Performance Evaluation: Neural Network Quantum States vs. Variational Quantum Algorithms

Navaneeth Krishnan Mohan

March 2024

1 Introduction

In this study, we aim to compare the performance of Neural Network Quantum States (NQS) using the Restricted Boltzmann Machine (RBM) as the variational ansatz, with Hamiltonian Variational Ansatz (HVA) in finding the ground state of the 1D Transverse Field Ising Model (TFIM). TFIM serves as an ideal testbed for quantum many-body simulations.

Neural Network Quantum States (NQS) use an Artificial Neural Network (ANN) as a variational ansatz, whose parameters need to be optimised to study the approximate wave function of quantum many-body systems. We have chosen Restricted Boltzmann Machine (RBM) as our NQS in this study. The effectiveness of RBM in determining ground states of spin models has been demonstrated in prior research such as [1], where simulations of 1D Transverse Field Ising (TFI) models with up to 80 spins exhibited high accuracy. RBM, a specific type of ANN often utilised in unsupervised learning tasks, can be understood from a physics perspective as a finite temperature Ising model with a distinct connection between spins. Detailed insights into RBM functionality and its applications can be found in [2]. The link between RBM and the Ising model is elaborated in [3], underscoring the relevance of RBM in simulating spin systems.

Hamiltonian Variational Ansatz belongs to the class of problem-specific ansatz, where we require the form of Hamiltonian to construct the ansatz. HVA takes inspiration from Quantum Approximate Optimization and adiabatic computation, which is believed to be mildly affected by the problem of barren plateaus. It has also been shown recently that by imposing certain constraints on parameters, the optimisation landscape can be made barren plateau-free [4]. The ability of HVA to find ground state and ground state properties is addressed in [5], where they have simulated TFIM and XXZ spin models up to 16 spins. They have also shown that increasing the parameters beyond a certain threshold (over-parameterisation) would make the optimisation landscape almost trap-free. Thus, the optimiser will not get stuck on local minima.

Both variational algorithms selected for the study show a close relation with the TFI model. Thus, a comparative study of both algorithms' performance helps us identify their strengths and weaknesses. In the study, we analyse the performance of finding the ground states using the relative error as a comparison metric. It is defined as $|(E_{ansatz} - E_{exact})/E_{exact}|$, where E_{ansatz} and E_{exact} represent energies from the variational ansatz and exact diagonalisation, respectively. We study the relation between relative error and the number of parameters to optimise across different TFIM configurations (different values of J and h). In the case of RBM, parameter count depends upon a quantity called *Alpha*, which is the ratio of the number of hidden neurons to visible neurons. For HVA, parameter count is dependent on the circuit depth. Thus, in this study, we analyse how accurately RBM and HVA can describe the ground state of different TFIM configurations for different alpha and circuit depth values. We have also validated and explained our results from the literature.

Before embarking on a comparative study of two variational algorithms, optimising various hyperparameters that significantly impact simulations is crucial. In the preliminary investigation, we focused on several key aspects. Firstly, we analysed the scaling of the number of parameters to optimise with Alpha and depth and how parameter count correlates with system size. Additionally, we explored the influence of iteration counts of optimisation algorithms on result accuracy. During this preliminary study, alongside examining the Hamiltonian Variational Ansatz (HVA), we also evaluated the performance of the heuristic trial wavefunction known as RealAmplitudes. The detailed outcomes of these trials are provided in the Appendix section, shedding light on the behaviour and efficacy of different optimisation strategies and trial wavefunctions in the context of ground state calculations for quantum many-body systems.

2 A note on Transverse Field Ising Model

The transverse field Ising model is one of the simplest quantum spin systems that can be studied, and it has now become a testbed for testing various quantum many-body simulation algorithms. In this study, we focus our attention on 1D TFIM, which is defined as;

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x$$

where J accounts for the spin-spin interaction strength. The external field influences each spin in the chain in the x direction, whose strength is indicated by h.

- If $h = 0$, it indicates no field, and h taking negative values implies the field is in the opposite direction.
- If $|h| < |J|$ the spins show an ordered phase, the ordering depends upon J value.
    - If $J > 0$ spins have ferromagnetic ordering and $J < 0$ indicate antiferromagnetic ordering.
    - Ferromagnetic ordering implies adjacent spins aligning in the same direction, and antiferromagnetic ordering implies adjacent spins aligning in opposite directions.
- If $|h| > |J|$, it indicates a disordered phase. All the spins will align in the direction of the x axis, or as in quantum computing language, all spins will be in the plus state.

3 Performance analysis of RBM and HVA

In our preliminary study, documented in Appendix 1, we studied three different TFIM models with spins equal to 10,15 and 18. Firstly, we studied how the parameters optimise scales with the system size and compared the execution time for finding the ground state. We found that parameter count in RBM depends on the system size and scales linearly with size, whereas HVA is independent of system size. Then, we studied how the parameter counts scaled with the Alpha values and circuit depths of RBM and HVA, respectively. We observed a linear scaling of parameters with both the Alpha and depth. Finally, we studied the influence of the optimiser's iteration count on the solutions' accuracy. Interestingly, we observed that the iteration counts minimally impacted result accuracy.

We utilised Netket to implement RBM and Qiskit to implement HVA. We follow the construction of the HVA ansatz for TFIM provided in ref. [5], and RBM is already available as a pre-built model in Netket. For parameter optimisation, we employed a gradient-based Stochastic Gradient Descent (SGD) algorithm for RBM and a gradient-free COBYLA algorithm for HVA, as these are the default optimisers in Netket and Qiskit, respectively.

In this section, we will compare the performance of RBM and HVA. The metric used for comparison is the relative error in predicted ground state energy and the exact ground state energy. We have tested RBM and HVA for different TFIMs with different values of h and J. We also investigated how varying Alpha and depth influence simulation accuracy across all TFIM configurations. Using the insights gained from the preliminary study, considering the execution time and resources required, we have chosen to fix the number of spins at N = 15 for future studies. We have also fixed the number of iterations to 300 since we found that for 15 spins, it would provide accurate solutions within less time. During our preliminary study, we observed the influence of the initial parameters chosen on the results, particularly in the case of HVA. We have also fixed the initial parameters throughout the simulation to remove this dependency.

3.1 Varying Magnetic Field Strength ("h" value).

In this trial, we are fixing the J value to 1 and varying the value of h, where $h \in \{0.1, 0.5, 1, 2\}$. Thus, we are considering four different TFIM configurations in this study. In the case of RBM, the alpha value varied from 1 to 4, and for HVA circuit depth considered are $\{1, 2, 3, 4, 5, 6, 8, 10\}$. The data of the trials conducted is given in Appendix 2.

