IBM Open Science Prize Submission

Jaewon Jung, Jinwoo Yu, Junho Lim, Hokyun Kim

April 15, 2023

We present supplemental material for our submission of the IBM open science project. In our study, we applied a Hamiltonian Variational Ansatz (HVA) to simulate the Kagome lattice using the 16-qubit IBM Quantum Experience device named *ibmq_guadalupe*. Our algorithm utilizes a novel approach for applying unitary gates to real quantum hardware and applies various error mitigation techniques, particularly Zero Noise Extrapolation (ZNE). After error mitigation, we achieved optimization results within a 1% margin, with the best result scoring a relative error rate of 0.12%

1 Background

The Hamiltonian for a spin-1/2 Kagome lattice is a mathematical representation of the total energy of the system. The physical implications of the Hamiltonian are significant, as it describes the behavior and properties of the spin-1/2 Kagome lattice system [1]. By determining the ground state energy and corresponding wave function, one can explore the intricate mechanisms behind emergent phenomena.

To achieve the ground state energy of the Kagome lattice we implement the Variational Quantum Eigensolver (VQE) [5]. The VQE algorithm has two main parts: (1) preparing a trial wave function for the system using a parameterized quantum circuit, known as an ansatz, and (2) optimizing the adjustable parameters within the ansatz to minimize the system's energy. The optimization process is conducted via classic computation to find the optimal values of the parameters in the ansatz that minimize the energy of the system, making the entire process hybrid between classic and quantum.

However, there is no guarantee for the VQE that a given optimization process will lead to the desired ground state values [5]. In most cases, there exists a trade-off relationship between the expressibility of the ansatz and whether it's computationally feasible at all. Moreover, other issues such as the barren plateau problem [3] might occur, where the stochastic optimization process is rendered almost irrelevant. Previous research suggests that the VQE algorithm heavily relies on its initial parameters and assumptions, making it crucial to design a custom ansatz tailored to fit the system in question.

2 Ansatz

2.1 Trotterization

Trotterization is a technique used in quantum computing to approximate the time evolution of a quantum system when the Hamiltonian consists of non-commuting terms. The technique is based on the Lie product formula and the first-order Trotter-Suzuki decomposition, which are mathematical tools that break down the time evolution operator into a product of more manageable operators. The Lie product formula is an expression that relates the exponential of the sum of two operators A and B to the product of their exponentials. It is given by:

$$\exp(\lambda(A+B)) = \lim_{n\to\infty}[\exp(\lambda A/n)\exp(\lambda B/n)]^n$$

Here, λ is a scalar parameter representing time, and n is the number of times the product of exponentials is repeated. As n goes to infinity, the product of exponential converges to the exponential of the sum of the two operators.

The first-order Trotter-Suzuki decomposition is a practical approximation that arises from the Lie product formula when we truncate the limit to a finite value of n:

$$\exp(\lambda(A+B)) \approx [\exp(\lambda A/n)\exp(\lambda B/n)]^n$$

In the context of quantum computing and simulating the time evolution of quantum systems, the Hamiltonian H is often composed of several non-commuting terms, $H = H_1 + H_2 + ... + H_m$. The time evolution operator for the system is given by:

$$U(\lambda) = \exp(-i\lambda H)$$

Trotterization involves applying the first-order Trotter-Suzuki decomposition to approximate the time evolution operator as a sequence of individual time evolutions of each Hamiltonian term:

$$U(\lambda) \approx [\exp(-i\lambda H_1/n)\exp(-i\lambda H_2/n)...\exp(-i\lambda H_m/n)]^n$$

The approximation becomes more accurate as n increases, but at the cost of a higher number of quantum gates required to implement the time evolution.

