VQE for the J_1-J_2 model

Niyousha Najmaei and Praveen Viswanathan

Mentor: Leonardo DiCarlo

Delft University of Technology, The Netherlands

30 January 2023

CONTENTS

I. INTRODUCTION

The significant progress made in quantum hardware over the past few years [1, 2], and to be made in the coming years, will soon open up the era of noisy intermediate-scale quantum computation(NISQ) [3]. Hybrid classical-quantum (HCQ) algorithms are the most likely candidates to be run on these non-fault-tolerant devices [3, 4]. The Variational Quantum Eigensolver (VQE) is one such class of HCQ algorithms.

VQEs use the time-independent variational principle to approximate the ground state energy of a given system. These algorithms use a parametrised quantum circuit called the ansatz and a classical optimiser to suitably modify this ansatz's parameters to generate the ground state of the system Hamiltonian. This optimiser requires evaluating the energy, which means averaging the energy output over many measurements. The freedom of choice of an ansatz makes VQE flexible, and the averaging over multiple measurements lends it some robustness against noise [4, 5]. These aspects make VQE a promising application for near-term quantum computers.

In this project, we use a VQE algorithm to obtain the ground state energy of the J1-J2 model, a spin-lattice system. We experiment with (dynamic) modifications of the ansatz mid-optimisation and explore the construction of an ansatz from just a few starting gates. The results are centred around the improvement to VQE energy estimates, usually with a reduction in the average number of parameters the optimiser has to optimise.

The rest of this report is organised as follows. Sec. II outlines the basics of the J_1-J_2 model and its relevant aspects for our experiments. Sec. III is a short introduction to the class of VQE algorithms and details our implementation. Sec. IV defines the two ansätze we worked with and is followed by a section on the results of running VQE using these (static) ansätze. This section also includes the issues we found with running VQE on these static ansätze and, thus, the motivation for our experiments. We propose our ideas for dynamic ansätze in Sec. VI and analyse their results against static ansätze in Sec. VII. We close in Sec. VIII with a summary of our work and an outlook on the project.

II. J_1-J_2 MODEL

The J_1-J_2 model is an extension of the Heisenberg model on a square lattice [5]. The Heisenberg model involves nearest neighbour interactions (of strength J_1), and the J_1-J_2 model extends this to include next-to-nearest neighbour interactions. These next-to-nearest neighbour interactions are denoted by the blue diagonal lines in Fig. 1 and have a coupling constant of J_2.

The Hamiltonian for the J_1-J_2 system is given by

$$\mathcal{H} = -J_1 \sum_{\langle i,j \rangle} \vec{S_i} \cdot \vec{S_j} - J_2 \sum_{\langle\langle i,j \rangle\rangle} \vec{S_i} \cdot \vec{S_j} \tag{1}$$

where $\langle i,j \rangle$ indicates a sum over nearest neighbours and $\langle\langle i,j \rangle\rangle$ indicates a sum over next-nearest neighbours. $\vec{S_i}$ are the three element Pauli spin-1/2 operator vectors (X_j, Y_j, Z_j) for spin i [5]. Note that we consider no external magnetic field, i.e., $h = 0$.

Studying the J_1-J_2 model is required to understand specific physical systems, for instance, the magnetic properties of CuO_2 planes in high-T_c cuprate superconductors and that of vanadates. For these and other relevant physical problems, we need to analyse the J_1-J_2 model in the antiferromagnetic regime. The interested reader is directed to Ref. [6] for

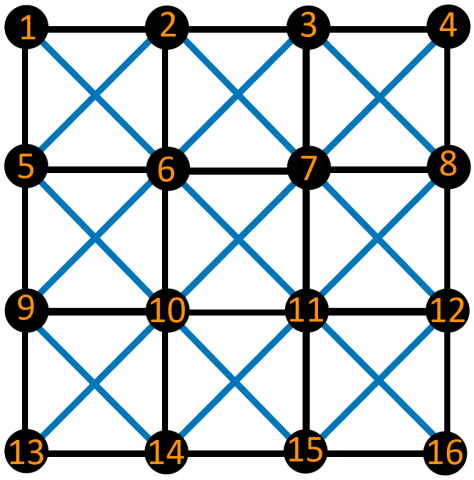

FIG. 1: The J_1-J_2 model on a 4x4 square lattice. The black lines between adjacent lattice sites represent nearest neighbour interaction with coupling constant J_1, and the blue diagonal lines represent next-to-nearest neighbour interactions with coupling constant J_2 (Ref. [5]).

more on different configurations of J_1 and J_2 and related physical applications.

The signs of the constants J_1 and J_2 denote the nature of interaction: a positive value of J_1 means ferromagnetic interaction between nearest neighbours, and a negative value indicates an antiferromagnetic one. The same holds for J_2 and next-to-nearest neighbours. In our experiments, we focus on antiferromagnetic interactions for both J_1 and J_2. A complete classical analytical method for analysing the J_1-J_2 model in the antiferromagnetic regime does not exist. Although classical numerical approximations exist, when $0.4 \leq J_2/J_1 \leq 0.6$, different methods give different, incompatible results [5–7]. We aim to use VQE to solve for the ground state of the J_1-J_2 model in this highly frustrated regime, at $J_2/J_1 = 0.5$ We take $J_1 = -1$, and all the energies in this report are in suitable units.

The boundaries of the square lattice can be of two types:

- Periodic Boundary: we consider wrap-around interactions between 1. the leftmost and rightmost spins and 2. the topmost and bottommost spins in the square lattice. For instance, in Fig. 1, we would consider nearest neighbour interactions between spins 2 and 14, 5 and 8, and the next-to-nearest neighbour interaction between 1 and 14.

- Open Boundary: we do not consider wrap-around interactions. In general, designing efficient problem-inspired ansätze is easier for open boundaries since we do not require entangling (2-qubit) gates between qubits on opposite ends of the lattice. These wrap-around 2-qubit gates would be tough to realise on current or near-future quantum processors, which are expected to be 2D arrays of qubits [5].

We focus on the open boundary condition since we have a few (good) VQE estimates for J_1-J_2 systems of this type in Ref. [5].

It should be noted before introducing VQEs that while the Hamiltonian for the J_1-J_2 model can be written as a simple matrix, it is infeasible to diagonalise it on a classical computer. This is because the dimensions of the problem grow exponentially with the lattice size. During our experiments, our computers struggled to find the eigenvalues for

the 4x4 lattice. Hence we are limited to classical numerical methods for the systems of interest, and hopefully HQCs like VQE in the near future, for the analysis of the J_1-J_2 model.

Regardless, the true eigenvalue of the system remains the best metric to judge the accuracy of our VQEs, so in this project, we stick to lattice sizes less than or equal to 4x4 since the corresponding matrix was the largest our computers could diagonalise. The actual ground state and first excited state energies that we obtained through matrix diagonalisation for the antiferromagnetic J_1-J_2 lattice with $J_2/J_1 = 0.5$ and open boundaries are listed below:

Lattice size	Ground state energy $\mathcal{E}_0$	First excited state energy $\mathcal{E}_1$
3x3	-15.837	-13.086
3x4	-22.138	-20.156

TABLE I: Values of actual ground state and first excited state energies for J_1-J_2 lattices of size 3x3 and 3x4

III. VARIATIONAL QUANTUM EIGENSOLVER

A. Recipe

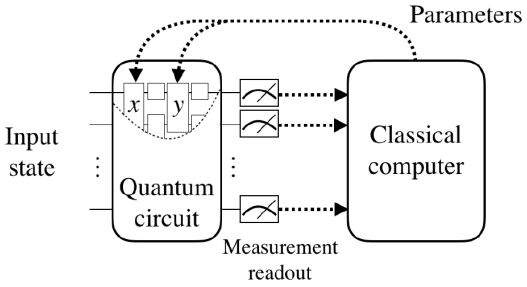

FIG. 2: Outline of a hybrid quantum-classical algorithm. An initial quantum state is passed through a parametrised quantum circuit(PQC), and relevant measurements are performed on the final quantum state output from the PQC. A classical computer uses these measurements to suitably update the parameters of the PQC (Ref. [4]).

The goal of a VQE is to approximate the eigenvalues, usually the ground state eigenvalue, of a given Hamiltonian. It uses the time-independent variational principle, which states that for a given Hamiltonian $\mathcal{H}$, and any state $|\Psi\rangle$,

$$\mathcal{E}_0 \leq \langle\Psi|\mathcal{H}|\Psi\rangle \tag{2}$$

where $\mathcal{E}_0$ is the ground state energy (lowest eigenvalue) of the Hamiltonian $\mathcal{H}$. Suppose we were somehow able to continuously modify $|\Psi\rangle$ to keep reducing its energy $E = \langle\Psi|\mathcal{H}|\Psi\rangle$. Subsequently, the state will reach the ground state, and we will no longer be able to decrease its energy. VQE attempts to do exactly this.

We start with the Hamiltonian for the system of interest. We generally need to use encodings like the Jordan-Wigner and Bravyi-Kitaev encodings to map quantum chemistry

problems to a quantum computer [8]. These mappings ensure that the Hamiltonian is expressed in terms of the Pauli operators, which are easy to work with on quantum computers. Fortunately, the J_1-J_2 model is already defined as a sum of Pauli operators, so we do not need to use these transformations. This collection of Pauli strings informs us of the measurements necessary to evaluate the energy of any given quantum state.

We then design a parametrised quantum circuit (PQC), also called an ansatz. Our target, at the end of classical optimisation, is for this PQC to output the system's ground state. This PQC is initialised with random parameters, so initially, the output of the PQC would be far from the ground state. We design measurements according to the Pauli strings obtained from the Hamiltonian to obtain the energy of the PQC output state.