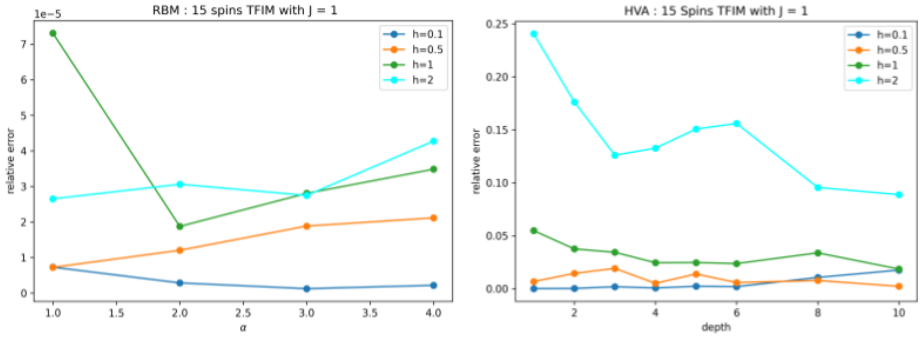

Figure 1: (left) Relative error against Alpha values for TFIM configurations with different h values. (right) Relative error against circuit depths for TFIM configurations with different h values.

Observations: RBMs exhibit strong performance in determining ground states for Transverse Field Ising (TFI) models, achieving a mean error on the order of 10^{-5}. Across all models examined, lower values of $alpha$ are highly effective in describing the ground state. In contrast, HVA achieves a minimum relative error of approximately 10^{-2} in our trials. The relative error increases significantly to around 10^{-1} for $h = 2$, indicating a trend of error escalation with higher h values. For lower h values, employing an ansatz depth of 1 yields high accuracy; however, as h increases, greater depth becomes advantageous for achieving better solutions. Balancing execution time and accuracy, a depth of 4 emerges as a favourable choice, delivering notably good solutions with reduced execution time.

3.2 Varying Exchange Coupling Strength ("J" value)

We are varying the J value in this trial while fixing the h value. The J values considered are $\{0.1, 0.5, 1\}$. The results are given in the figure 2. In the case of RBM, the alpha value varied from 1 to 4, and for HVA circuit depth considered are $\{1, 2, 3, 4, 5, 6, 8, 10\}$. The full results of the simulation are given in Appendix 2.

Observations: RBMs excel with a mean relative error of approximately 10^{-5}. Surprisingly, even with an alpha value of 1, RBM consistently delivers good solutions with an error in the same order. However, HVA struggles in comparison, achieving only a mean error of around 10^{-1}. Notably, HVA encounters significant challenges with disordered Transverse Field Ising Models (TFIM), i.e., when the magnitude of the external magnetic field ($|h|$) exceeds that of the interaction strength ($|J|$), resulting in errors of approximately 10^{-1}. It can also be inferred from figure 1 that there is a sudden spike in error for the case $|h| = 2$. Furthermore, across various tested models, a depth of 4 consistently yields optimal solutions, indicating its robustness and efficacy in these contexts.

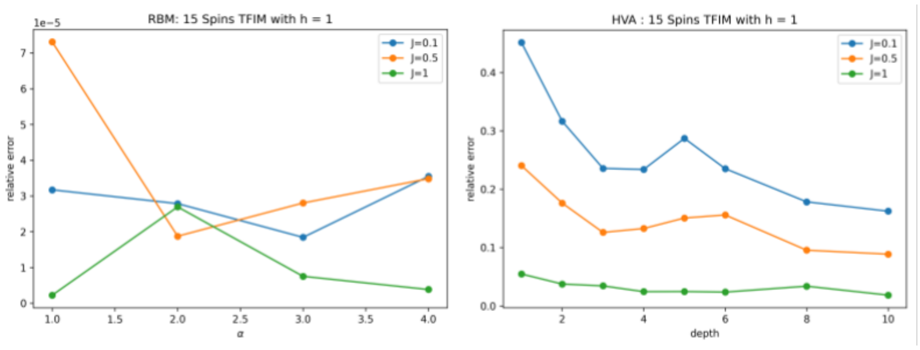

Figure 2: (left) Relative error against Alpha values for TFIM configurations with different J values. (right) Relative error against Circuit depths for TFIM configurations with different J values.

The curious case of negative J values: While conducting trials, we observed that when J takes negative values, the accuracy of simulation declines for both RBM and HVA. The simulation results are given in the figure 3. Changing the J value to negative physically means we are making the model antiferromagnetic instead of ferromagnetic interaction. We cannot explain the sudden decline or find evidence supporting this observation. We believe this inconsistency may go away if we use better optimisers.

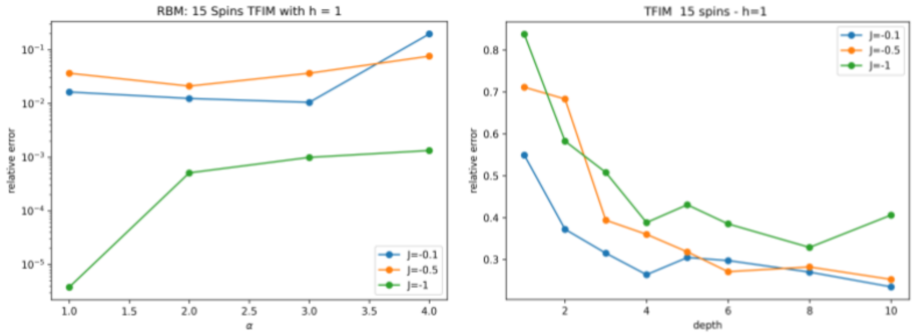

Figure 3: (left) Relative error against Alpha values for TFIM configurations with different $-J$ values. (right) Relative error against Circuit depths for TFIM configurations with different $-J$ values.

Inferences:

- From the trials conducted, we can see that RBM performs better than HVA with high accuracy in finding the ground state.
- In terms of the execution time and number of parameters to optimise, HVA have very few parameters and thus takes less time to converge to a solution compared to RBM.
- Even at higher circuit depths, HVA fails to achieve the accuracy level of RBM. In contrast, RBM performs very well and achieves an error of the order of 10^{-5} even for alpha = 1.
- HVA struggles with disordered TFI models and has a relative error of order 10^{-1}.
- Both RBM and HVA experience a decrease in accuracy when simulating the antiferromagnetic TFI model. Compared to the ferromagnetic model, RBM shows a notable increase of 10^3 in error for antiferromagnetic models. Similarly, HVA exhibits a higher error rate on the order of 10^{-1} when simulating antiferromagnetic TFI configurations. It's worth noting that the minimum error achieved by HVA for the ferromagnetic model is of the order of 10^{-2}. Therefore, while RBM's performance deteriorates more significantly than HVA when simulating

antiferromagnetic TFI models, it still provides more accurate solutions overall.

4 Discussion and Conclusion

In our trials, the performance of Restricted Boltzmann Machines (RBM) surpassed that of Hamiltonian Variational Ansatz (HVA), exhibiting superior convergence. We evaluated their performance by determining the ground state of a Transverse Field Ising (TFI) model with 15 spins arranged in a chain.RBM consistently achieved high accuracy even with low alpha values. However, we did not observe any improvement in results with high alpha values, although literature, such as [1], indicates that this may not always be true. Their study found the ground state of TFIM with 80 spins on a chain and demonstrated how accuracy consistently improves with increasing alpha values. Therefore, while small alpha values may suffice for smaller or moderate system sizes, larger Alpha values may be necessary for higher system sizes to achieve optimal results.

