freedom to accurately approximate the complex ground state wavefunction, resulting in a high error floor. However, for a **high Density ($\alpha \geq 1$)**, the theoretical minimum error decreases but of course realizing this potential depends critically on the optimization capability.

In fact, the **figure 9**, contrasts the performance of **Gradient Descent (GD)** and **Stochastic Reconfiguration (SR)** as the model size grows : GD shows limited improvement with increasing α. While the error decreases slightly, it quickly saturates. This suggests that even if the larger network *can* represent a better state, standard GD struggles to navigate the higher-dimensional parameter space to find it. But, SR effectively leverages the additional capacity. We observe a sharp drop in relative error starting from $\alpha \approx 1$ ($M = 10$). Unlike GD, SR continues to improve significantly as α increases to 2 or 4, often reaching errors orders of magnitude lower than GD.



Impact du Nombre d'Unités Cachées (M) sur la Convergence (RBM-SR vs RBM-GD)
$N = 10 \mid J = 1.0, H = 1.0 \mid M \in$ \{ 5, 10, 20, 30 \}
$N_{epochs} = 120, LR = 0.04, N_{samples} = 1000$

Energy ($M = 5, \alpha = 0.5$)

Rel. Error ($M = 5, \alpha = 0.5$)

Energy ($M = 10, \alpha = 1.0$)

Rel. Error ($M = 10, \alpha = 1.0$)

Energy ($M = 20, \alpha = 2.0$)

Rel. Error ($M = 20, \alpha = 2.0$)

Energy ($M = 30, \alpha = 3.0$)

Rel. Error ($M = 30, \alpha = 3.0$)

Figure 9 – Final relative error vs. hidden unit density $\alpha = M/N$ for a system of size $N = 10$. SR (green) fully exploits the increased capacity, showing a significant performance jump around $\alpha = 1$, whereas GD (red) saturates early.

However, we have to make a critical observation from our hyperparameter analysis is the interdependence between the network capacity (M) and the required sample size (N_{samples}).

Figure 10 compares the convergence trajectories for varying hidden unit counts, where the sample size increases with the network capacity.

In fact, as we increase the number of hidden units M, the dimensionality of the parameter space grows. If N_{samples} is kept constant, the variance of the gradient estimator increases, introducing significant stochastic noise that can destabilize the optimization (as seen in the erratic oscillations for $M = 30$ with fixed low samples in **figure 9**).

Therefore, to maintain a constant signal-to-noise ratio in the gradient updates, it is necessary to increase N_{samples} alongside M. Our results show that doubling the samples when doubling the hidden units (e.g., $M = 10 \to 20$ with $N_{\text{samples}} = 1000 \to 2000$) preserves the stability of both algorithms. However, even with this adjustment, **SR remains the superior strategy.**

So, we find that for (SR) a density of $\alpha = 2$ $(M = 2N)$ combined with an adaptive sampling strategy yields the best trade-off between computational cost and accuracy.

