Accelerating Noisy Algorithm Research with PennyLane-Lightning and NVIDIA cuQuantum SDK

Lion Frangoulis[1*], Cristian Emiliano Godinez Ramirez[1*], Emily Haworth[1*], Aaron Sander[1*]
[1] *Technical University Munich*
[*] *All authors contributed equally*

(Dated: February 28, 2023)

In this project, we consider the challenges of simulating noisy quantum algorithms, which are known to require significant computational resources. We address this issue by leveraging the GPU tools available in Xanadu's PennyLane-Lightning-GPU and NVIDIA's cuQuantum SDK, which enable us to scale up our simulations and gain deeper insights into the impact of noise on quantum algorithms. Our analysis sheds light on the general effects of noise on simulation and identifies areas where it can accelerate the simulation of open quantum systems and ground state optimization. Through our work, we hope to contribute to a better understanding of how to effectively simulate noisy quantum algorithms, which could have far-reaching implications for quantum computing and finding NISQ era use cases.

I. INTRODUCTION

The delicate nature of quantum computers means they are inherently prone to noise and errors that can limit their usefulness for practical applications. To overcome this, researchers are investigating various mitigation techniques, including algorithms that can tolerate noise and hardware components that are more resilient to noise.
Noisy simulations play a critical role in this research by providing a way to simulate the behavior of quantum systems under noisy conditions, which can help researchers to better understand and characterize the sources of noise, its influence in the quantum systems under study, and develop strategies to mitigate its effects. Meanwhile, performing classical simulation of quantum systems is limited by the amount of resources needed, such as computational cost and required memory, however, it allows for controlled introduction of noise. These simulations can be enhanced through the use of GPUs, which offer parallel processing power.

In this project for QuHack 2023 we investigate the classical simulation of noisy quantum systems, to probe the advantage of GPU-aided simulations over traditional methods that employ CPU and density matrices, as well as the effect of noise in quantum algorithms. We investigate applications where noise can have an advantageous influence, thus, providing a use case for NISQ-era QPUs. We achieve this using the open-source Python library PennyLane-Lightning [1], which offloads computations to NVIDIA's cuQuantum [2] backend, to create quantum circuits that simulate the Heisenberg Hamiltonian model with stochastically applied noise. We utilize an NVIDIA A100 GPU provided by the Cyxtera/run:ai cluster. We show an advantage in noise-assisted algorithms which can lead to faster ground state optimization and open-quantum system simulations.

Throughout this project, we use a ubiquitous method for the simulation of the Hamiltonian evolution implemented as a quantum circuit called the Trotter-Suzuki decomposition. As quantum gates can only be represented by unitary matrices, the goal is to split the Hamiltonian such that our quantum circuit approximates its evolution. This approximation is then achieved as follows:

$$U(t) \approx \prod_{d=1}^{D} \prod_{n=1}^{N} \exp\left(-iH_n t/D\right) \tag{1}$$

Where the individual n terms do not necessarily commute, and D represents the number of Trotter steps, or depth as will be used in our simulations.

The model used in each section is the Transverse-Field Heisenberg model described by

$$\begin{split} H = -\sum_{i}^{n} [J_{\mathrm{x}} \cdot (S_i^x \otimes S_{i+1}^x) + J_{\mathrm{y}} \cdot (S_i^y \otimes S_{i+1}^y) + \\ J_{\mathrm{z}} \cdot (S_i^z \otimes S_{i+1}^z) + h \cdot S_i^z]. \end{split} \tag{2}$$

Applying the Trotter decomposition to our Heisenberg model, we can decompose Eq. 2 as the consecutive application of quantum gates. The quantum circuit simulating the noiseless evolution of this model is shown in Fig. 1.

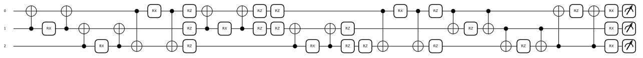

FIG. 1. Quantum circuit for one Trotter step noiseless simulation of the Heisenberg Hamiltonian applied to three qubits.

II. SCALING OF GPU STATEVECTOR SIMULATIONS

In the context of this project, we have analyzed the runtime complexity of the CPU and GPU on the cluster available to us. In order to scale statevector simulations, it is necessary to understand the limits of the hardware we have available. In particular, it is necessary to see which system sizes (i.e. circuit width or number of qubits) lead to exponential growth in computational time. The memory complexity is not a limiting factor in determining this as the runtime growth becomes exponential before the memory limit of the hardware. To investigate this, we tested the computational time of a First-Order Trotter-Suzuki Decomposition [3] of the Heisenberg model found in Eq. 2 with parameters $J_x = 1, J_y = 2, J_z = 1, h = 0.5$ up to an elapsed time $T = 1$ with 100 Trotter steps.

The exponential scaling of the statevector simulation should however be dependent only on the number of qubits and not the algorithm itself.

The results of the algorithm scaling can be seen in Fig. 2. It can be seen that the CPU begins to scale exponentially at approximately 20 qubits, while the GPU can be pushed further to roughly 25 qubits. This justifies the usage of a GPU for larger statevector simulations although exponential scaling does still occur eventually.