2.2 Hamiltonian Variational Ansatz (HVA)

The Kagome antiferromagnetic Heisenberg Model H is denoted as follows, where $\langle i,j\rangle$ represents each pair of spin-interacting lattice sites.

$$H = \sum_{\langle i,j\rangle} \vec{\mathbf{S}}_i \cdot \vec{\mathbf{S}}_j = \sum_{\langle i,j\rangle} X_iX_j + Y_iY_j + Z_iZ_j$$

As we estimate the ground state of a Heisenberg spin-1/2 model on a kagome lattice, we use the Hamiltonian variational ansatz proposed by [7]. The Hamiltonian variational ansatz is based on the adiabatic theorem of quantum mechanics where an initial Hamiltonian H_0 evolves under a time-dependent Hamiltonian $H(t)$ set to $H(t) = (1 - \frac{t}{T})H_0 + \frac{t}{T}H$. As the terms for the Hamiltonian H are non-commutable, we trotterize $e^{-iH_0\Delta t}$ and $e^{-iH\Delta t}$. This returns the unitary circuit

$$U(\theta) = \prod_{i=1}^{p} e^{-i\theta_i H_0} e^{-i\theta_i H}$$

.

In the previous section, we have mentioned the first-order Trotter-Suzuki decomposition approximates the time evolution operator as a sequence of *individual* time evolutions. Therefore, the same circuit as above can be denoted as the unitary

$$U(\theta) = \prod_{i=1}^{p} \left[e^{-i\theta_i H_0} \prod_{k,l} e^{-i\theta_{i,kl}, S_k \cdot S_l} \right]$$

,

where each parameter corresponds to an edge in the Kagome lattice. Following this circuit, each parameter can be optimized separately regardless of the actual temporal order of node connection.

2.3 Optimization

We denote the initial Hamiltonian H_0, where its ground state $|\psi_i\rangle$ consists of the product of $\frac{1}{\sqrt{2}}[|\downarrow\uparrow\rangle - |\uparrow\downarrow\rangle]$ for all initially connected singlets, which in our model is assigned to 6 qubit pairs.

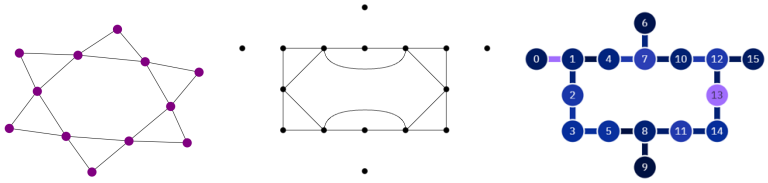
$$HEIS_{jk}(t) = e^{-it(X_k X_j + Y_k Y_j + Z_k Z_j)}$$

The unitary HEIS gates are employed as two nodes j and k are connected together. θ is parameterized for each connection. Under the premise of each connection being time-independent, θ is able to optimize rigorously.

However, the available quantum hardware only permits connection between adjacent qubits. Therefore, it is imperative to apply appropriate SWAP gates between HEIS gates to allow lattice connections between nodes physically non-adjacent in real hardware.

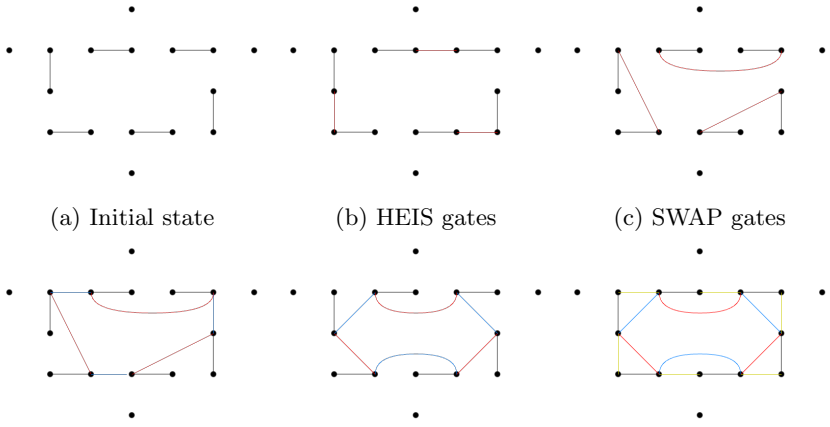
To achieve this, we implement a novel approach to connect qubits as depicted in Figure 1. Our goal is to make our qubit connections isomorphic with Figure 1(a), which would result in the qubit configuration shown in Figure 1(b). The numbering of each qubit node is shown in Figure 1(c).