The final ingredient for a VQE is a classical optimiser. A classical optimiser uses the measurements obtained from running the PQC to update its parameters. Following the parameter updates, we rerun the PQC to obtain new measurements, which hopefully lead to a lower energy estimate than the prior run. These new measurements are used to rerun the classical optimiser to get fresh parameter updates. This cycle of PQC→measurements→optimiser→parameter updates→PQC is repeated many times, and one such cycle is called an iteration. Once the optimiser attains convergence for the energy estimate, we would stop noticing significant decreases in the energy estimate. This final convergent energy estimate is called the VQE Estimate.

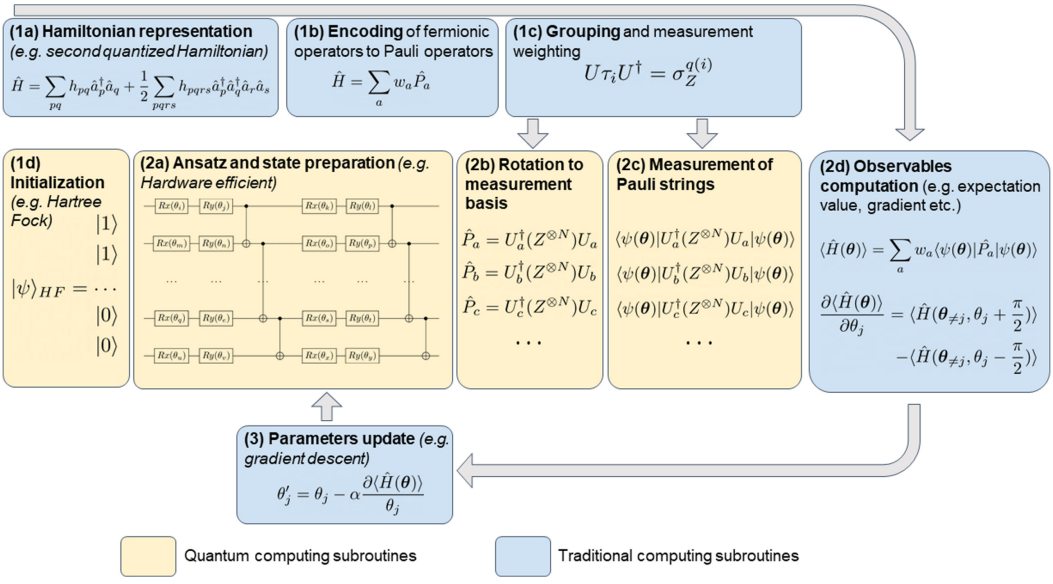

FIG. 3: An example of the VQE algorithm, a hybrid quantum-classical algorithm that uses a parameterised quantum circuit. Measurements on the states generated by the circuit and a classical optimiser are used to update the parameters of the quantum circuit. The steps under (1) are pre-processing and must only be done once. Steps (2) and (3) are done in a loop until we have a convergence of the VQE Estimate (Ref. [9]).

.

Classically, ansätze are initial guesses to a given problem. They can be thought of similarly with quantum circuits, as these circuits are usually designed with the system's symmetry in mind. Such problem-inspired ansätze typically leads to higher accuracy of the final energy estimate with lower circuit depths [10].

B. Classical Optimiser

Classical optimisers play an important role in deciding the parameter updates. While selecting a classical optimiser for an HQC algorithm, we have to ensure that it is capable of handling shot-noise; despite the averaging of measurements, the number of shots is finite and the energy estimate obtained through averaging over measurements may not be exactly equal to the expectation value of the Hamiltonian. We can attempt to increase the shot-count to very large numbers so that the energy estimate is essentially exact, but this is infeasible both with simulators as well as quantum hardware. Furthermore, when running on quantum hardware, the classical optimiser would also need to handle noise due to quantum gate fidelities and qubit decoherence.

These optimisers can be classified into gradient-based and gradient-free optimisers, depending on whether the optimiser requires gradient information. The effect and performance difference due to the choice of classical optimiser has been explored in numerous articles, and we referred to Ref. [11]. The authors claim that in general gradient-based optimisers perform better than their counterparts if we only have shot-noise, but when we also factor in the quantum device noise, the gradient-free optimisers perform better. Regardless, we performed our experiments with one optimiser from each category.

The entire VQE procedure can be considered to be the classical optimiser attempting to navigate the parameter landscape to find the global minima. Such an image immediately brings to light the problems associated with local minima and the relevance of the choice of initial parameters. If we start with bad initial parameters, the optimiser may get stuck in local minima and never reach the global minima, i.e., the actual ground state of the system. Usually, this is tackled by running the entire VQE protocol multiple times, starting with random initial parameters. This leads to a large overhead in the overall computational time.

Investigating methods for preventing the classical optimiser from getting stuck in these local minima, was a major part of our experiments. These ideas and results are presented in Secs. VI, VII.

C. Implementation Details

In our experiments, we stuck to ansätze with only single and two-qubit gates. This is mainly because these are the gates likely to have high fidelities in the NISQ regime. Furthermore, many superconducting and spin-qubit systems are planned to be designed in 2D rectangular arrays, which allows for easy mapping of the J_1-J_2 lattice spins (ref. Fig.1) onto the qubits [5, 12]. Two qubit gates are likely to exist between nearest neighbour qubits, so we only use two-qubit gates between these pairs. This means we do not use any two-qubit gates to represent the diagonal interactions directly, as these will lead to the compiler-level introduction of multiple swaps, which may lead to longer circuits, longer evaluation times, and more noise introduced in the experiments.

We were primarily interested in lattice sizes of 3x4 and 4x4. Since we had access to a maximum of 7 qubits, we were restricted to using simulations for this project. Furthermore, the few runs (of a lattice of 2x2) we ran on cloud quantum computing services taught us that it is infeasible to run VQE on even a moderately large quantum circuit. Each iteration of the optimiser is queued independently, each queue entry takes a few hours to run, and we require a few hundred iterations for the optimiser to converge. Thus it is prohibitively lengthy for us to do any meaningful experiments.

The majority of our experiments were carried out using Sequential Least Squares Programming (SLSQP) [13], a gradient-based optimiser, and Constrained Optimization BY Linear Approximation (COBYLA) [14], a gradient-free one which uses linear approximation of the target function. According to Ref. [11], SLSQP and COBYLA are among the best performers in their optimiser-categories. Another factor that motivated the choice of COBYLA was its use by the authors of Ref. [5].

We experimented with other classical optimisers that have been bench-marked for the VQE algorithm and shown to be useful in [15], as well as well-known extensions to the classic gradient descent. Specifically, we compared the estimated energies and the convergence times of Simultaneous Perturbation Stochastic Approximation (SPSA) [16], a gradient-based optimiser which uses stochastic perturbation vector to compute simultaneously an approximate gradient of the objective, and ADAptive Moment estimation (ADAM) [17] and AMSGrad [18], which are memory-based extensions to the classic gradient descent, to the previously mentioned optimisers. An example convergence plot which compares these optimisers can be seen in Figure 4. Considering the fact that COBYLA is a gradient-free optimiser, we were expecting this optimiser to perform better than the gradient-based ones. However, it is evident from the plots that SLSQP, ADAM and AMSGrad perform better than COBYLA in terms of the energy estimate, and that SLSQP achieves the best energy estimate among them all. Moreover, it can be seen that SPSA has not converged to a value even after twice the number of iterations as the other optimisers. This is also aligned with the findings of [15], which state that SPSA may not be well-suited for large problems.

We used IBM's qiskit library, the AerSimulator from the qiskit library with 10^6 shots and no device backend noise. That is, the optimiser had to deal with some shot-noise, but no realistic device noise. Our optimisers use the functions provided in the scipy library.

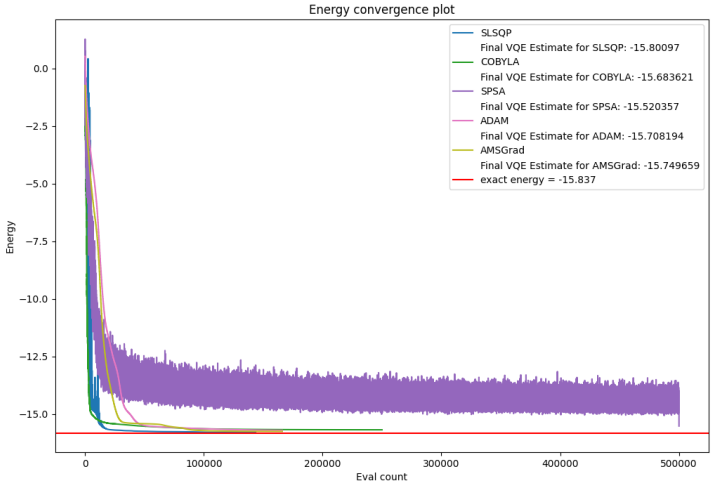

FIG. 4: Convergence comparison for the algorithm between four optimisers, SLSQP, COBYLA, ADAM and AMSGrad, when using the Feulner-Hartmann ansatz for a 3x3 lattice.

IV. STATIC ANSÄTZE

In this section, we describe the two main ansätze we investigated and lay out the terminology used in the rest of the report. The ansätze shown in this section are for the 3x3 J_1-J_2 lattice, as shown in Fig.5. The qubits in the ansätze are numbered directly after the numbers within the lattice sites. To avoid confusion, when referring to the sites of the lattice, say the first site, we call them q0, and when referring to their corresponding qubits, we call them q_0 (note the italicisation and subscript).