When comparing the accuracy of both algorithms for the model studied, RBM exhibited approximately three times better accuracy than HVA. While HVA achieved a mean relative error on the order of 10^{-2}, RBM achieved an error of approximately 10^{-3}. Notably, HVA struggles notably with disordered Transverse Field Ising Models (TFIMs) where $|h| > |J|$, with relative error spiking up to the order of 10^{-1}. Furthermore, both algorithms encounter difficulties with antiferromagnetic TFIM models (negative J values). The sudden decline in accuracy when we simulate the antiferromagnetic TFI model for RBM and HVA remains unexplained, indicating a need for improved optimisers and initialisation strategies to address these inconsistencies effectively.

It's important to note that accuracy and convergence are influenced not only by the variational ansatz but also the choice of optimiser used for parameter optimisation. In the case of HVA, research referenced in [5] demonstrated that accuracy could be significantly improved by employing the ADAM optimiser with a learning rate within the range of $1\times10^{-3} \leq r \leq 4\times10^{-2}$ and a maximum iteration value of 15000. They specifically studied the Transverse Field Ising Model (TFIM) with $N = 16$ spins and used infidelity as a metric to assess convergence. For $J = 1$, they achieved an infidelity on the order of 10^{-13} for $|h| < 1.2$ and 10^{-6} for $|h| > 1.2$. This sudden spike in infidelity for $|h| > 1.2$ supports our observation that HVA struggles with disordered TFIMs.

We could not validate the results in [5] since the current version of Qiskit does not support Adam optimiser. Nevertheless, we conducted tests using RBM with an RBM optimiser, utilising a learning rate of 0.001 and a *maxiter* value of 1000. The results of these tests are presented in Table 1. Here, we can see the minimum infidelity RBM attain is of the order 10^{-7} for small values of h. For large values of h, infidelity jumps to the order of 10^{-1}. Thus, with Adam's Optimiser, HVA gives better results than RBM. However, it should be noted that all the results presented in [5] are performed in quantum simulators. Thus, while we scale up the TFIM size, it becomes unfeasible to do the simulation regarding computational resources.

J-Value	h-Value	alpha	relative-error	infidelity	exec-time
1	0.100000	1	7.0489414e-06	8.77954181e-07	30:10
1	1.000000	1	0.0007861586	0.00342110607016	37:11
1	2.000000	1	3.435093e-05	0.292878788659302	30:53

Table 1: RBM trails with Adam Optimizer with learning rate 0.001 and $maxiter = 1000$.

We could not validate the results in [5] since the current version of Qiskit does not support Adam optimiser. Nevertheless, we conducted tests using RBM with an RBM optimiser, utilising a learning rate of 0.001 and a *maxiter* value of 1000. The results of these tests are presented in Table 1. Here, we can see the minimum infidelity RBM attain is of the order 10^{-7} for small values of h. For large values of h, infidelity jumps to the order of 10^{-1}. Thus, with Adam's Optimiser, HVA gives better results than RBM. However, it should be noted that all the results presented in [5] are performed

in quantum simulators. Thus, while we scale up the TFIM size, it becomes unfeasible to do the simulation regarding computational resources.

When comparing both algorithms in terms of the number of parameters to optimise, we observed distinct scaling behaviours. In HVA, the parameter count is solely determined by the circuit depth and scales linearly with depth. On the other hand, in RBM, the parameter count scales linearly with both the alpha values and the system size. Even with an alpha value of 1, the parameter count in RBM is on the order of 10^2, whereas in HVA, it is given by $2N$, where N represents the circuit depth. This difference in parameter scaling results in significantly fewer parameters to optimise in HVA compared to RBM, thus resulting in shorter execution times, especially for moderate circuit depths. Moreover, in quantum backend simulations, execution time is more heavily influenced by circuit depth rather than the number of qubits. This further emphasises the advantage of HVA in terms of computational efficiency as system sizes increase.

Despite HVA's efficiency in terms of parameter counts, it falls short of RBM's accuracy level, making a direct comparison challenging. However, literature such as [5] showcases the phenomenon of a "computational phase transition" in over-parameterised HVAs, where the optimisation landscape becomes almost convex. This characteristic allows gradient-based optimisers like Stochastic Gradient Descent (SGD) to exploit convexity for faster convergence, suggesting that over-parameterisation can enhance convergence rates in certain instances. Over-parameterisation has been extensively studied in the context of deep neural networks as well [6].

For a specific scenario, such as the TFIM with $N = 10$ spins, the over-parameterisation threshold for HVA is identified as 20 parameters, corresponding to a circuit depth of 10 [5]. We also investigated over-parameterisation in HVA for the N=10 spin TFIM chain with $J = 1$ and $h = 1$. The results of the trial are detailed in the table 2. In our trials, we could see a consistent increase in accuracy with circuit depth. However, it's notable that despite this increase in accuracy, we did not observe a corresponding improvement in convergence rates during our experiments. This may be because we are using COBYLA optimiser, which is a gradient-free optimiser.

circuit-depth	num-parameters	num-of-iterations	relative-error	exec-time
1	2	28	0.065611	0:12
2	4	56	0.052389	0:31
3	6	71	0.065436	0:40
4	8	96	0.028364	0:55
5	10	103	0.047670	0:56
6	12	137	0.023688	1:18
8	16	146	0.045866	1:35
10	20	200	0.031622	2:19
Over parameterized regime				
12	24	277	0.042972	3:23
14	28	247	0.033862	3:14
16	32	331	0.028748	4:36
18	36	374	0.025586	5:27
20	40	456	0.018994	7:8

Table 2: 10 spins trail with HVA, extending into the overparameterised regime.

In conclusion, our trials showcased the superior performance of Restricted Boltzmann Machines (RBM) over Hamiltonian Variational Ansatz (HVA) in terms of accuracy and convergence. RBM and HVA show linear scaling of parameters to optimise with respect to alpha and circuit depth, respectively. While the parameter count of RBM depends on system size, HVA, on the other hand, shows independence from system size. Thus, even at moderate circuit depth, the convergence rate of HVA is faster than for the alpha=1 case of RBM. On the other hand, RBM shows high and consistent accuracy even at alpha=1, demonstrating its robustness. Moreover, the performance of HVA is inconsistent across all TFI models, with high errors for disordered systems. Further, RBM and HVA demonstrated a decline in accuracy while simulation antiferromagnetic models (-ve ”J” value) compared to ferromagnetic models (+ve ”J” value), which we were not able to explain.

We also want to address some improvements that could be made in our analysis. First, the hyperparameter optimisation should be extended to test various optimisers and find the best optimiser for RBM and HVA. Secondly, the HVA results heavily depend on the initial parameters chosen for the optimisation. Our analysis removed this dependence by fixing the seed for random initialisation. However, for a more elaborate analysis, the HVA must be tested for different initial parameter values, and each instance's convergence rate should be studied. In addition, the initial state of all the qubits in HVA is fixed to be in the $|+\rangle$ state. Thus, during optimisation, the algorithm must search the entire Hilbert space to find the ground state. We believe better initialisation strategies would lead to faster convergence and may solve the problem with disordered systems. Further, an analysis of the performance of HVA in the real quantum backend must also be done. By considering errors in computation, IBM open plans do not provide enough runtime for solutions to converge and require premium plans for such an analysis.