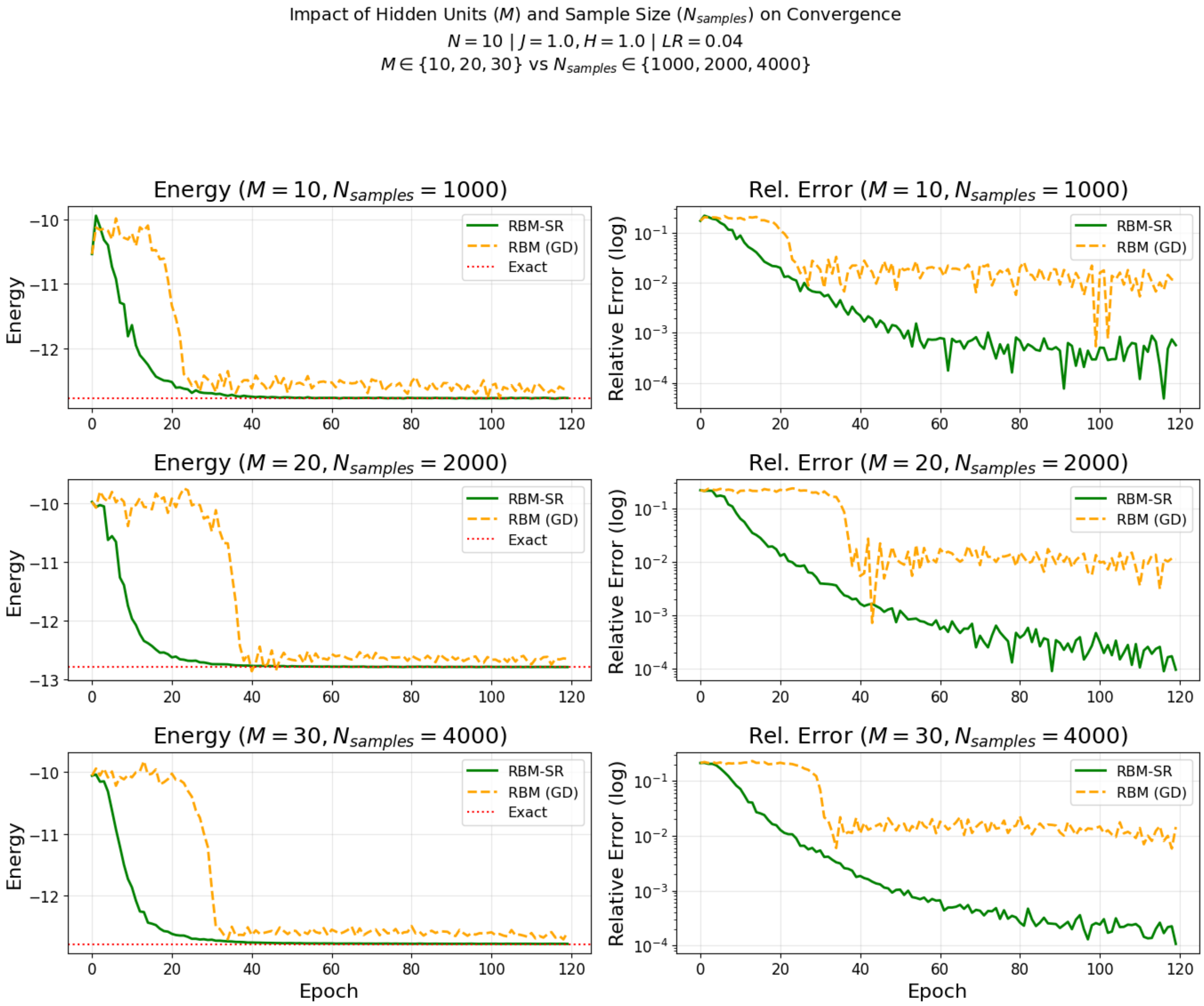


Figure 10 – Impact of scaling sample size with model complexity. Increasing N_{samples} concurrently with M stabilizes the learning of larger networks. The configuration with $\alpha = 2$ (middle row) demonstrates the optimal balance, achieving faster convergence than smaller networks while avoiding the noise instability of larger ones.

• Conclusion

Our comprehensive investigation into the RBM hyperparameters for a fixed system size ($N = 10$) reveals a clear hierarchy of constraints governing the optimization :

1. **Stability** is primarily dictated by the *learning rate*, which must be confined to an optimal window to avoid overshooting.
2. **Precision** is controlled by the *sample size* (N_{samples}), where higher counts mitigate gradient variance.

3. **Potential Accuracy** is determined by the *hidden unit density* (α), determining the model's capacity to capture complex correlations.

Crucially, our comparison exposes standard Gradient Descent (GD) as a performance bottleneck. GD fails to fully exploit the expressivity of dense networks ($\alpha \geq 1$) and remains highly sensitive to sampling noise. In stark contrast, **Stochastic Reconfiguration (SR)** consistently outperforms GD.

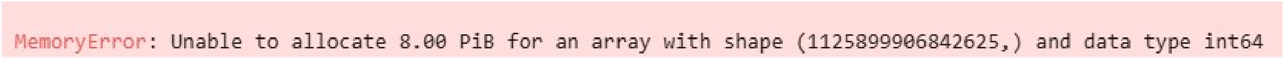
Given these distinct advantages, we conclude that geometry-aware optimization is indispensable for high-precision Quantum Monte Carlo. Consequently, for the remainder of this study will exclusively utilize the **Stochastic Reconfiguration (SR)** method.

6.3 Validation on Large Systems : The V-score Approach

For our previous analysis ($N = 10$), we computed the reference ground state energy E_0 by explicitly constructing the Hamiltonian matrix and performing dense diagonalization. However, the dimension of the Hilbert space grows as 2^N.

— For $N = 10$, the dimension is $2^{10} = 1024$, which is computationally trivial.
— For $N = 20$, the dimension reaches $\approx 10^6$, which is manageable.
— For $N = 50$, the dimension explodes to $2^{50} \approx 1.1 \times 10^{15}$.