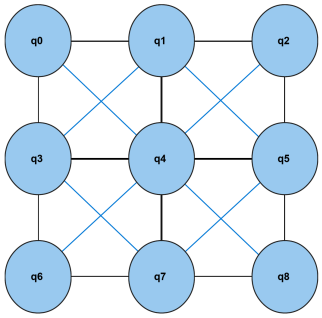

FIG. 5: A square J_1-J_2 lattice for three rows and three columns, with nearest neighbour interactions indicated as black lines and next-to-nearest neighbour interactions as blue lines.

A. Two Local Ansatz

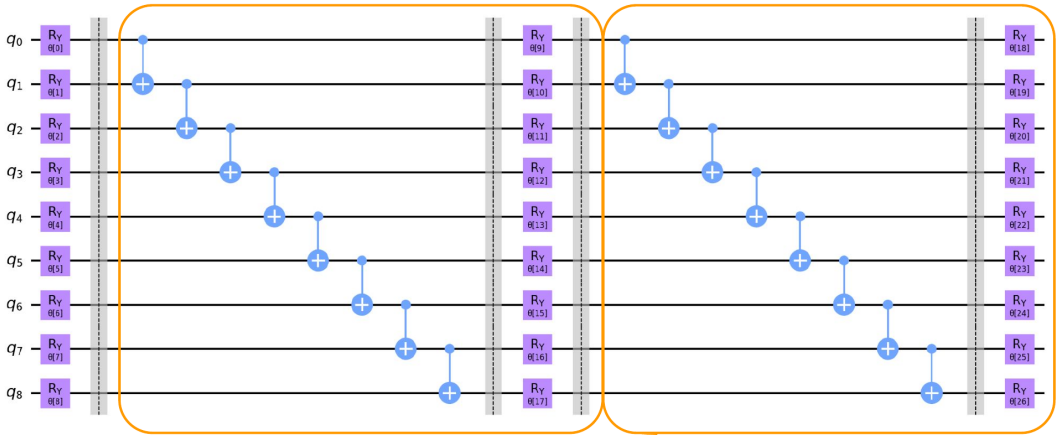

FIG. 6: Two layers of the two-local ansatz. A layer of the two-local ansatz, which comprises one set of entangling gates followed by one set of rotating gates, is indicated within an orange box. The grey rectangles are simple barriers that prevent the drawing software from moving around the gates. These have no role during execution.

The first ansatz we considered was a simple, hardware-efficient ansatz consisting of alternating sets of rotating gates and sets of entangling gates [1]. Our set of rotating gates had rotations about the Y-axis ($\mathsf{R_Y}(\theta)$) for all qubits, and the set of entangling gates were CNOT gates between adjacent qubits. Note that for this ansatz, we assumed a simple 1D layout

for the qubits, so each qubit has only two adjacent qubits. We map the lattice spins in 2 dimensions to a one-dimensional qubit lattice, so we lose most of the original connectivity. This 1D layout, however, means that it only needs a linear chain of connected qubits to work, so it can be realised sooner with quantum hardware, as compared to ansätze which require 2D connectivity. We call one entangling layer followed by one rotating layer, one layer of the two-local ansatz. Two such layers of the two-local ansatz are shown in Fig.6. The total number of parameters for the two-local ansatz for N qubits is

$$N \times (\#layers + 1)$$

As mentioned above, most of the original connectivity of the original lattice in Fig.5 is lost while using this ansatz. For instance, given the nearest neighbour interaction between sites q0 and q3 in Fig.5, we would incorporate some direct parametrised entangling gate between qubits q_0 and q_3 in our ansatz. However, we can see in Fig.6 that there is no such entangling gate in our two-local ansatz. Thus, we have to use a few layers of the two-local ansatz to allow the optimiser to generate the expected entanglement between qubits q_0 and q_3.

B. Feulner-Hartmann Ansatz

The other ansatz we considered was from Ref. [5]. The ansatz follows a very similar scheme to the two-local ansatz, with one layer containing rotating gates followed by entangling gates between two qubits. We use the XXYYZZ gate, which is the sequential composition of an $XX(\theta)$ gate, a $YY(\theta)$ gate and a $ZZ(\theta)$ gate, all of them sharing the same parameter θ, as the parametrised entangling gate for the ansatz. This XXYYZZ gate is shown for two qubits in Figs. 7 and 8. One complete layer of this ansatz is shown in Fig. 9. The total number of parameters for this ansatz for a lattice of dimensions $m \times n$ is

$$2mn + \#layers \times (mn + 2mn - (m + n))$$

$$\mathrm{XX}(\theta) = (X \otimes X)^\theta = \begin{pmatrix} c & 0 & 0 & s \\ 0 & c & s & 0 \\ 0 & s & c & 0 \\ s & 0 & 0 & c \end{pmatrix}$$

$$\mathrm{YY}(\theta) = (Y \otimes Y)^\theta = \begin{pmatrix} c & 0 & 0 & -s \\ 0 & c & s & 0 \\ 0 & s & c & 0 \\ -s & 0 & 0 & c \end{pmatrix}$$

$$\mathrm{ZZ}(\theta) = (Z \otimes Z)^\theta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & w & 0 & 0 \\ 0 & 0 & w & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

FIG. 7: These three gates are composed sequentially to form the XXYYZZ gate between two qubits. Note that they all share the same parameter θ (Ref.[5]).

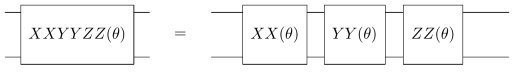

FIG. 8: The parametrised entangling block used in this ansatz, the XXYYZZ gate (Ref.[5]).

The entangling gates in this ansatz are between the nearest neighbours on the square lattice (Fig.5). For instance, the entangling gate between qubits q_0 and q_3 that we were missing in the two-local ansatz is present here. Since we are incorporating more elements of the problem-symmetry into this ansatz, we would expect the VQE estimates to reach the ground state with just a few layers of this ansatz. However, this also means we would require

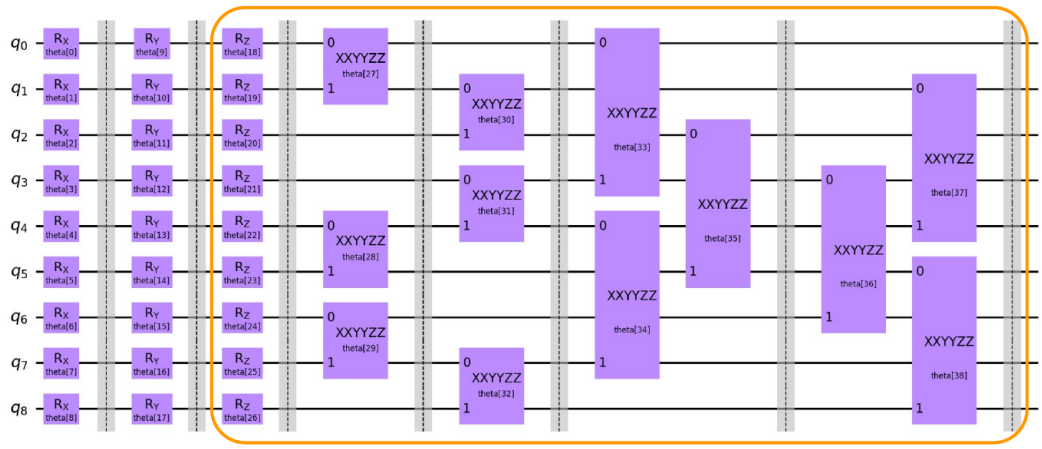

FIG. 9: One layer of the Feulner-Hartmann ansatz. A layer of this ansatz, which comprises one set of rotations about the Z-axis, followed by a set of XXYYZZ gates, is indicated within an orange box. The grey rectangles are simple barriers that prevent the drawing software from moving around the gates. These have no role during execution.

either a qubit array with two-dimensional connectivity or many swap gates for running this on quantum hardware. We should also note that we still do not have any entangling gates between diagonal qubits, e.g., q_0 and q_4. Again, looking at Fig.5, we would expect some entanglement between the spins at sites q0 and q4 and would expect that incorporating entangling gates between these diagonal qubits would lead to better results with fewer layers. However, these gates would not be easy to implement on quantum hardware and would require many additional swap gates to bring the diagonal qubits into connectivity. Furthermore, Ref.[5] claims that the reduction in the number of parameters by removing the diagonal gates actually improves the performance of the optimiser. We thus omit these in our simulations.

V. STATIC ANSÄTZE RESULTS

In this section, we provide the results of running the VQE algorithm for the two ansätze in Sec. IV and outline the issues we found while attempting to generate the ground state energy as the final VQE estimate. We use a metric from Ref. [5] to compare our results in this section and Sec.VII. If the final VQE estimate is E_{VQE}, and the actual ground state and first excited energies of the system are $\mathcal{E}_0$ and $\mathcal{E}_1$ respectively, we look at

$$\frac{|\mathcal{E}_0 - E_{VQE}|}{|\mathcal{E}_0 - \mathcal{E}_1|}$$

This value tells us how close the VQE estimate is to the ground state energy compared to the first excited state energy. For convenience, we refer to this as the accuracy of our final VQE estimate. As the VQE estimate gets closer to the ground state, this accuracy goes to 0. If the VQE estimate is terrible so that it does not even cross the first excited state energy $\mathcal{E}_1$, this value exceeds 1, and in these cases we omit mentioning the value. Note that the VQE estimate will never decrease beyond the ground state energy $\mathcal{E}_0$ due to the time-independent variational principle. Table I lists the relevant values of ground and first excited state energies for the systems we consider.