References

[1] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):pp. 602–606, 2017. ISSN 00368075, 10959203. URL `https://www.jstor.org/stable/24918355`.

[2] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1): 926, 2010.

[3] Bryan Clark. Computing in Physics — Computing in Physics (498CMP). URL `https://courses.engr.illinois.edu/phys498cmp/sp2022/Overview.html`.

[4] Chae-Yeun Park and Nathan Killoran. Hamiltonian variational ansatz without barren plateaus. *Quantum*, 8:1239, February 2024. ISSN 2521-327X. doi: 10.22331/q-2024-02-01-1239. URL `http://dx.doi.org/10.22331/q-2024-02-01-1239`.

[5] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. Exploring entanglement and optimization within the hamiltonian variational ansatz. *PRX Quantum*, 1(2), December 2020. ISSN 2691-3399. doi: 10.1103/prxquantum.1.020319. URL `http://dx.doi.org/10.1103/PRXQuantum.1.020319`.

[6] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers, 2020.

[7] Filippo Vicentini, Damian Hofmann, Attila Szabó, Dian Wu, Christopher Roth, Clemens Giuliani, Gabriel Pescia, Jannes Nys, Vladimir Vargas-Calderón, Nikita Astrakhantsev, and Giuseppe Carleo. NetKet 3: Machine Learning Toolbox for Many-Body Quantum Systems. *SciPost Phys. Codebases*, page 7, 2022. doi: 10.21468/SciPostPhysCodeb.7. URL `https://scipost.org/10.21468/SciPostPhysCodeb.7`.

Appendix 1: Preliminary Study

In this preliminary analysis, we will study how the number of parameters to optimise influences the accuracy of simulations. We will also examine how the number of parameters scales with system size and how execution time increases with parameters. The system chosen for the study is the TFI model with $J = 1$ and $h = 1$ with periodic boundary conditions. Algorithms need to find the ground state of the system. These values of J and h correspond to a quantum critical point. Thus, algorithms face trouble finding the ground state.

The NQS used for the study is the Restricted Boltzmann Machine (RBM), and the VQA are RealAmplitudes (RAmp) and Hamiltonian Variational Ansatz (HVA).

RBM

We used Netket[7] python package to implement RBM and conduct the study. NetKet provides RBM model and we chose textitparam-dtype as complex. As a result, RBM will have complex parameters to optimise. The RBM states are sampled using the $MetropolisLocal$ sampler with the number of samples set to 10000. The algorithm used for optimisation was Stochastic Gradient Descent (SGD) with a learning rate of 0.1. Stochastic reconfiguration was used as the preconditioner to improve the convergence of SGD.

In RBM, the number of parameters depends upon the alpha value. We have studied how the number of parameters to be optimised scales with the α value and the number of spins. We also investigated how the accuracy is affected by the $alpha$ value and its relation to the number of iterations of the optimisation algorithm.

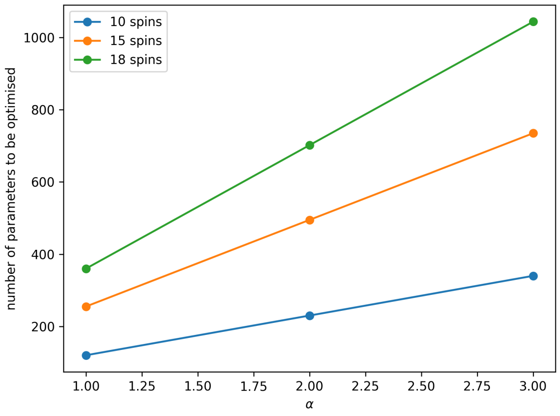

Figure 4: Alpha versus number of parameters to optimise. The number of parameters to optimise scales linearly with the alpha value.

Observations:

- From the figure 4, we can see the number of parameters scales linearly with the alpha value. In addition, the number of parameters to be optimised depends on the number of spins, which is evident at higher alpha values.
- The influence of α parameter on accuracy was studied and is given in figure 5. Further, the dependence of the number of iterations on improving accuracy was also studied.
- The results of the iterations are given in Appendix 2.
  - For $N = 10$ spins, the least error of the order $\mathcal{O}(10^{-6})$is found at $\alpha = 3$ and $iterations = 500$ but takes roughly 20 minutes to converge with 340 parameters to optimise. A sub-

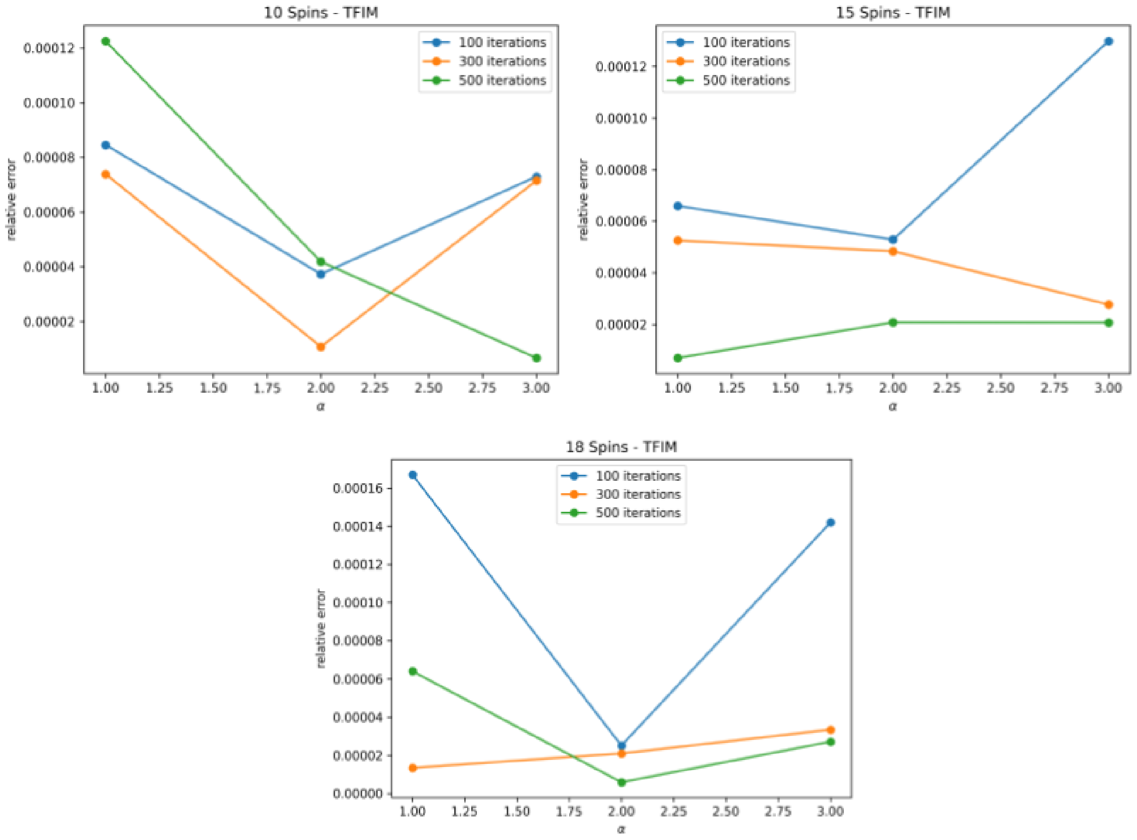

Figure 5: Alpha versus Relative error. Number of iterations was chosen as {100,300,500}.

optimal solution is given by $alpha = 1$ and $iterations = 300$ with a good $\hat{R}$ value in $\sim 5mins$. The error of the sub-optimal solution is of the order $\mathcal{O}(10^{-5})$.