Attempting to construct the dense Hamiltonian matrix for $N = 50$ would require allocating a contiguous memory block for over 10^{15} elements, corresponding to approximately **8 Petabytes (PiB)** of RAM. This represents an insurmountable "exponential wall" for standard numerical linear algebra. (**Figure 11**)



```
MemoryError: Unable to allocate 8.00 PiB for an array with shape (1125899906842625,) and data type int64
```

Figure 11 – insurmountable exponential wall.

To rigorously validate our results on large systems where exact diagonalization is computationally intractable, we adopted the benchmarking methodology recently established in 2024 [5]. This validation relies on the fundamental quantum mechanical principle that the **energy variance** of a quantum state vanishes if and only if that state is an **exact eigenstate** of the Hamiltonian :

$$\sigma_H^2 = \langle \Psi_\theta | \hat{H}^2 | \Psi_\theta \rangle - \langle \Psi_\theta | \hat{H} | \Psi_\theta \rangle^2 = 0 \quad \Longleftrightarrow \quad \hat{H} | \Psi_\theta \rangle = E | \Psi_\theta \rangle \tag{26}$$

In the context of Variational Monte Carlo (VMC), this variance is efficiently estimated "on the fly" using the **statistical fluctuations** of the local energies $E_{\text{loc}}(\mathbf{s})$ computed during the sampling phase :

$$\sigma_H^2 \approx \frac{1}{N_s}\sum_{k=1}^{N_s}(E_{\text{loc}}(\mathbf{s}_k))^2 - \left(\frac{1}{N_s}\sum_{k=1}^{N_s}E_{\text{loc}}(\mathbf{s}_k)\right)^2 \tag{27}$$

However, comparing raw variance across different system sizes is misleading because the energy is an extensive quantity ($E \propto N$). To address this, the article [5] introduces a standardized metric, the **V-score** ($\mathcal{V}$), defined as :

$$\mathcal{V} = N\frac{\sigma_H^2}{\langle\hat{H}\rangle^2} \tag{28}$$

This specific normalization serves two critical purposes :

1. **Dimensionless Metric :** Dividing by the squared expectation value $\langle\hat{H}\rangle^2$ creates a relative metric independent of the energy units.
2. **Extensivity Correction :** Since σ_H^2 typically scales linearly with N for short-range correlated states, multiplying by N (and dividing by $E^2 \propto N^2$) makes the V-score an *intensive* quantity. This allows for a fair comparison of variational accuracy between systems of widely different sizes.

Consequently, a **vanishing V-score combined with minimized variational energy** provides a robust, self-contained proof that the Neural Quantum State has successfully converged to the ground state, independent of any external reference.

Results and Interpretation

Figure 12 presents the training dynamics for a large Ising chain of $N = 50$ spins. The left panel displays the monotonic minimization of the variational energy density E/N, indicating effective optimization by the SR algorithm even in high-dimensional parameter spaces.

Crucially, the right panel confirms the physical validity of these results through the V-score analysis. **Starting from a highly fluctuating random state with a variance per spin of 9.62×10^{-1}, the network achieves a drastic reduction to 3.95×10^{-2} after only 50 epochs.** This decay by nearly two orders of magnitude is significant : according to the benchmarking criteria established by [5], such a rapid minimization of the V-score provides strong evidence that the Neural Quantum State is effectively approximating the true ground state with increasing fidelity. This successful convergence for $N = 50$ (where the Hilbert space dimension exceeds 10^{15}) demonstrates that the RBM ansatz scales efficiently, successfully overcoming the "exponential wall" encountered by exact diagonalization methods.