A. Two-Local Ansatz

We ran a set of experiments to find the number of layers for which the Two-Local ansatz converges to an acceptable estimate of the energy. We observed that the accuracy of the estimate improved when using more layers in the ansatz up to having 10 layers, and when increasing this number beyond 10, the results became worse. More specifically, when using 10 layers of the two-local ansatz for the 3x3 J_1-J_2 system with SLSQP as the optimiser, we obtained a final VQE Estimate of $E_{VQE} = -15.412$. The convergence plot for this run is shown in Fig.10. This result corresponds to an accuracy of

$$\frac{|\mathcal{E}_0 - E_{VQE}|}{|\mathcal{E}_0 - \mathcal{E}_1|} \approx 15.5\%$$

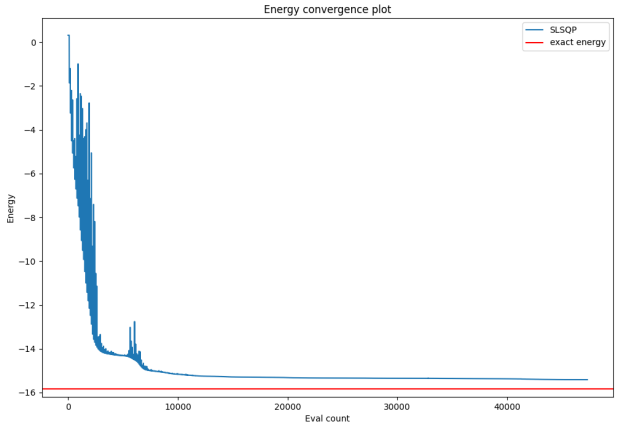

FIG. 10: Convergence plot for optimisation of 10 layers of the two-local ansatz using SLSQP. This was for a 3x3 J_1-J_2 lattice with $J_2/J_1 = 0.5$. The exact ground state energy for this system, $\mathcal{E}_0 = -15.837$, is shown as a horizontal red line. In the Y-axis, we have the VQE energy estimate obtained through the measurements of the Pauli strings encodings of the Hamiltonian (step 2c in Fig.3). In the X-axis, we have the number of evaluations of the quantum circuit. Depending on the optimiser, each iteration of the optimiser may require multiple evaluations. For instance, a gradient-based optimiser may evaluate the energy estimate for randomly sampled points within a small ball around the current parameter values to determine how to update the parameters. In this case, the optimiser ran for 1000 iterations but required around 50000 evaluations.

On increasing the number of layers to 16, our optimiser converged to a value of $E_{VQE} = -11.845$, which is quite far from the ground state energy. We hypothesised that the optimiser was getting stuck in a local minima and was unable to reach the global minimum corresponding to the actual ground-state energy, as we increased the number of layers beyond 10 layers.

To test this hypothesis, we ran the algorithm 100 times, for each case of the Two-Local ansatz having 10 and 16 layers, starting from different initial points. We expected to see most of the results to be distributed close to the actual ground state energy, and some far away which correspond to the optimiser getting stuck in local minima for those values of the initial parameters. Moreover, we expected the obtained results to be more focused towards the actual ground state in the case of using 10 layers. For each of these runs, i.e., for each

random initial value of the parameter, the optimiser was allowed to converge completely. These final convergent values are plotted as a histogram in Figs. 11 and 12. It is evident from the two histograms that the optimiser gets stuck in local minima more frequently when we have more layers in our ansatz. This is a central problem with optimisation for VQEs [15, 19], and we look at a few methods to tackle this in Sec. VI.

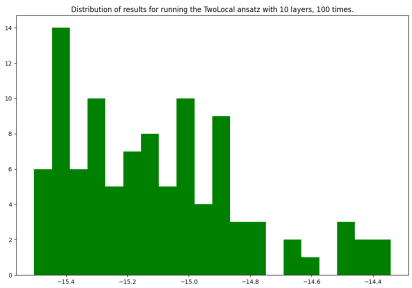

FIG. 11: Histogram of final VQE estimates when run with 100 different initial parameters for the 3x3 J_1-J_2 lattice, using 10 layers of the two-local ansatz. The optimiser was allowed to converge for each initial value, giving plots similar to Fig.10 for each.

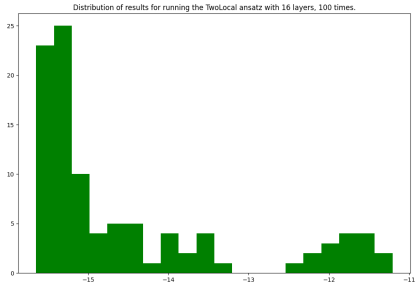

FIG. 12: Histogram of final VQE estimates when run with 100 different initial parameters for the 3x3 J_1-J_2 lattice, using 16 layers of the two-local ansatz. Note that in a few cases, despite the convergence, the optimiser gets stuck in local minima around -11.8. Note that the scale of the x-axis of this histogram is different from that of Figure 11.

B. Feulner-Hartmann Ansatz

On the optimisation of 7 layers of this ansatz (165 parameters) for the 3x3 J_1-J_2 system with SLSQP, we obtain a final VQE estimate of $E_{VQE} = -15.801$, with an accuracy of $|\mathcal{E}_0 - E_{VQE}|/|\mathcal{E}_0 - \mathcal{E}_1| \approx 1.3\%$, which is quite good. However, running the same ansatz with COBYLA, we obtain $E_{VQE} = -15.563$, which corresponds to an accuracy of $|\mathcal{E}_0 - E_{VQE}|/|\mathcal{E}_0 - \mathcal{E}_1| \approx 10\%$, and is a stark decrease. The convergence plot for the COBYLA run is shown in Fig.13.

For a 3x4 lattice and 7 layers of this ansatz, SLSQP gives a final VQE estimate of -22.01 and an accuracy of 6.45%, while COBYLA gives a final VQE estimate of -21.79 with an accuracy of 19.7%. Clearly, the ansatz performs worse as the lattice size increases. Even with many parameters (227 parameters for 7 layers), the optimisers cannot get the ansatz within an accuracy of 5%.

C. Summary

Table II lists the results we obtained from the two static ansätze we considered. We encounter issues with local minima as we increase the number of parameters, so increasing the number of gates (or layers) does not definitely improve our results with static ansätze. Thus, even the 7-layer Feulner-Hartmann ansatz, with 227 parameters, is unable to reach accuracies less than 5%. Furthermore, these issues also mean that we have to be incredibly careful with the choice of initial point for optimisation. One way to tackle this is to run

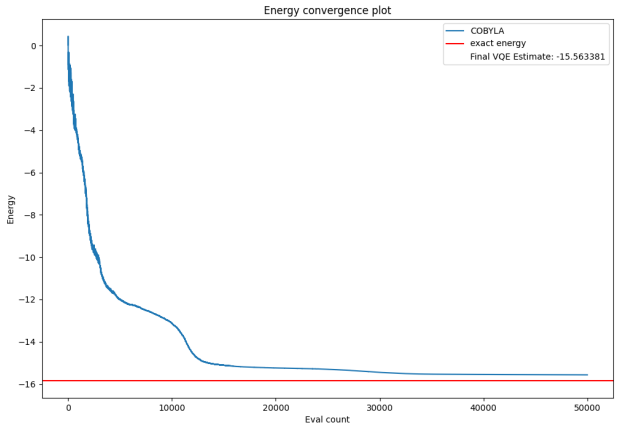

FIG. 13: Convergence plot for optimisation of 7 layers of the Feulner-Hartmann ansatz using COBYLA. This was for a 3x3 J_1-J_2 lattice with $J_2/J_1 = 0.5$. The exact ground state energy for this system, $\mathcal{E}_0 = -15.837$, is shown as a horizontal red line.

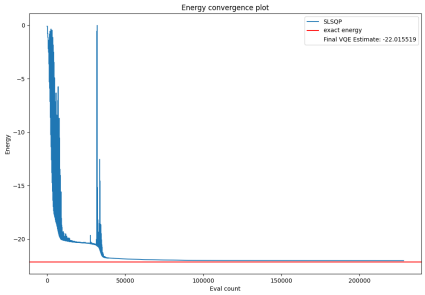

FIG. 14: Convergence plot for optimisation of 7 layers of the Feulner-Hartmann ansatz using SLSQP. This was for a 3x4 J_1-J_2 lattice with $J_2/J_1 = 0.5$. The exact ground state energy for this system, $\mathcal{E}_0 = -22.138$, is shown as a horizontal red line.

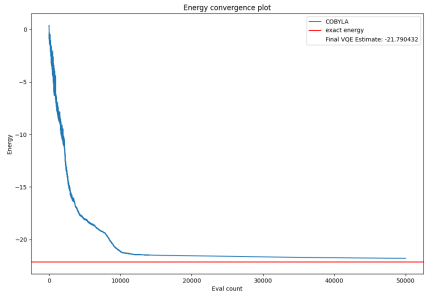

FIG. 15: Convergence plot for optimisation of 7 layers of the Feulner-Hartmann ansatz using COBYLA. This was for a 3x4 J_1-J_2 lattice with $J_2/J_1 = 0.5$. The exact ground state energy for this system, $\mathcal{E}_0 = -22.138$, is shown as a horizontal red line.

our VQE algorithm multiple times, starting at a different point each time, but this incurs a large additional cost to the executions.

Our main focus in the following sections is on improving our results as we increase the number of layers, and to try and mitigate the issue of local minima. Furthermore, we also question whether we really need so many gates (and parameters) and whether we can design an ansatz with fewer gates but produce similar results.