- For $N = 15$ spins,the the least error is achieved when $alpha = 1$ and $iterations = 500$. However, as discussed in the case of $N = 10$, the time for convergence is high, and a sub-optimal solution will be $alpha = 1$ and $iterations = 3$, which only takes half the time for convergence than the former. The error of the sub-optimal solution is of the order $\mathcal{O}(10^{-5})$.
- For $N = 15$ spins,the the least error is achieved when $alpha = 2$ and $iterations = 500$.By considering the time for convergence and $\hat{R}$ values, we can consider a sub-optimal solution of $alpha = 1$ and $iterations = 300$ which have error of order $\mathcal{O}(10^{-5})$.

- In all trials conducted, the metropolis sampling converged with $|\hat{R}| < 1.1$.

RealAmplitudes

RealAmplitudes (RAmp) ansatz was tested to find the TFI model's ground state for three different values of spin $N = \{10, 15, 18\}$. It belongs to the class of template ansatzes provided by Qiskit. Using RAmp, the ground state is found using the variational quantum eigensolver algorithm. The quantum part of VQE is implemented in the local simulator called *BackendEstimator* provided by Qiskit. The classical optimisation part is done using the "COBYLA" algorithm provided by Scipy. COBYLA belongs to the class of gradient-free optimisers. The *maxiter* value of the optimiser was fixed at 1000 throughout the simulation. To remove the dependence of initialisation on the result, the initial state was fixed through simulation by fixing the seed.

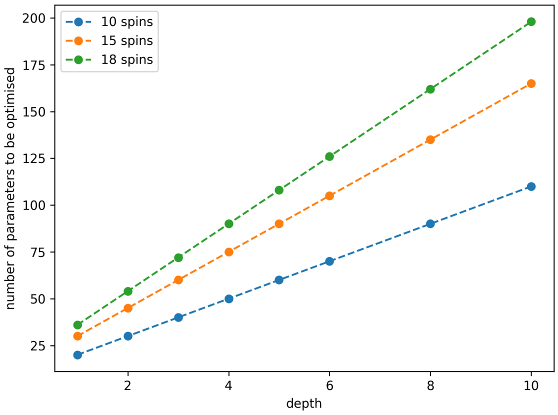

Figure 6: Relation between the number of variational parameters and depth of the RealAmplitudes ansatz.

In variational quantum algorithms, the depth of the quantum circuit determines the parameters to be optimised. We have studied how the number of parameters scales with the circuit depth. We also studied how the circuit's depth influences the simulations' accuracy. The depths considered are $\{1, 2, 3, 4, 5, 6, 8, 10\}$.

Note: *COBYLA is a gradient-free optimization algorithm. In the case of RBM, we used SGD, which is gradient-based. We faced problems when we tried to use gradient-based optimisers in the current version of qiskit. Thus, we carried out studies using COBYLA. Qiskit's optimisers work differently than Netket's optimisers. In Qiskit, we cannot control the number of iterations. We can only give an upper bound of iterations to be done called maxiter value. The iterations can terminate before or at the maxiter value depending upon the landscape of the cost function of the optimisation problem.*

Observations:

- Similar to the case of RBM, the number of parameters scales linearly with the depth of the circuit and is dependent on the number of spins in the system (ref. figure 6).
- Relation between accuracy and circuit depth was studied on various TFI models by varying spins and plotted in the figure 7.
- The results of the simulation are provided in Appendix 2.
  - For $N = 10$ system, at a circuit depth of 6 to least error solutions. However, the iterations terminated at the *maxiter* value of 1000, which indicates the algorithm still hasn't converged properly. Thus, $depth = 1, 2, 3$ is a better alternative, considering the execution time and number of iterations.
  - For $N = 15$ system, optimal solution is found at $depth = 3$. But here, the algorithm hasn't properly converged. Thus, $depth = 1, 2$ will be a better alternative regarding execution time and iteration count.
  - For $N = 18$, the optimal solution is given by $depth = 2$ but execution takes more than 50 minutes to converge. Considering the time taken, $depth = 1$ provides a sub-optimal solution with an error of the same order.

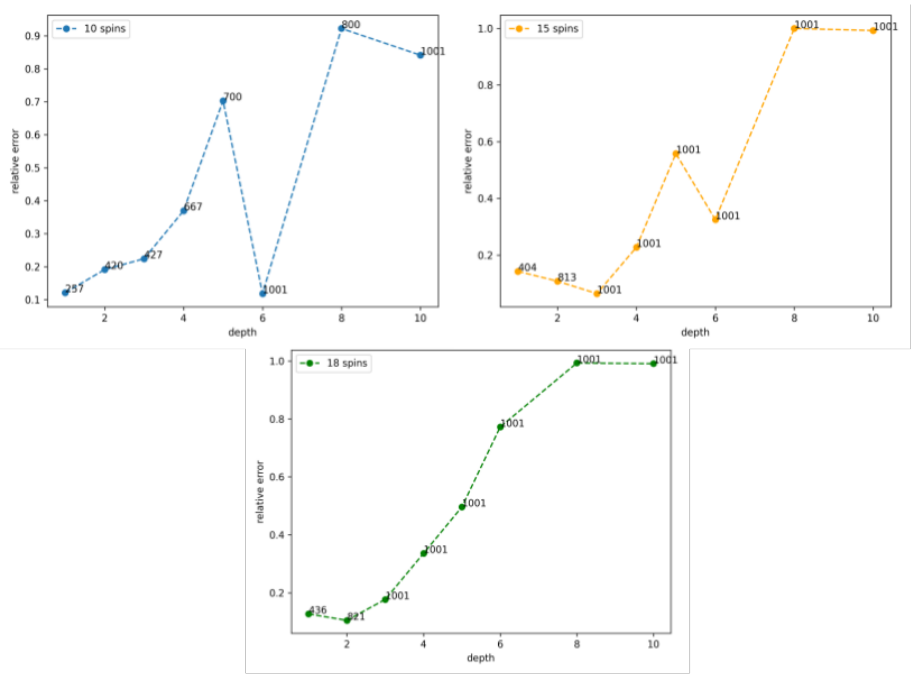

Figure 7: Relation between circuit depth and accuracy of the simulation in RAmp. Each point in the plot is annotated by the number of iterations required to find the optima.