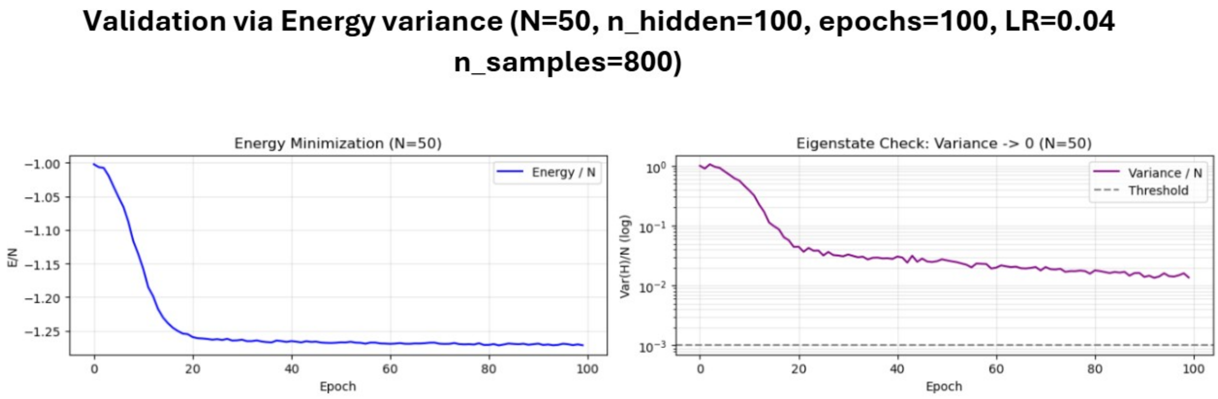


Figure 12 – **Variational optimization dynamics for a large-scale Ising chain ($N = 50$).** **Left :** Evolution of the variational energy density E/N during training. The monotonic decrease indicates the effective minimization of the Hamiltonian expectation value by the SR algorithm. **Right :** Evolution of the energy variance per spin (V-score)

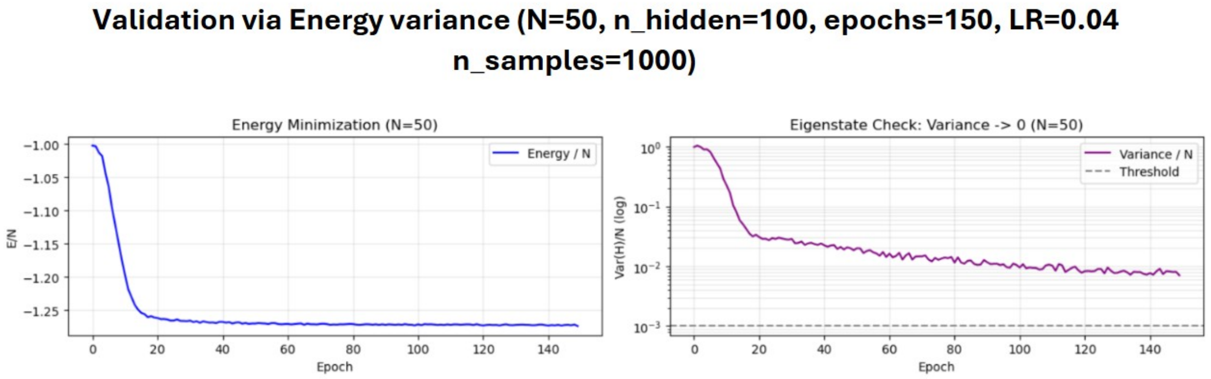


Figure 13 – **High-precision optimization for $N = 50$ (Epochs $= 150, N_{samples} = 1000$).** Compared to the standard run in Figure 12, the extended training duration and increased sampling rate lead to a further reduction in the V-score.

Figure 13 illustrates the impact of increased computational resources on the convergence quality for the same $N = 50$ system. Here, the training duration was extended to **150 epochs**, and the Monte Carlo sampling size was increased to $N_{samples} = 1000$ to reduce stochastic noise.

Comparing this to the previous result (Figure **12**), we observe a significant enhancement in the final state fidelity. The energy variance per spin drops further to 7.69×10^{-3}, representing a factor of **5.1** improvement over the standard run.

This confirms that the residual variance observed previously was partly due to statistical noise in the gradient estimation. By providing clearer gradient signals (higher $N_{samples}$) and allowing more optimization steps, the SR algorithm can fine-tune the RBM parameters to reach a V-score closer to the theoretical zero, pushing the boundaries of variational precision.

7
RESULTS AND DISCUSSION

The results obtained throughout this project demonstrate the efficacy of **Neural Quantum States (NQS)**, specifically **Restricted Boltzmann Machines (RBM)**, in solving the quantum many-body problem. Here, we discuss the critical factors influencing performance and the implications of our large-scale validation :

Optimization Landscape

Our comparative analysis highlights a fundamental disparity between Standard Gradient Descent (GD) and Stochastic Reconfiguration (SR). While GD often suffers from learning plateaus and slow convergence due to the complex geometry of the variational manifold, SR effectively acts as a *Natural Gradient* method. By using the covariance matrix S to correct for the curvature of the parameter space, SR navigates directly towards the global minimum.