VI. DYNAMIC ANSATZ

To mitigate the problems mentioned in subsection V C, we investigated the usage of dynamic ansätze. We use the term static ansatz to refer to an ansatz which is predefined and remains the same for all iterations of the algorithm. Contrarily, the term dynamic ansatz refers to ansätze that change through the algorithm. We propose two different

3x3 Lattice				
Ansatz	#layers	Result	#params	$\frac{\lvert\mathcal{E}_0-E_{VQE}\rvert}{\lvert\mathcal{E}_0-\mathcal{E}_1\rvert}$
Feulner-Hartmann	7	-15.801	165	1.3%
Two-Local	17	-11.913	162	-
Two-Local	10	-15.412	99	15.45%
Exact Value: -15.837				

3x4 Lattice				
Ansatz	#layers	Result	#params	$\frac{\lvert\mathcal{E}_0-E_{VQE}\rvert}{\lvert\mathcal{E}_0-\mathcal{E}_1\rvert}$
Feulner-Hartmann	7	-21.790	227	17.56%
Feulner-Hartmann (result from [5])	7	-21.980	227	7.97%
Two-Local	18	-20.015	228	-
Exact Value: -22.138				

TABLE II: Comparison between the Feulner-Hartmann ansatz with 7 layers and the Two-Local ansatz with almost the same number of parameters for 3x3 and 3x4 lattices with $J_2/J_1 = 0.5$. For the 3x3 case, the Two-Local ansatz with an optimum number of parameters (10 layers) is given as well.

approaches of using dynamic ansätze to modify the Feulner-Hartmann ansatz [5]. The two approaches, which we call Sequential Layer Addition and Gradient Based Gate Addition, are discussed in the following.

A. Sequential Layer Addition

In Sequential Layer Addition we start with one layer of the ansatz and gradually deepen the circuit by the addition of one layer to the ansatz every t iterations until we reach a desired number of layers, n, by the end of the algorithm. The parameter t is chosen in a way that the expectation value of the Hamiltonian converges to the local minima in the current Hilbert space before adding another layer and moving to a larger Hilbert space. After adding the new layer, the parameters of the old layer(s) remain unchanged, so that the initial point of our new search in the larger Hilbert space is the (local) minimum found before adding the layer. In the case of the Feulner-Hartmann ansatz, we start with the initial parameterised rotation gates on each qubit and one layer of the repetition block of parameterised rotations and XXYYZZ-gates. Consequently, we add one layer of the repetition block to the end of the circuit every t iterations. An evolution of the Feulner-Hartmann ansatz using Sequential Layer Addition is illustrated in Figure 16.

Since using Sequential Layer Addition lets the algorithm converge in a smaller Hilbert space by changing a much smaller number of parameters compared to the case of the static ansatz, we hypothesised that this procedure would decrease the probability of the algorithm getting stuck in local minima while improving the result obtained using a smaller number of parameters in average. Moreover, by reducing the average depth of the circuit, this method

reduces the effect of noise when implemented on quantum hardware [20].

To the best of our knowledge, a similar approach of gradually increasing the number of layers of the ansatz has not been investigated for VQE algorithm, however, a similar approach was taken for Variational Quantum Linear Solver algorithm in Ref.[20].

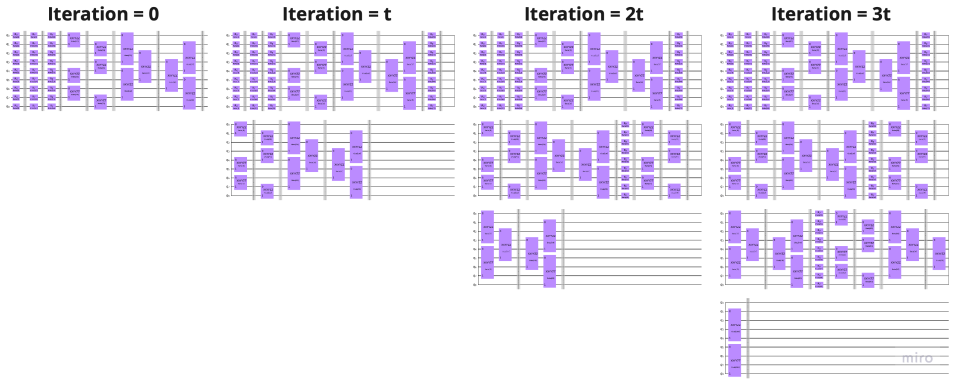

FIG. 16: The first $4t$ iterations of Sequential Layer Addition for 3x3 Feulner-Hartmann ansatz.

B. Gradient Based Gate Addition

In Gradient Based Gate Addition we start with k layers of the ansatz and gradually deepen the circuit by adding selected gates. The gates are selected based on the norm of the average changing rate of their parameters. After each t iterations of the algorithm, a fraction α of the initial number of gates for which the norm of the gradient of the gate parameter over the past t iterations is higher are selected and added to the right end of the ansatz. Subsequently, the parameters of these gates are optimised for the next t iterations. Similar to the previous approach, the parameter t is chosen in a way that the expectation value of the Hamiltonian converges to the local minima in the current Hilbert space before adding the new gates. Parameters k and α, however, were chosen empirically from the sets $\{1, 2, 3\}$ and $\{5, 10, 20\}$ respectively.

The evolution of the Feulner-Hartmann ansatz using Gradient Based Gate Addition is illustrated in Figure 17. In this case, k and α were taken to be 2 and 20% respectively, which meant that after each t iterations, 20% of the initial number of gates, were selected based on the changing rate of their parameters and added to the end of the ansatz.

Similar to the previous case, this approach uses a smaller average number of parameters and circuit depth with respect to the case of the static ansatz. Moreover, we wanted to investigate whether this method could be used for building a viable ansatz starting from the building blocks of the circuit.

This idea was loosely based on the ADAPT-VQE algorithm, in which the gradient of the expectation value of the Hamiltonian is calculated with respect to each gate parameter and the gate with the largest norm of the gradient is added to the left end of the ansatz in order to systematically grow the ansatz throughout the algorithm [21].

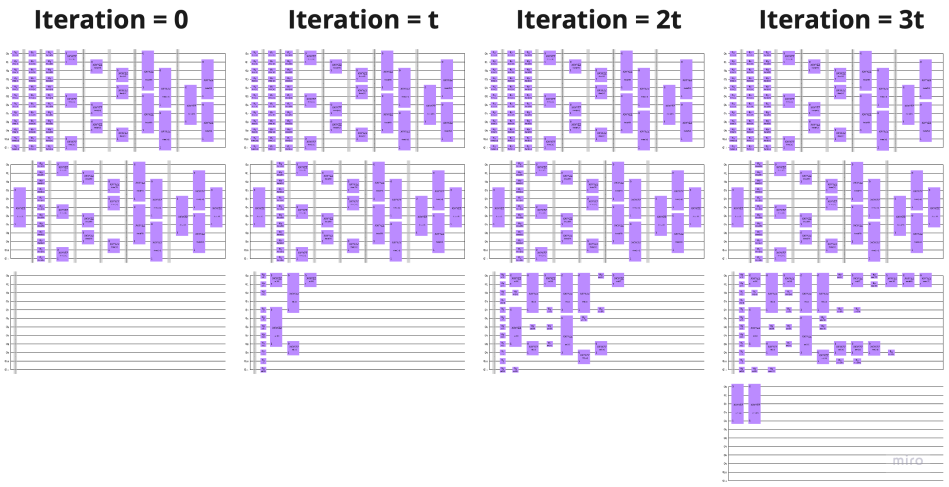

FIG. 17: The first $4t$ iterations of Gradient Based Gate Addition for 3x4 Feulner-Hartmann ansatz with $k = 2$ and $\alpha = 20\%$

VII. RESULTS

Four set of experiments were carried out in order to evaluate the performance of the two proposed approaches to using dynamic ansätze. The experiments were focused on comparing the achieved estimation of the ground state energy, the probability of the optimiser getting stuck in local minima, further comparison of the static ansatz and Sequential Layer Addition, and the comparison between the performance when using different classical optimisers. In all of these experiments, a static or dynamic version of the Feulner-Hartmann ansatz [5] was used for the 3x3 or 3x4 J_1-J_2 lattices with $J_2/J_1 = 0.5$. Moreover, as mentioned in sub-section III C, IBM's AerSimulator with 10^6 shots was used for all of these experiments.

A. Dynamic vs Static Ansatz for Better Estimations

The first set of experiments were focused on comparing the performance of the static ansatz and the two proposed approaches to dynamic ansätze in terms of the achieved estimation of the ground state energy. We investigated using a static Feulner-Hartmann ansatz as well as a dynamic ansatz using Sequential Layer Addition and Gradient Based Gate Addition separately for 3x3 and 3x4 lattices respectively. For the case of the static ansatz, the Feulner-Hartmann ansatz with 7 layers was used, whereas for the case of Sequential Layer Addition we started with a one-layer ansatz and iteratively added up to 7 layers to the ansatz. In empirically tuning k and α values for the case of Gradient Based Gate Addition, the best results were achieved when using $k = 2$ and $\alpha = 20\%$. The results of these experiments are illustrates in Table III. It should be noted that the initial parameters and initialisation seed of the classical optimiser were the same for these cases.

Both in the case of 3x3 and 3x4 lattices, the Sequential Layer Addition approach performed significantly better than the case of the static ansatz with a lower average number of parameters and circuit depth, as it can be seen in Table III. More specifically, when using

3x3 Lattice			
Ansatz	Result	Final #params	Avg. #params
Static Ansatz	-15.801	165	165
Sequential Layer Addition (Up to 7)	-15.836	165	102
Gradient Based Gate Addition ($k = 2$, $\alpha = 20\%$)	-15.645	144	102
Exact Value: -15.837			

3x4 Lattice			
Ansatz	Result	Final #params	Avg. #params
Static Ansatz	-21.790	227	227
Sequential Layer Addition (Up to 7)	-22.130	227	140
Gradient Based Gate Addition ($k = 2$, $\alpha = 20\%$)	-21.984	146	114
Exact Value: -22.138			

TABLE III: Comparison between static and dynamic ansatz using Sequential Layer Addition and Gradient Based Gate Addition with a 7 layer Feulner-Hartmann ansatz for 3x3 and 3x4 lattices with $J_2 = -0.5$ and $J_1 = -1$.