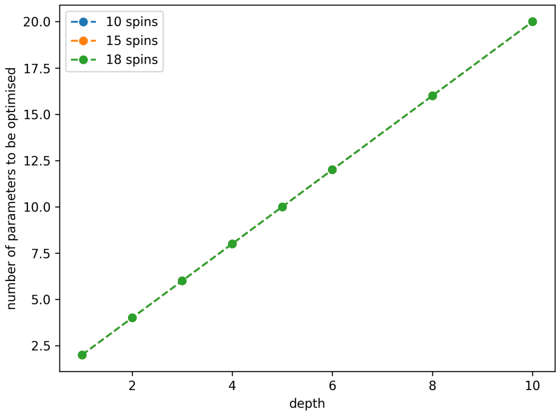

Figure 8: Relation between the number of variational parameters and depth of the quantum ansatz.

HVA

We constructed an HVA ansatz for TFIM using Qiskit. By using it as an ansatz in VQE, we can find the ground state of the system considered. The quantum part was carried out in the local simulator (*BackendEstimator*), and the classical part was carried out using the 'COBYLA' algorithm. Similar to the case of RAmp, the *maxiter* value was fixed at 1000, and the initial point was also fixed using a seed. The depth of the quantum circuit determines the number of parameters to optimise. We have studied how the circuit depth and parameters are related. Further, we also investigated how the circuit's depth influences the simulation's accuracy. The depth considered is {1,2,3,4,5,6,8,10}.

Observations

- From the figure 8, we can see that the number of variational parameters scales linearly with

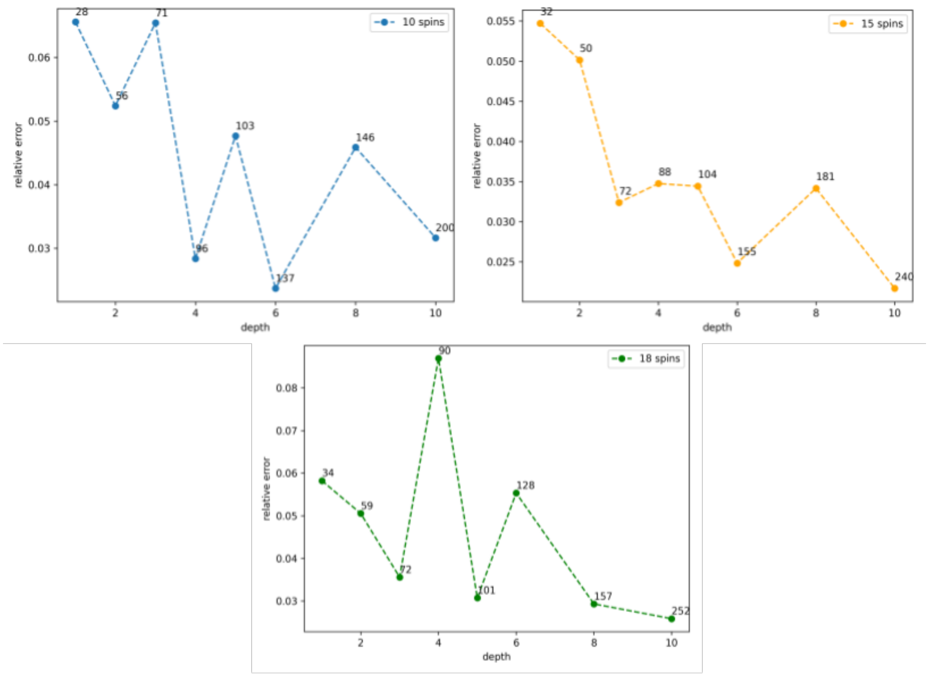

Figure 9: Relation between circuit depth and accuracy of the simulation in HVA. Each point in the plot is annotated by the number of iterations required to find the optima.

circuit depth, and it is **independent of the number of spins**.

- Relation between accuracy and circuit depth was studied on various TFI models by varying spins and plotted in the figure 9.
  - For $N = 10$ systems, the lowest error was found to circuit depth 6 with number of iterations = 137. For circuit depth 4, the error is roughly the same as the case $depth = 6$. Compared to later, $depth = 4$ requires fewer iterations, thus resulting in faster convergence. The timing comparison for various depths is given in the Appendix 2.
  - For $N = 15$ systems, the least error is found at $depth = 10$, but requires roughly $12mins$ to converge. Roughly the same level of accuracy can be found with $depth = 6$, which roughly takes only half the time required by $depth = 10$.
  - For $N = 18$, least error is found at $depth = 10$, which as in the case of $N = 15$ takes larger time for convergence (more than $1hr$). The same result can be obtained with $depth = 3$ circuits under 10 mins.

Inference

- From the preliminary studies on RBM and VQA, we could conclude that accuracy depends on the number of parameters for optimisation.
- In both algorithms, the accuracy and number of parameters do not show a direct relation. In some instances of the model tested, RBM attains an error of order $\mathcal{O}(10^{-6})$, whereas VQA fails to cross the 10^{-2} mark.
- In the case of RBM, the number of parameters depends upon the alpha value, and it scales linearly with the alpha value and number of spins in the system.
- In the case of RAmp and HVA, the number of parameters depends upon the circuit depth,

and it has linear scaling with depth.

- In comparison, RAmp shows a dependence of parameters on the number of spins, whereas HVA is independent of the number of spins.
- When comparing the accuracy of the simulations, RAmp shows an error of $\mathcal{O}(10^{-1})$ whereas HVA only has an error of $\mathcal{O}(10^{-2})$.

Key Takeaways

- The optimum alpha value found in the case of RBM may not apply to other TFI models with J and h different. The same goes for RAmp and HVA. Further studies must be conducted by varying J and h values to find the actual optimum values of Alpha or depth.
- At least for the TFI model, iteration counts do not influence much of the result given by RBM. The error mainly changes by an order for different iteration counts. Thus, considering accuracy and execution time, the number of iterations is fixed at 300 for future simulations with RBM.
- For comparing both VQA and RBM on the same foot, we are also fixing $maxiter = 300$ for VQA simulations in future studies.
- We are also fixing the number of spins = 15 for further studies considering execution time and computing capabilities.
- Since HVA performs better than RAmp, we will only consider HVA as a variational quantum ansatz in future studies. Since both RBM and HVA are closely related to the structure of TFIM Hamiltonian, comparing their performance gives much more insight than using heuristic trial wavefunction like RealAmplitudes.