For the critical transverse-field Ising model ($J = H = 1$), SR achieved a relative energy precision of the order of 10^{-4}, outperforming GD by several orders of magnitude. **This confirms that for quantum wavefunctions, accounting for the geometry of the parameter space is not optional, but essential.**

Hyperparameter Analysis and Stability

A significant portion of our work focused on characterizing the influence of hyperparameters on convergence stability.

- **Hidden Unit Density ($\alpha = M/N$) :** We observed a **monotonic improvement** in variational accuracy as the hidden unit density increases. Higher values of α (e.g., $\alpha > 3$) endow the network with greater representational power, allowing it to capture intricate entanglement structures and converge closer to the exact ground state energy, provided the optimization is stabilized.
- **Adaptive Sampling Strategy :** Our results reveal a strong correlation between network complexity and gradient noise. As we increase the number of hidden units M to gain accuracy, the variance of the gradient estimator increases. To maintain a constant

signal-to-noise ratio and prevent divergence, it is imperative to increase the number of Monte Carlo samples ($N_{samples}$) proportionally to M.

— **Learning Rate (LR) :** Our analysis reveals that Stochastic Reconfiguration is highly sensitive to the learning rate magnitude. While SR provides a superior optimization *direction* (by correcting for curvature), a learning rate that is too aggressive leads to immediate instability. We can therefore identify an optimal operating window.

Overcoming the Exponential Wall

The important part of this work is the validation of the RBM ansatz on macroscopic systems ($N = 50$), where the Hilbert space dimension ($2^{50} \approx 10^{15}$) renders exact diagonalization impossible (the "exponential wall").

By adopting the benchmarking protocol established by [5], we utilized the **V-score** (normalized energy variance) as an intrinsic quality metric. The exponential decay of the V-score confirms that NQS can efficiently encode the ground state of large many-body systems with polynomial scaling in complexity.

8
CONCLUSION

In this project, we successfully implemented a **Restricted Boltzmann Machine (RBM)** optimized via **Variational Monte Carlo (VMC)** to solve the 1D Transverse-Field Ising Model. Our work bridges the gap between machine learning architectures and computational quantum physics.

Our numerical experiments yielded several decisive insights. Regarding **algorithmic robustness**, we established that the Stochastic Reconfiguration (SR) algorithm is strictly superior to standard gradient descent for minimizing the variational energy, as it effectively corrects for the curvature of the optimization landscape. Crucially, our analysis of **hyperparameters** revealed the necessity of an adaptive sampling strategy, where the number of Monte Carlo samples must scale with network complexity to mitigate stochastic noise.

Furthermore, we demonstrated the **scalability** of the RBM ansatz, which scales efficiently from small prototypes ($N = 10$) to large systems ($N = 50$) without degrading its ability to capture long-range correlations.

Finally, we achieved **validation without reference** by successfully applying the V-score metric [5] to certify the quality of our solution in the absence of exact solutions, marking a critical step for modern quantum simulation where ground truth is often inaccessible.

Looking forward, this framework paves the way for studying more complex phases of matter, such as frustrated spins in 2D or fermionic systems. Neural Quantum States represent a paradigm shift in computational physics, offering a flexible and powerful tool to explore the frontiers of the quantum world.

A
STOCHASTIC RECONFIGURATION VS GRADIENT DESCENT

Geometry : The Curvature Advantage

The superior performance of Stochastic Reconfiguration (SR) observed in our results originates from the fundamentally different way in which parameter updates $\Delta\theta$ are defined with respect to the geometry of the loss landscape. While standard Gradient Descent (GD) implicitly assumes a flat (Euclidean) parameter space, SR explicitly incorporates the local curvature of the variational manifold.

To build intuition, consider a "narrow valley" loss landscape (Fig. 14), characterized by a strong anisotropy : a high curvature in the transverse direction (steep valley walls) and a low curvature in the longitudinal direction (flat valley floor).

- **Standard Gradient Descent (Euclidean Gradient).** The GD update rule is

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}, \tag{29}$$

where the step size in each direction is directly proportional to the local slope.

  - In **steep directions** (high curvature), the gradient $\nabla_\theta \mathcal{L}$ becomes large. As a consequence, GD takes excessively large steps, overshoots the valley minimum, and produces the oscillatory zig-zag trajectory shown in red in Fig. 14.
  - In **flat directions** (low curvature), the gradient is small, leading to very slow progress along the valley floor andison.

  Overall, GD fails to adapt its step size to the anisotropy of the landscape, resulting in inefficient convergence.