To begin, we create the initial state [1], in which sets of singlets are connected as $\frac{1}{\sqrt{2}}[|\downarrow\uparrow\rangle - |\uparrow\downarrow\rangle]$. We then we apply the HEIS gates to the connections depicted as red in Figure 2(b). Applying SWAP gates changes the structure to



(a) Kagome lattice (b) Desired structure (c) Qubit numbering

Figure 1: Structure of the Kagome lattice and Quantum hardware



(a) Initial state (b) HEIS gates (c) SWAP gates

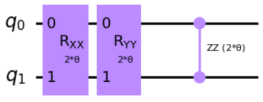
(d) HEIS gates (e) SWAP gates (f) HEIS gates

Figure 2: Illustration of applying unitary gates

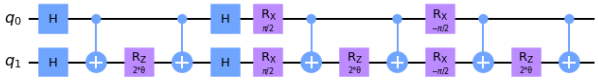
that shown in Figure 2(c). We then apply HEIS gates to the connections depicted in blue, resulting in Figure 2(d). After applying the SWAP gates once more we achieve the structure shown in Figure 2(e), Finally, the last application of HEIS gates, illustrated in yellow, results in the desired structure shown in Figure 1(b). The resulting gates are isomorphic to the original Kagome lattice structure. The above process is iterated per number of trotter steps.

2.4 Reducing CNOT gates

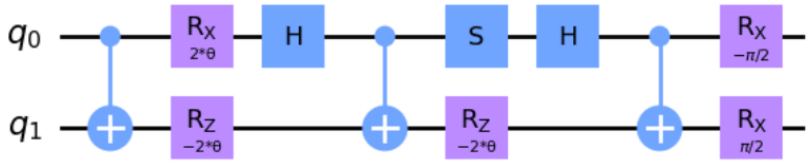
During the implementation of individual connections in the Hamiltonian Variational ansatz, we were able to reduce the number CNOT gates required from 6 to 3 by following the qubit architecture published by [6]. Figure 3(a) represents the original HEIS gate containing Rxx, Ryy, and Rzz gates, which includes 6 CNOT gates. In comparison, the circuit in Figure 3(c) is equivalent to the original circuit in Figure 3(a) while containing only 3 CNOT gates. It is known that utilizing fewer CNOT gates are expected to return better results as doing so implies less margin for quantum error to occur.



(a) Original HEIS gate



(b) Decomposition of Original HEIS gate



(c) Optimized HEIS gate with 3 CNOTs

Figure 3: Optimization of HEIS gate

3 Error Mitigation

3.1 Qubit Layout

When constructing the ansatz, we ensured that it was already aligned with the machine's physical layout as used during transpiling, with the qubits numbered as [0, 1, 2, 3, 4, ..., 15]. We did not utilize the qubits 0, 6, 9, and 15 in our algorithm due to poor connectivity.

3.2 Scheduling Method

We selected ALAP (As Late As Possible) as our preferred method for circuit scheduling. This approach reduces idling time and allows more time for qubits to reset at the beginning, as shown in Figure 4. By doing so, we were able to significantly reduce decoherence errors. The later idling time was only 160dt, which is about the time for single qubit gate, making it a fairly optimized ansatz. However, due to technical issues, we were unable to select ALAP as our scheduling method for real hardware. Instead, we transpiled the circuit without specifying a scheduling method.

3.3 Zero Noise Extrapolation

[4]. To mitigate errors, we utilized Zero Noise Extrapolation (ZNE), which involves intentionally amplifying noise and then extrapolating the results to estimate the zero-noise scenario. We achieved this by applying a CX noise amplifier that only amplifies CNOT gates, and then using linear extrapolation.