Appendix 2: Data

Preliminary Trails

RBM-Preliminary Trails

Trails were conducted using a Restricted Boltzmann machine using TFIM with spins N = 10,15,18 with $J = 1$ and $h = 1$. Hyperparameters chosen:

- Optimiser: Stochastic Gradient Descent
- Learning Rate: 0.1
- Sampler: MetropolisLocal
- Number of samples: 10000
- Preconditioner = Stochastic Reconfiguration with Holmorphic = True.

alpha	num parameters	num of iterations	relative error	R hat	exec time
1	120	100	8.459757e-05	1.001112	1:47
2	230	100	3.7308397e-05	1.001206	2:49
3	340	100	7.29621e-05	1.001141	4:10
1	120	300	7.386877e-05	1.000911	5:13
2	230	300	1.0788759e-05	1.000880	9:31
3	340	300	7.146389e-05	1.000905	13:55
1	120	500	0.00012249289	1.000848	7:47
2	230	500	4.1871514e-05	1.000832	13:36
3	340	500	6.7106507e-06	1.000830	20:11

Table 3: 10 spin trails

alpha	num-parameters	num-of-iterations	relative-error	R-hat	exec-time
1	255	100	6.593547e-05	1.001226	3:27
2	495	100	5.284625e-05	1.001305	6:7
3	735	100	0.00012979655	1.001472	9:36
1	255	300	5.2526528e-05	1.000965	10:40
2	495	300	4.835761e-05	1.000979	19:4
3	735	300	2.7723849e-05	1.001006	31:6
1	255	500	7.0150104e-06	1.000883	18:14
2	495	500	2.076912e-05	1.000883	32:41
3	735	500	2.0696745e-05	1.000883	51:19

Table 4: 15 spin trials

-

alpha	num parameters	num of iterations	relative error	R hat	exec time
1	360	100	0.00016704111	1.001207	5:20
2	702	100	2.505726e-05	1.001486	9:40
3	1044	100	0.00014201373	1.001359	14:39
1	360	300	1.3399025e-05	1.000938	16:37
2	702	300	2.0936941e-05	1.001043	30:30
3	1044	300	3.3387758e-05	1.001019	45:27
1	360	500	6.402951e-05	1.000924	27:40
2	702	500	5.8390874e-06	1.000917	51:7
3	1044	500	2.709171e-05	1.000916	77:32

Table 5: 18 spin trails

4.0.1 RAmp-Preliminary Trails

Trails were conducted using a Restricted Boltzmann machine using TFIM with spins N = 10,15,18 with $J = 1$ and $h = 1$. Hyperparameters Chosen:

- Optimizer = COBYLA
- Maxiter = 1000
- Number of Shots = 10000
- Backend = BackendEstimator (Local simulator)

circuit-depth	num-parameters	num-of-iterations	relative-error	exec-time
1	20	257	0.121035	1:59
2	30	420	0.191112	3:12
3	40	427	0.224159	3:23
4	50	667	0.369583	5:33
5	60	700	0.703018	6:6
6	70	1001	0.118432	8:52
8	90	800	0.921807	7:43
10	110	1001	0.840932	10:16

Table 6: 10 Spins trails

circuit-depth	num-parameters	num-of-iterations	relative-error	exec-time
1	30	404	0.142754	7:1
2	45	813	0.108148	15:24
3	60	1001	0.064330	21:25
4	75	1001	0.227564	23:59
5	90	1001	0.558238	28:24
6	105	1001	0.324978	30:23
8	135	1001	0.999367	38:31
10	165	1001	0.992192	43:13

Table 7: 15 Spins trail

-

circuit-depth	num-parameters	num-of-iterations	relative-error	exec-time
1	36	436	0.127322	20:21
2	54	821	0.105198	52:33
3	72	1001	0.177339	82:16
4	90	1001	0.336151	101:34
5	108	1001	0.496227	121:34
6	126	1001	0.772166	143:22
8	162	1001	0.992935	180:30
10	198	1001	0.990528	212:35

Table 8: 18 Spins trail

4.0.2 HVA- Preliminary Trails

Trails were conducted using a Restricted Boltzmann machine using TFIM with spins N = 10,15,18 with $J = 1$ and $h = 1$. Hyperparameters Chosen:

- Optimizer = COBYLA
- Maxiter = 1000
- Number of Shots = 10000
- Backend = BackendEstimator (Local simulator)

circuit-depth	num-parameters	num-of-iterations	relative-error	exec-time
1	2	28	0.065611	0:12
2	4	56	0.052389	0:31
3	6	71	0.065436	0:40
4	8	96	0.028364	0:55
5	10	103	0.047670	0:56
6	12	137	0.023688	1:18
8	16	146	0.045866	1:35
10	20	200	0.031622	2:19
Over parameterization regime				
12	24	277	0.042972	3:23
14	28	247	0.033862	3:14
16	32	331	0.028748	4:36
18	36	374	0.025586	5:27
20	40	456	0.018994	7:8

Table 9: 10 spins trail

—-

circuit-depth	num-parameters	num-of-iterations	relative-error	exec-time
1	2	32	0.054713	0:35
2	4	50	0.050150	1:8
3	6	72	0.032384	1:50
4	8	88	0.034747	2:33
5	10	104	0.034424	3:28
6	12	155	0.024840	5:38
8	16	181	0.034161	7:58
10	20	240	0.021691	12:17

Table 10: 15 Spin trails

circuit-depth	num-parameters	num-of-iterations	relative-error	exec-time
1	2	34	0.058229	2:11
2	4	59	0.050531	5:47
3	6	72	0.035588	9:4
4	8	90	0.086915	14:22
5	10	101	0.030719	19:13
6	12	128	0.055322	28:21
8	16	157	0.029316	43:58
10	20	252	0.025846	85:40

Table 11: 18 Spin trails

–

Varying "h" value

This section contains data from trials conducted for different values of 'h' in the TFI model with 15 spins on a chain. The number of iterations of SGD was fixed at 300 and for COBYLA max. iteration value was given as 300.

RBM trails

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
1	0.100000	1	7.272996e-06	255	1.000793	7:0
1	0.100000	2	2.8440663e-06	495	1.000732	12:32
1	0.100000	3	1.2234735e-06	735	1.000815	18:48
1	0.100000	4	2.1617586e-06	975	1.000755	24:6

Table 12: TFIM with $h = 0.1$.

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
1	0.500000	1	7.2125085e-06	255	1.000761	9:33
1	0.500000	2	1.2024654e-05	495	1.000757	15:36
1	0.500000	3	1.8829393e-05	735	1.000782	23:16
1	0.500000	4	2.1124491e-05	975	1.000782	29:43

Table 13: TFIM with $h = 0.5$.

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
1	1	1	7.305345e-05	255	1.001005	10:14
1	1	2	1.8743029e-05	495	1.000933	18:28
1	1	3	2.8032013e-05	735	1.000977	28:49
1	1	4	3.482331e-05	975	1.000997	37:25

Table 14: TFIM with $h = 1$.

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
1	2	1	2.648958e-05	255	1.001062	7:43
1	2	2	3.0588726e-05	495	1.001302	14:31
1	2	3	2.7444303e-05	735	1.001237	23:5
1	2	4	4.2729578e-05	975	1.001362	30:8

Table 15: TFIM with $h = 2$.

–

HVA trails

	J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
0	1	0.100000	1	0.000043	2	33	0:38
1	1	0.100000	2	0.000139	4	56	1:18
2	1	0.100000	3	0.001781	6	68	1:47
3	1	0.100000	4	0.000584	8	94	2:50
4	1	0.100000	5	0.002222	10	119	4:0
5	1	0.100000	6	0.001843	12	140	5:12
6	1	0.100000	8	0.010553	16	160	7:5
7	1	0.100000	10	0.017487	20	248	12:49

Table 16: TFIM with $h = 0.1$.