- **Stochastic Reconfiguration (Natural Gradient).** SR modifies the update rule by introducing the inverse covariance matrix S^{-1} :

$$\theta_{t+1} = \theta_t - \eta\, S^{-1} \nabla_\theta \mathcal{L}. \tag{30}$$

The matrix S is the Fisher information matrix associated with the variational wave function and is given by

$$S_{ij} = \langle O_i O_j \rangle - \langle O_i \rangle \langle O_j \rangle, \qquad O_i = \partial_{\theta_i} \ln \psi(\mathbf{x};\theta). \tag{31}$$

This matrix encodes the local curvature of the variational manifold : directions in parameter space along which the wave function is highly sensitive correspond to large eigenvalues of S, whereas weakly sensitive directions are associated with small eigenvalues.

— **Damping effect (steep directions).** In directions of large curvature, the gradient magnitude is large but is strongly suppressed by multiplication with S^{-1}. This damping prevents overshooting and stabilizes the optimization.

— **Boosting effect (flat directions).** In directions of small curvature, the gradient is weak but is amplified by S^{-1}, significantly accelerating the motion along the valley floor.

From a geometric perspective, SR performs gradient descent in the curved space of variational parameters endowed with the Fisher metric. By acting as a preconditioner, S^{-1} effectively corrects the anisotropy of the loss landscape, transforming a narrow valley into an approximately isotropic one from the optimizer's point of view. This allows SR to follow a direct and stable trajectory (green curve) toward the minimum, as illustrated in Fig. 14.

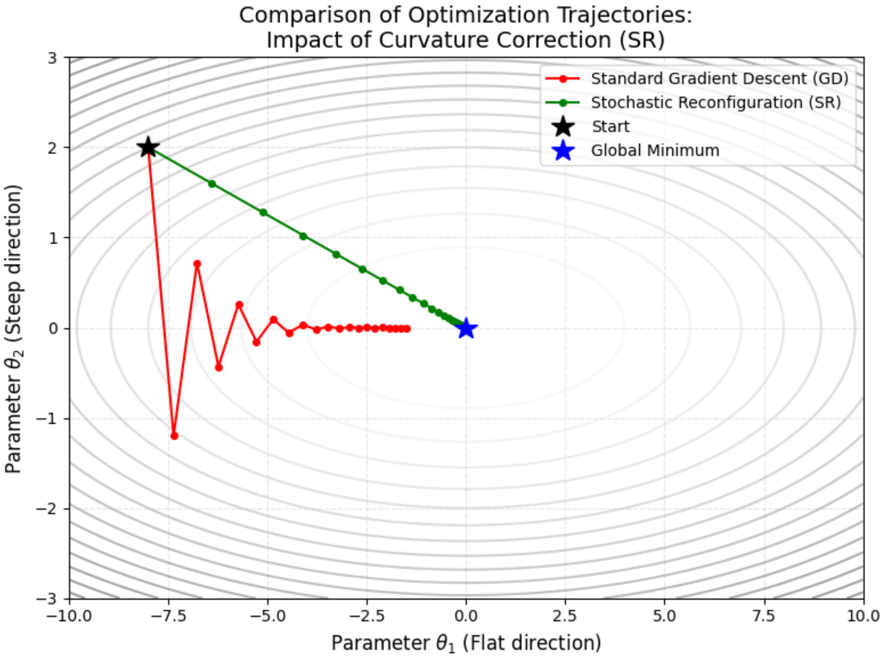


Figure 14 – Visual comparison of optimization trajectories. Standard Gradient Descent (red) exhibits inefficient oscillations due to strong curvature anisotropy, while Stochastic Reconfiguration (green) corrects for curvature and converges directly toward the minimum.

RÉFÉRENCES

[1] G. Carleo and M. Troyer, *Solving the quantum many-body problem with artificial neural networks*, Science **355**, 602–606 (2017). DOI : 10.1126/science.aag2302

[2] A. Decelle and C. Furtlehner, *Restricted Boltzmann Machine, recent advances and mean-field theory*, Departamento de Física Teórica I, Universidad Complutense (Madrid) and TAU team INRIA Saclay & LISN, Université Paris-Saclay (2017).

[3] L. Gay, *Modèle d'Ising*, Unité de mathématiques pures et appliquées, Lyon, Supervised by M. Simon, (2019).

[4] `https://www.attaccalite.com/PhDThesis/html/node15.html`

[5] D. Wu, R. Rossi, F. Vicentini, et al., *Variational benchmarks for quantum many-body problems*, Science, Vol. 386, Issue 6719, pp. 296–301, (2024).