3.4 Qiskit runtime

The Qiskit Runtime platform provides a way to offload quantum computations to a cloud-based environment that is optimized for running quantum

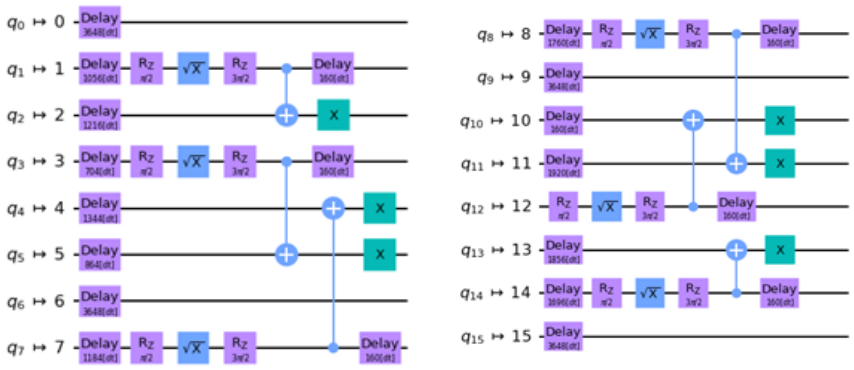


Figure 4: Ansatz with ALAP scheduling

programs [2]. One of its key features is the ability to support pre-compiled quantum circuits, which are circuits that have been optimized and compiled ahead of time. This allows users to avoid the overhead associated with compiling circuits on the fly, which can be time-consuming and resource-intensive.

The key hyperparameters utilized in Qiskit Runtime are the resilience level and optimization level. We selected a resilience level of 1 which supports Twirled Readout Error eXtinction(TREX) because we used a custom ZNE estimator instead of the default ZNE estimator provided in resilience level 2. Additionally, we chose an optimization level of 3 to achieve higher performance via quantum circuits, while supporting error suppression using Dynamic Decoupling.

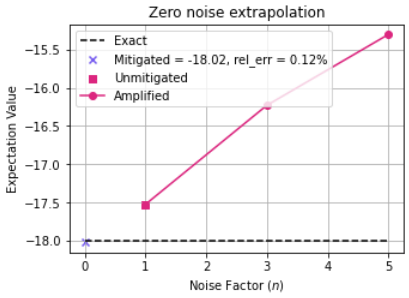
Zne strategy
Noise factors = (1,3,5)
Noise amplifier = CxAmplifier
Extrapolator = LinearExtrapolator

Table 1: Zne strategy for Error Mitigation

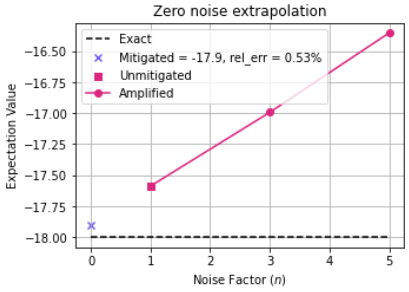
Hyperparameters
Runtime option: (shots = 4000)
Resilience level = 1
Optimization level = 3
Approximation degree = 1
Routing method = none

Table 2: Qiskit-runtime Hyperparameters for Error Mitigation

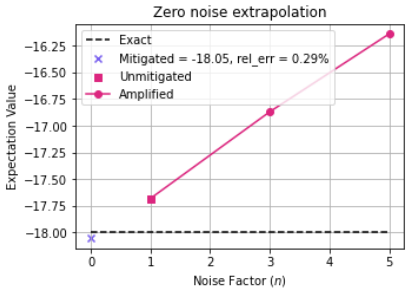
4 Results



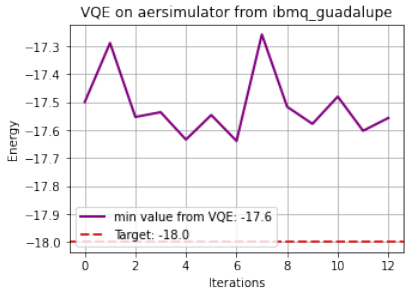
(a) Relative Error: 0.12%



(b) Relative Error: 0.53%



(c) Relative Error: 0.29%



(d) VQE results from noisy simulator (optimizer : COBYLA, rhobeg = 1/8)

Figure 5: (a), (b), (c): Results from *ibmq_guadalupe* after error mitigation via Zero Noise Extrapolation and their respective relative error values. (d): Results from AerSimulator

Prior to applying the Hamiltonian Variational ansatz, we found that the ground state energy could be exactly calculated by running the estimator with an initial state composed of a product of singlet states. After parameterizing the X gate of this initial circuit with RX gate and running it on AerSimulator (Figure 5(d)), it was concluded from comparing the parameter info (saved parameters from training) obtained from the callback function and the measured energy values that the X gate could be used as is (setting the parameter as simply π).