	J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
0	1	0.500000	1	0.006684	2	26	0:29
1	1	0.500000	2	0.014349	4	51	1:9
2	1	0.500000	3	0.019104	6	73	1:56
3	1	0.500000	4	0.004977	8	102	2:59
4	1	0.500000	5	0.013830	10	116	3:50
5	1	0.500000	6	0.005694	12	139	5:6
6	1	0.500000	8	0.007673	16	150	6:35
7	1	0.500000	10	0.002176	20	225	11:24

Table 17: TFIM with $h = 0.5$.

	J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
0	1	1	1	0.054713	2	32	0:34
1	1	1	2	0.037494	4	60	1:20
2	1	1	3	0.034303	6	64	1:36
3	1	1	4	0.024411	8	90	2:33
4	1	1	5	0.024611	10	115	3:39
5	1	1	6	0.023650	12	136	4:50
6	1	1	8	0.033768	16	175	7:32
7	1	1	10	0.018500	20	214	10:41

Table 18: TFIM with $h = 1$.

	J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
0	1	2	1	0.240443	2	31	0:32
1	1	2	2	0.176214	4	54	1:17
2	1	2	3	0.125955	6	69	1:44
3	1	2	4	0.132583	8	91	2:34
4	1	2	5	0.150628	10	116	3:43
5	1	2	6	0.155865	12	126	4:31
6	1	2	8	0.095460	16	208	9:2
7	1	2	10	0.088703	20	209	10:36

Table 19: TFIM with $h = 2$.

Varying “J” value

This section contains data from trials conducted for different ‘ values’ in the TFI model with 15 spins on a chain. The number of iterations of SGD was fixed at 300 and for COBYLA max. iteration value was given as 300.

RBM trails

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
0.100000	1	1	2.262109e-06	255	1.001729	6:20
0.100000	1	2	2.6943195e-05	495	1.002259	11:44
0.100000	1	3	7.5254648e-06	735	1.002960	18:16
0.100000	1	4	3.840932e-06	975	1.001774	23:52

Table 20: TFIM with $J = 0.1$.

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
0.500000	1	1	3.170925e-05	255	1.001427	7:26
0.500000	1	2	2.7847793e-05	495	1.001346	13:38
0.500000	1	3	1.8421266e-05	735	1.001241	20:42
0.500000	1	4	3.5468554e-05	975	1.001443	27:11

Table 21: TFIM with $J = 0.5$.

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
1	1	1	7.305345e-05	255	1.001005	10:14
1	1	2	1.8743029e-05	495	1.000933	18:28
1	1	3	2.8032013e-05	735	1.000977	28:49
1	1	4	3.482331e-05	975	1.000997	37:25

Table 22: TFIM with $J = 1$.

-

HVA trails

J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
0.100000	1	1	0.451303	2	37	0:38
0.100000	1	2	0.316464	4	45	0:55
0.100000	1	3	0.235703	6	73	1:46
0.100000	1	4	0.233872	8	98	2:44
0.100000	1	5	0.287115	10	119	3:46
0.100000	1	6	0.235240	12	138	4:51
0.100000	1	8	0.178340	16	211	9:8
0.100000	1	10	0.162308	20	222	11:10

Table 23: TFIM with $J = 0.1$.

J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
0.500000	1	1	0.240443	2	31	0:32
0.500000	1	2	0.176214	4	54	1:11
0.500000	1	3	0.125955	6	69	1:45
0.500000	1	4	0.132583	8	91	2:32
0.500000	1	5	0.150628	10	116	3:40
0.500000	1	6	0.155865	12	126	4:29
0.500000	1	8	0.095460	16	208	9:7
0.500000	1	10	0.088703	20	209	10:37

Table 24: TFIM with $J = 0.5$.

J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
1	1	1	0.054713	2	32	0:34
1	1	2	0.037494	4	60	1:20
1	1	3	0.034303	6	64	1:36
1	1	4	0.024411	8	90	2:33
1	1	5	0.024611	10	115	3:39
1	1	6	0.023650	12	136	4:50
1	1	8	0.033768	16	175	7:32
1	1	10	0.018500	20	214	10:41

Table 25: TFIM with $J = 1$.

-

Negative J values:

Trails were conducted for the negative values of J for TFIM of 15 spins on a chain. The J values considered are $J = \{-0.1, -0.5, -1.0\}$. The number of iterations of SGD was fixed at 300 and for COBYLA max. iteration value was given as 300.

RBM trails

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
-0.100000	1	1	3.7988611e-06	255	1.003269	7:7
-0.100000	1	2	0.00051051314	495	1.002919	12:58
-0.100000	1	3	0.0009904812	735	1.002657	19:5
-0.100000	1	4	0.001329247	975	1.004491	24:58

Table 26: TFIM with J=-0.1

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
-0.500000	1	1	0.01633764	255	1.001204	8:1
-0.500000	1	2	0.012395287	495	1.001738	15:25
-0.500000	1	3	0.010488318	735	1.007990	24:35
-0.500000	1	4	0.19704048	975	1.005655	33:57

Table 27: TFIM with J=-0.5.

J-value	h-value	alpha	relative-error	num-parameters	R-hat	exec-time
-1	1	1	0.036539823	255	1.001027	10:0
-1	1	2	0.021062749	495	1.002181	20:35
-1	1	3	0.036558688	735	1.001102	28:15
-1	1	4	0.07598338	975	1.001463	38:31

Table 28: TFIM with J=-1.

-

HVA trails

J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
-0.100000	1	1	0.549399	2	34	0:36
-0.100000	1	2	0.372191	4	61	1:19
-0.100000	1	3	0.315183	6	81	1:59
-0.100000	1	4	0.263871	8	86	2:26
-0.100000	1	5	0.304724	10	123	3:58
-0.100000	1	6	0.297572	12	129	4:39
-0.100000	1	8	0.270032	16	162	6:57
-0.100000	1	10	0.234830	20	207	10:29

Table 29: TFIM with J=-0.1.

J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
-0.500000	1	1	0.711474	2	32	0:38
-0.500000	1	2	0.683295	4	50	1:11
-0.500000	1	3	0.393932	6	91	2:24
-0.500000	1	4	0.360169	8	95	2:46
-0.500000	1	5	0.318357	10	127	4:17
-0.500000	1	6	0.271035	12	132	4:55
-0.500000	1	8	0.282254	16	167	7:23
-0.500000	1	10	0.252525	20	233	12:7

Table 30: TFIM with J=-0.5.

J-value	h-value	reps	relative-error	num-parameters	num-of-iterations	exec-time
-1	1	1	0.837798	2	33	0:40
-1	1	2	0.582674	4	54	1:20
-1	1	3	0.508039	6	80	2:14
-1	1	4	0.388370	8	100	3:4
-1	1	5	0.430755	10	99	3:25
-1	1	6	0.385125	12	125	4:41
-1	1	8	0.328925	16	174	7:50
-1	1	10	0.406093	20	199	10:30

Table 31: TFIM with J=-1.