Sequential Layer Addition for the 3x4 lattice we get

$$\frac{|\mathcal{E}_0 - E_{VQE}|}{|\mathcal{E}_0 - \mathcal{E}_1|} \approx 2.1\%$$

As mentioned in sub-section V B, when using a static ansatz with 7 layers this value is approximately 19.7%.

Gradient Based Gate Addition, however, performed better than the static ansatz case only for the 3x4 lattice. It is worth noting that using Gradient Based Gate Addition resulted in an estimate of -21.984 using half of the number of parameters used in the case of the static ansatz, while the latter resulted in a slightly poorer estimate of -21.790.

B. Sequential Layer Addition for Local Minima

In the second set of experiments, the effectiveness of the Sequential Layer Addition method was evaluated in terms of how often the optimiser got stuck in local minima. To this end, we performed the algorithm for 30 different initialisations of the optimiser with Sequential Layer Addition as well as 30 different initialisations of the optimiser using a static ansatz. These experiments were performed for the 3x4 lattice using Feulner-Hartmann ansatz [5] and the COBYLA optimiser.

The results of this experiment can be found in Figure 18 and Table IV. It is evident

from the box-plot and the table that using Sequential Layer Addition performs significantly better than using a static ansatz both in terms of the mean and variance of the results. This means that using Sequential Layer Addition significantly improves the performance of the algorithm both in terms of the local minima problem as well as the accuracy of the estimated energy.

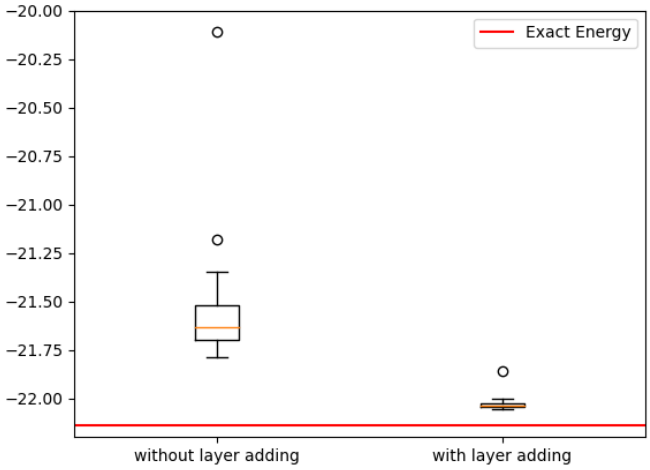

FIG. 18: Box plot of results of running the static ansatz and dynamic ansatz with Sequential Layer Addition. The optimiser COBYLA was used to obtain this data.

3x4 Lattice		
Ansatz	Mean	Std. Deviation
Static Ansatz	-21.560	0.301
Sequential Layer Addition (Up to 7)	-22.030	0.034
Exact Value: -22.138		

TABLE IV: Comparison between the mean and standard deviation of the results obtained by running the static and dynamic ansatz with Sequential Layer Adding 30 times each for a 3x4 lattice

To further analyse the significance of these results, a Welch's t-test was performed to compare these two distributions. We chose to perform Welch's t-test as the variance of the two distributions are significantly different. Moreover, it is appropriate to perform such a test as we have 30 samples for each distribution and this number is enough to use central limit theorem to assume Gaussian distributions for both cases and conduct a test of statistical significance. The result of the t-test, which demonstrated the significance of the result, is as follows:

> The 30 runs with the dynamic ansatz using Sequential Layer Adding ($M = -21.560, SD = 0.301$) compared to the 30 runs with the static ansatz ($M = -22.030, SD = 0.034$) demonstrated significantly better estimations of the

ground state energy with Sequential Layer Addition compared to using a static ansatz, $t(58) = -8.355$, $p = 2.689 \times 10^{-9}$.

C. Limits of Sequential Layer Addition

In the third set of experiments, we ran experiments using Sequential Layer Addition on the Feulner-Hartmann ansatz for a 3x4 lattice and reduced the number of final layers, n, in order to find the minimum n for which the Sequential Layer Addition method could still perform better than the static ansatz approach. The results of this experiment are demonstrated in Table V.

Table V shows the comparison between the results, final number of parameters and average number of parameters in Sequential Layers Addition for different values of n. It is evident from the table that when using Sequential Layer Addition for 3x4 lattice and adding layers up to 4 layer, with 96.5 parameters on average, we still got better results than using the static ansatz with 227 parameters.

Ansatz	#layers	Result	Final #params	Avg. #params
Layer Adding	Up to 7 Layers	-22.130	227	140
	Up to 6 Layers	-22.120	198	125.5
	Up to 5 Layers	-22.085	169	111
	Up to 4 Layers	-22.024	140	96.5
	Up to 3 Layers	-20.105	111	82
	Up to 2 Layers	-11.621	82	67.5
Static	7 Layers	-21.790	227	227
	4 Layers	-21.667	140	140
Exact Value: -22.138				

TABLE V: The comparison of results, final number of parameters and average number of parameters using Sequential Layer Adding with the Feulner-Hartmann ansatz for 3x4 lattice with different values of final number of layers.

D. Different Choices of Classical Optimisers

In this set of experiments, we compared the use of gradient-based and gradient-free optimisers by running the algorithm using a static ansatz and a dynamic ansatz with Sequential Layer Addition with SLSQP and COBYLA optimisers. For each of the optimisers, the algorithm was performed for 30 different initialisations of the classical optimiser with a static ansatz as well as 30 different initialisations of the optimiser with Sequential Layer Addition. In all cases, the experiment was performed for the 3x4 lattice using the Feulner-Hartmann ansatz [5]. The static ansatz runs of the algorithm were performed with 7 layers of the ansatz, whereas in Sequential Layer Addition the algorithm was performed starting with one layer and iteratively adding layers up to 7 layers to the ansatz. The results of these experiments are illustrated in Figure 19 and 20.

SLSQP is a gradient-based optimisation method that determines a local search direction by solving the second-order local approximation of the cost function that satisfies the constrains, whereas COBYLA uses linear approximation of the target and constraints functions to find the simplex that satisfies the constraints to perform optimisation [15]. Hence, we were expecting to get better results using COBYLA as the optimiser. Nevertheless, Figures 19 and 20 show the contrary to be true. Both the mean and the variance of data is lower in the case of using SLSQP as an optimiser. These results can be seen in Table VI. Moreover, the convergence plots for the cases of 3x3 and 3x4 lattices are illustrated in Figures 21 and 22 respectively.

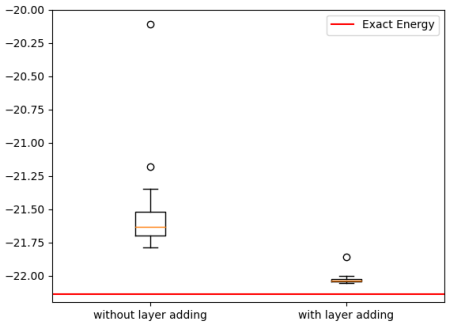

FIG. 19: Box Plot of the results using a static Feulner-Hartmann ansatz vs a dynamic one with Sequential Layer Addition for 3x4 lattice using COBYLA as the Optimizer.

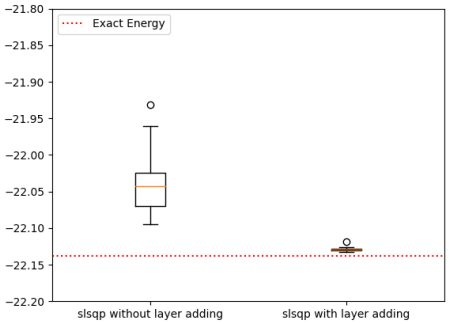

FIG. 20: Box Plot of the results using a static Feulner-Hartmann ansatz vs a dynamic one with Sequential Layer Addition for 3x4 lattice using SLSQP as the Optimizer. (Note that the scale of the y-axis of this figure is different from that of Figure 19.)

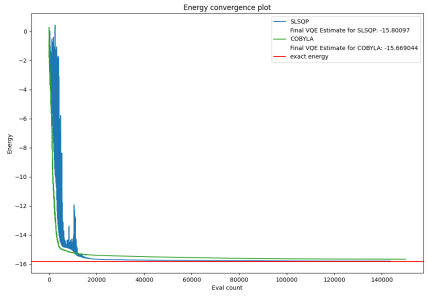

FIG. 21: The convergence plots of 3x3 Feulner-Hartmann ansatz using the COBYLA (green) and SLSQP (blue) optimisers with the same seed. When using SLSQP the estimated energy is -15.801 whereas using COBYLA estimates -15.669.

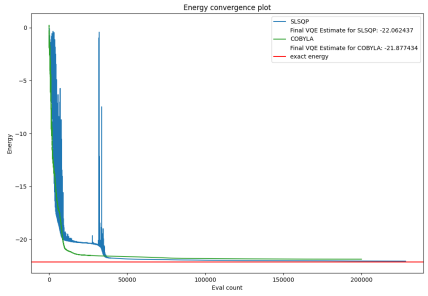

FIG. 22: The convergence plots of 3x4 Feulner-Hartmann ansatz using the COBYLA (green) and SLSQP (blue) optimisers with the same seed. When using SLSQP the estimated energy is -22.062 whereas using COBYLA estimates -21.877.

	Ansatz	Mean	Std. Deviation
COBYLA	Static Ansatz	-21.560	0.301
	Sequential Layer Addition (Up to 7)	-22.030	0.034
SLSQP	Static Ansatz	-22.039	0.041
	Sequential Layer Addition (Up to 7)	-22.129	0.002
Exact Value: -22.138			

TABLE VI: Comparison between the mean and standard deviation of the results obtained by using the COBYLA ans SLSQP optimisers for running the static and dynamic ansatz with Sequential Layer Adding 30 times each for a 3x4 lattice.