The results we report are achieved from *ibmq_guadalupe* All of the error mitigation techniques mentioned above, particularly ZNE were implemented. As shown in Figures 5(a), 5(b), and 5(c), we were able to achieve relative error weights within 1%.

5 Discussion

In the case of a single Kagome lattice, the product of singlet states was the ground state, and allowing the HVA to accurately calculate the ground state energy with a relatively short depth (=6) and duration time(= 3648dt $\approx$ 810.7 ns). However, as the lattice structure becomes larger and more complicated, the ground state cannot be obtained solely by singlet states. As a result, we have implemented our novel approach of applying HEIS and SWAP gates to perform trotterization on real quantum hardware.

Our work faced some technical limitations in implementing ZNE, as the current state of transpilation performs noise amplification before transpiling, which can lead to jumbled results when running on real hardware. However, we anticipate that once this issue is resolved, the transpilation process will be better able to reduce randomness than the current status quo. Additionally, we were unable to use the scheduling method ALAP during transpiling, as instruction of Delay time is not present in the current hardware although the *ibmq_guadalupe* machine does has Delay operation along with the basis gates. We expect that ALAP would enhance our algorithm's performance substantially, as when run on an AerSimulater, ALAP resulted in obtaining ground state energy with less than 1% error with a probability of approximately 38%, while running the simulator without ALAP resulted in obtaining ground state energy with less than 1% error only with a probability of approximately 18%.

However, there are concerns regarding error mitigation, specifically related to the use of ZNE and scalability. In some cases, applying ZNE can result in worse error rates than those obtained before error mitigation, indicating its limitations and making it unreliable for consistent performance improvement. A possible alternative strategy could involve running VQE without ZNE and subsequently using the obtained parameters for error mitigation. However, obtaining accurate parameters in the presence of errors and ensuring stable convergence may be challenging at the current stage of research. This brings us to the next issue of scalability. Increasing the trotter step is usually tolerable in noiseless systems, but it may not be feasible in real hardware due to the difficulty of distinguishing the gap between the ground state and excited states because of the added errors. While we achieved sufficient performance levels for the ground state of the given Kagome lattice without addressing this issue, scalability may turn out to be problematic for other more complex structures when large trotter steps are required. Thus, it may be better to obtain parameters from a noiseless simulator and then run them on real hardware, unless the latter has excellent error mitigation capabilities. We leave it to future research to address these challenges.

References

[1] Jan Lukas Bosse and Ashley Montanaro. "Probing ground-state properties of the kagome antiferromagnetic Heisenberg model using the variational quantum eigensolver". In: *Physical Review B* 105.9 (2022), p. 094409.

[2] Blake Johnson. "Qiskit runtime, a quantum-classical execution platform for cloud-accessible quantum computers". In: *APS March Meeting Abstracts*. Vol. 2022. 2022, T28–002.

[3] Jarrod R McClean et al. "Barren plateaus in quantum neural network training landscapes". In: *Nature communications* 9.1 (2018), p. 4812.

[4] Pedro Rivero et al. *Zero Noise Extrapolation prototype*. Version 1.0.0. 2023. URL: `https://github.com/qiskit-community/prototype-zne`.

[5] Jules Tilly et al. "The variational quantum eigensolver: a review of methods and best practices". In: *Physics Reports* 986 (2022), pp. 1–128.

[6] Rajat Vishwakarma. "Quantum Simulation and Dynamics". In: (2022).

[7] Dave Wecker, Matthew B Hastings, and Matthias Troyer. "Progress towards practical quantum variational algorithms". In: *Physical Review A* 92.4 (2015), p. 042303.