VIII. DISCUSSION

In this project, we investigated estimating the ground state energy of the J_1-J_2 model using the VQE algorithm. In the first part of the project, we ran experiments using two static ansätze, namely Two-Local and Feulner-Hartmann [5], on IBM's AerSimulator. The results of this part unveiled the problem of the optimiser getting stuck in local minima for deeper circuits with a larger number of parameters. This also implies that the initial starting point for optimisation is extremely important for these optimisations since starting at a bad initial point would get our optimiser stuck in local minima. We also believed that the large number of gates used in both ansätze was an issue, especially if we hope to run these on quantum hardware in the future. Moreover, we obtained better estimates of the energy using the SLSQP optimiser than the COBYLA optimiser, which defied our expectations that a gradient-free optimiser would be more suitable for this problem, given the presence of shot-noise. In order to mitigate this local minima issue as well as to decrease the number of required parameters, we proposed two approaches to using a dynamic ansatz.

The first approach, Sequential Layer Addition, which is similar to that of [20], proved to be a substantial improvement with respect to the case of the static ansatz by converging significantly better to the ground state energy as well as significantly reducing the variance of the results of the algorithm. This means that using this approach, the results were less dependent on the initial point of the optimiser. Moreover, these results are achieved by a smaller average number of parameters and a less deep circuit.

The second approach, Gradient Based Gate Addition, which is loosely based on the AdaptVQE algorithm [21], produced better results than the case of a static ansatz for the 3x4 lattice using a smaller average number of parameters and a less deep circuit. Nevertheless, for the 3x3 ansatz, we did not surpass the static ansatz results. A possible reason for this could be that we may have failed to tune the hyper-parameters of the algorithm, as this method requires tuning the values of k, the number of layers to start with, t, how often gates are added to the ansatz, and α, which determines the number of added gates. Despite this drawback, Gradient Based Gate Addition could potentially be a tool for iteratively building better ansätze starting from the building blocks of the ansatz in case the parameterised gates are entangling. This means that in the case of an ansatz like the Two-Local ansatz where the ansatz is made of (non-parameterised) C-Not gates and one-qubit rotations, this method can not be useful for growing the ansatz effectively.

A. Outlook

As future work, we are interested in experimenting with the two dynamic approaches on quantum hardware, or incorporating simulated noise on a simulator. Moreover, we would like to run more experiments with other values of J_2/J_1 as well as a periodic-boundary Hamiltonian, as we only considered the case of an open-boundary Hamiltonian and $J_2/J_1 = 0.5$. Furthermore, the Sequential Layer Addition approach could be applied to the TwoLocal ansatz as well, maybe by adding two layers every time we pause the optimiser.

Following the approaches to using dynamic ansätze, we would like to investigate a third approach, Gradient Based Gate Deletion, which is analogous to Gradient Based Gate Addition. In this approach, we start with n layers in the ansatz and iteratively remove a fraction β of the gates with the smallest average gradient of parameters every t iterations.

Intuitively, in Sequential Layer Adding, the optimiser can never perform worse after the addition of a new layer. At worst, it stays stuck in the same local minima, and given that the number of parameters added is large enough, the optimiser should be able to find a way out of the local minima. Contrarily, we have seen that with static ansätze, adding another layer may lead to the optimiser getting stuck in local minima more often. We believe that the stark performance improvement and variance reduction for the Sequential Layer Addition results (compared to static ansätze) can be attributed to this. This approach should be a very useful tool for other realistic VQE problems and should indeed be tested for other systems.

The Gradient Based Gate Addition results when we start with just one or two layers of a parent ansatz suggest that it does indeed help the optimiser, but its additions stop being as useful later in the optimisation. We think some combination of Gradient Based Gate Addition, and Gradient Based Gate Deletion may be able to tackle this and find a good configuration of the ansatz. Regardless, the fact that this approach constructed an ansatz (given just 2 initial layers) which gave results better than its static counterpart with half the number of parameters, proves that this approach has some merit to it.

REFERENCES

[1] [Online; accessed 29. Jan. 2023]. Jan. 2023. URL: https://qiskit.org/documentation/stubs/qiskit.circuit.library.TwoLocal.html?highlight=twolocal.

[2] Stephan G. J. Philips et al. "Universal control of a six-qubit quantum processor in silicon". In: *Nature* 609 (Sept. 2022), pp. 919–924. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05117-x.

[3] John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (Aug. 2018), p. 79. DOI: 10.22331/q-2018-08-06-79. eprint: 1801.00862v3.

[4] Yudong Cao et al. "Quantum Chemistry in the Age of Quantum Computing". In: *Chem. Rev.* 119.19 (Oct. 2019), pp. 10856–10915. ISSN: 0009-2665. DOI: 10.1021/acs.chemrev.8b00803.

[5] Verena Feulner and Michael J. Hartmann. "Variational quantum eigensolver ansatz for the J_1-J_2-model". In: *Phys. Rev. B* 106.14 (Oct. 2022), p. 144426. ISSN: 2469-9969. DOI: 10.1103/PhysRevB.106.144426.

[6] A. V. Mikheyenkov, A. V. Shvartsberg, and A. F. Barabanov. "Phase transitions in the 2D J 1-J 2 Heisenberg model with arbitrary signs of exchange interactions". In: *JETP Lett.* 98.3 (Oct. 2013), pp. 156–160. ISSN: 1090-6487. DOI: 10.1134/S0021364013160121.

[7] Dietrich Roscher et al. "Cluster functional renormalization group and absence of a bilinear spin liquid in the J_1-J_2 Heisenberg model". In: *Phys. Rev. B* 100.12 (Sept. 2019), p. 125130. ISSN: 2469-9969. DOI: 10.1103/PhysRevB.100.125130.

[8] Sam McArdle et al. “Quantum computational chemistry”. In: *Rev. Mod. Phys.* 92.1 (Mar. 2020), p. 015003. ISSN: 1539-0756. DOI: 10.1103/RevModPhys.92.015003.

[9] Jules Tilly et al. “The Variational Quantum Eigensolver: A review of methods and best practices”. In: *Phys. Rep.* 986 (Nov. 2022), pp. 1–128. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2022.08.003.

[10] Dmitry A. Fedorov et al. “VQE method: a short survey and recent developments”. In: *Mater. Theory* 6.1 (Dec. 2022), pp. 1–21. ISSN: 2509-8012. DOI: 10.1186/s41313-021-00032-6.

[11] Aidan Pellow-Jarman et al. “A comparison of various classical optimizers for a variational quantum linear solver”. In: *Quantum Inf. Process.* 20.6 (June 2021), pp. 1–14. ISSN: 1573-1332. DOI: 10.1007/s11128-021-03140-x.

[12] Jelmer M. Boter et al. “Spiderweb Array: A Sparse Spin-Qubit Array”. In: *Phys. Rev. Appl.* 18.2 (Aug. 2022), p. 024053. ISSN: 2331-7019. DOI: 10.1103/PhysRevApplied.18.024053.

[13] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988. URL: https://books.google.nl/books?id=4rKaGwAACAAJ.

[14] M. J. D. Powell. “Direct search algorithms for optimization calculations”. In: *Acta Numerica* 7 (Jan. 1998), pp. 287–336. DOI: 10.1017/s0962492900002841. URL: https://doi.org/10.1017/s0962492900002841.

[15] Xavier Bonet-Monroig et al. *Performance comparison of optimization methods on variational quantum algorithms*. 2021. DOI: 10.48550/ARXIV.2111.13454. URL: https://arxiv.org/abs/2111.13454.

[16] James Spall. “An Overview of the Simultaneous Perturbation Method for Efficient Optimization”. In: (Feb. 2001).

[17] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[18] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. “On the Convergence of Adam and Beyond”. In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=ryQu7f-RZ.

[19] Manpreet Singh Jattana et al. “Improved variational quantum eigensolver via quasi-dynamical evolution”. In: *arXiv* (Feb. 2022). DOI: 10.48550/arXiv.2202.10130. eprint: 2202.10130.

[20] Hrushikesh Patil, Yulun Wang, and Predrag S. Krstić. “Variational quantum linear solver with a dynamic ansatz”. In: *Physical Review A* 105.1 (Jan. 2022). DOI: 10.1103/physreva.105.012423. URL: https://doi.org/10.1103%2Fphysreva.105.012423.

[21] Harper R. Grimsley et al. “An adaptive variational algorithm for exact molecular simulations on a quantum computer”. In: *Nature Communications* 10.1 (July 2019). DOI: 10.1038/s41467-019-10988-2. URL: https://doi.org/10.1038/s41467-019-10988-2.

Appendix A: Circuit Built by Gradient Based Gate Addition

The complete evolution of the Feulner-Hartmann ansatz using Gradient Based Gate Addition starting from 2 layers with $\alpha = 20\%$ can be seen in Figure 23.

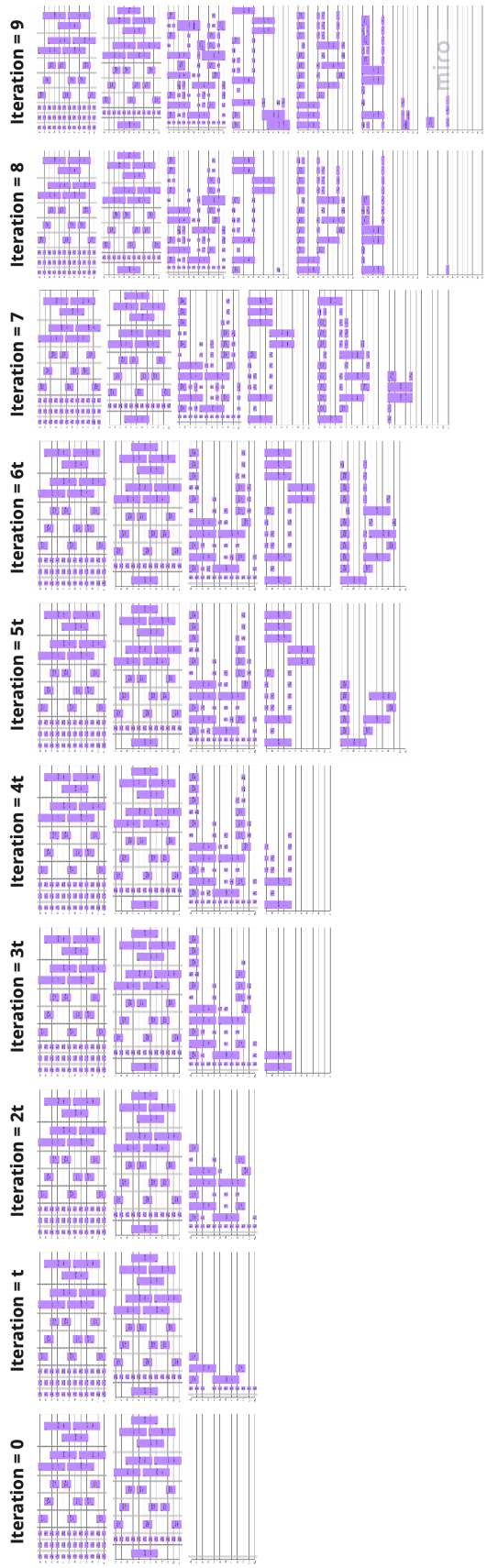

FIG. 23: The evolution of the Feulner-Hartmann ansatz using Gradient Based Gate Addition for a 3x4 lattice with $k = 2$ and $\alpha = 20\%$

Appendix B: COBLYA Convergence Graphs

In order to visualise and compare the convergence of the algorithm in the case of static and dynamic ansätze, two convergence graphs are included in this appendix. Figure 24 corresponds to running the VQE algorithm with a static ansatz, whereas Figure 25 corresponds to running the algorithm with Sequential Layer Addition. In both cases the Feulner-Hartmann ansatz was used for a 3x4 lattice using COBYLA as the classical optimiser. Moreover, the initialisation seed of the optimiser is the same in both cases.

The peaks in Figure 25 correspond to the initial stages of the optimisation problem after a new layer containing parametrised gates has been added to the circuit. It can be seen from the two graphs that despite these frequent initial setbacks in the optimisation process, the final result to which the algorithm converges is closer to the exact value in the case of using Sequential Layer Addition.

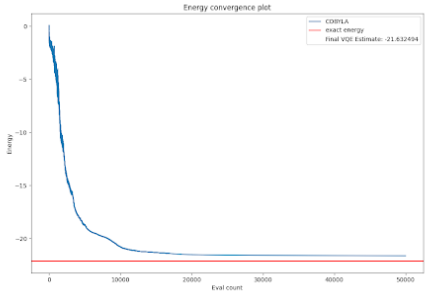

FIG. 24: The convergence graph of running the algorithm with a static Feulner-Hartmann ansatz for a 3x4 lattice.

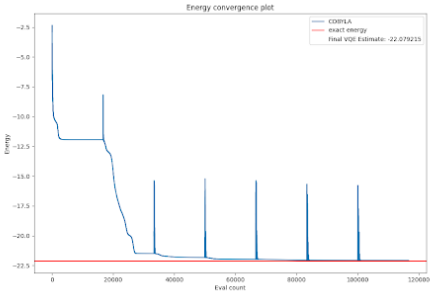

FIG. 25: The convergence graph of running the algorithm with a dynamic ansatz using Sequential Layer Addition for a 3x4 lattice.

In order to ensure that the mentioned peaks in Figure 25 were in fact natural after a layer being added, we ran two additional experiments. First, we interrupted the algorithm every t iterations, as done in the case of Sequential Layer Addition, and purposefully randomised the parameter values every time we interrupted the optimiser. We expected that this would prevent the optimiser from achieving convergence, and this is exactly what we observed in Figure 26.

Next, we wanted to ensure that the jumps were not happening due to a mistake in the process of adding a layer, but rather due to some reason inherent to the functioning of the optimiser itself and the act of pausing and continuing the optimiser. For this, we interrupted the optimiser every t iterations as we did for sequential layer addition, but instead of adding a new layer or randomising the parameters, we simply allowed the optimiser to continue. If our hypothesis that these small jumps were due to the pausing and resuming of the optimiser was correct, we would observe peaks similar to Figure 25, and the optimiser would still converge to some value close to the actual ground state energy, however this value would be less accurate than that achieved in the case of Sequential Layer Addition. This behaviour was confirmed in Figure 27.

It should be noted that when interrupting the optimiser, we create a new instance of the optimiser which has to perform the initial stages of linearly approximating the target function before starting to converge. This explains the peaks in Figure 27.

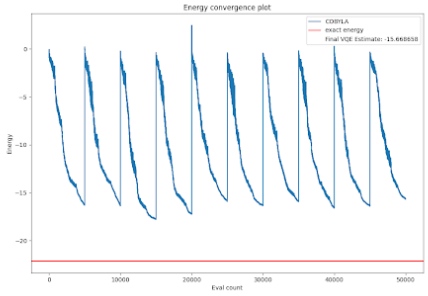

FIG. 26: The convergence graph of running the algorithm with a static Feulner-Hartmann ansatz for a 3x4 lattice and interrupting the optimiser every t iterations and reinitialising the starting point of the optimiser each time.

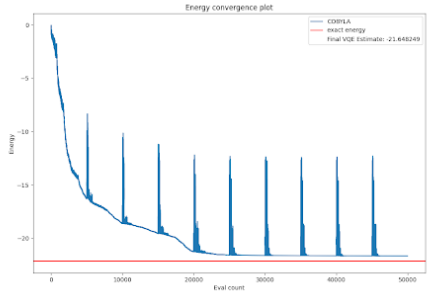

FIG. 27: The convergence graph of running the algorithm with a static Feulner-Hartmann ansatz for a 3x4 lattice and interrupting the optimiser every t iterations.

Appendix C: Outlier in Sequential Layer Addition - COBYLA

In Fig. 18, we notice a single outlier for the sequential layer addition, which was at a final VQE estimate of -21.85, which corresponds to an accuracy of 14.5%. Upon closer inspection, it seems as though the optimiser was stopped before it converged. This convergence plot is shown in Fig. 28. For comparison, Fig. 30 shows the convergence for another run for the same system (which was also used to make the box plot), and clearly, the optimiser in this case was allowed to converge. To test whether this was actually an outlier, or

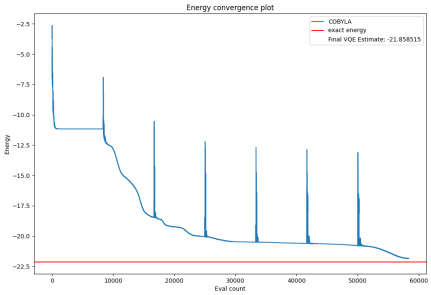

FIG. 28: Convergence plot when COBYLA is used to optimise the Sequential Layer Addition for Feulner-Hartmann ansatz, starting from 1 layer up to 7 layers.

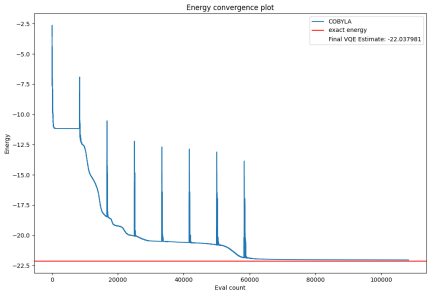

FIG. 29: Convergence plot when COBYLA is used to optimise the Sequential Layer Addition for Feulner-Hartmann ansatz, starting from 1 layer up to 7 layers. Same as Fig.28, but the optimiser was allowed to completely converge.

if the optimiser was simply not given enough iterations to converge, we ran the exact same dynamic ansatz program with the same initial parameters, but this time allowed the optimiser to completely converge. This is shown in Fig. 29. This run had a final VQE estimate of -22.04, corresponding to an accuracy of 5%. That is, this wasn't really an issue of the optimiser getting stuck in local minima, but rather of not allowing the optimiser enough iterations to converge. Regardless, we used the original results for the comparison. We also checked the other runs, but none of the others faced a similar cut-off issue.

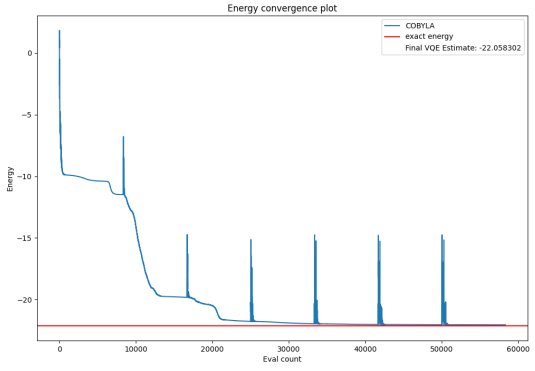

FIG. 30: Convergence plot when COBYLA is used to optimise the Sequential Layer Addition for Feulner-Hartmann ansatz, starting from 1 layer up to 7 layers. Same as Fig.28, but with different initial values for the parameters.

Appendix D: Code and Data

The code is available at the following Github repository:
https://github.com/niyoushanajmaei/VQE_for_J1J2

The data used for the statistical analyses in Subsections VII B and VII C is available in the following directory of the repository:
https://github.com/niyoushanajmaei/VQE_for_J1J2/tree/main/stat_data