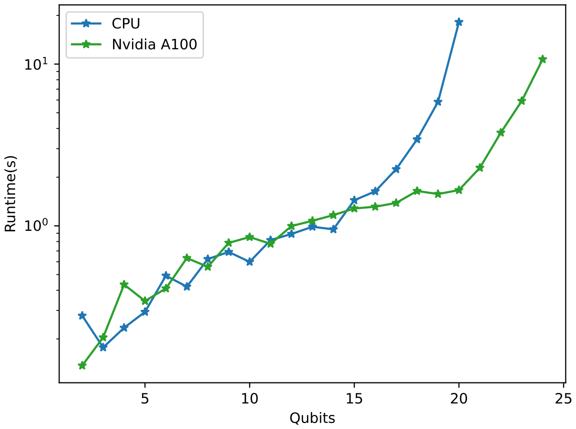

FIG. 2. Scaling of statevector simulator against number of qubits on Cyxtera/run:ai cluster.

III. EFFECT OF NOISE ON QUANTUM SYSTEMS

To properly use noise in applications, it is important to understand its fundamental effects on quantum systems. In this section, we focus on depolarizing noise [4]. The Kraus operators corresponding to the depolarizing channel are

$$K_1 = \sqrt{1-p}I_2, K_2 = \sqrt{\frac{p}{3}}X, \\ K_3 = \sqrt{\frac{p}{3}}Y, K_4 = \sqrt{\frac{p}{3}}Z \tag{3}$$

As explained in [5], the noise in quantum hardware has been shown to behave on average like a depolarizing channel, which makes this choice of noise very suitable to simulate real quantum hardware. The strength of this noise is controlled by a parameter p equivalent to the probability of a noise channel being applied. This noise channel is applied following each CNOT gate in the simulation of the time-evolution of the Heisenberg model. The results in this section are based on a 5-qubit Heisenberg model with 100 timesteps (depth 100).

The first fundamental property needed to understand noisy algorithms is how the strength of the noise affects the fidelity compared to the theoretical noiseless algorithm. This was analyzed by tuning the noise strength p and calculating the fidelity between the noisy result and the default.mixed density matrix simulator in PennyLane-Lightning. The results of this are found in Fig. 3.

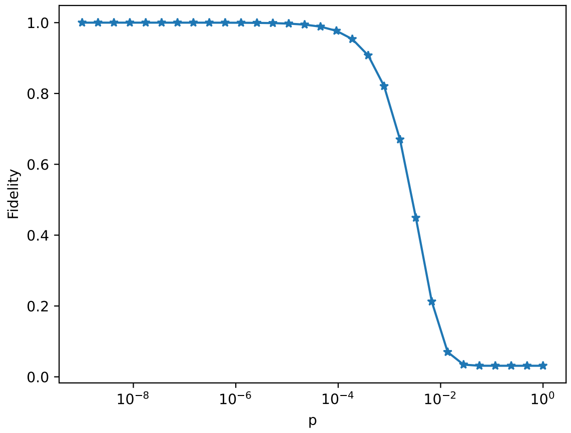

FIG. 3. Effect on fidelity relative to noiseless circuit for various noise strengths

This analysis shows that there are 3 distinct regions of noise. Weak noise under roughly $p = 10^{-4}$ has no effect on fidelity. Strong noise over $p = 10^{-2}$ overwhelms the algorithm and destroys the fidelity compared to the theoretical noiseless algorithm. Most interesting however is the intermediate regime where the depolarizing noise begins to lower the fidelity. This is where we expect the most interesting effects of noise to occur. Note that these regions are fundamentally related to the interaction parameters J_i of the Hamiltonian being simulated. Scaling J_i also scales this curve accordingly.

Following this, the Von Neumann entropy was calculated during a time-evolution with different noise strengths. This is important for classical simulations when considering methods such as tensor networks where the bond dimension growth is related to entanglement growth in the state [6]. The results be seen in 4.

It can be seen that depolarizing noise causes a growth in entanglement compared to the noiseless algorithm. We hoped that noise could inhibit entanglement growth without destroying the fidelity. This would allow injecting

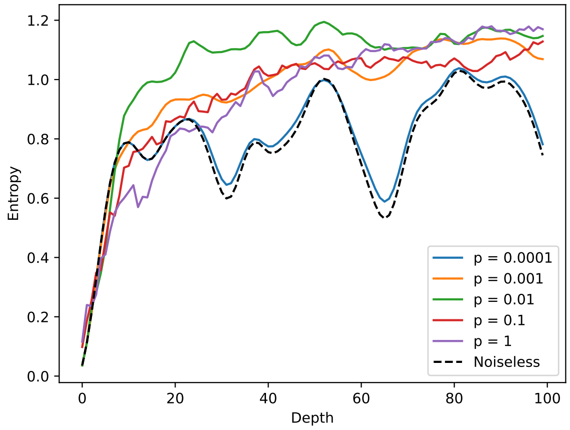

FIG. 4. Effect on entropy over time for various noise strengths.

noise into an algorithm in order to better simulate it for longer times by slowing the growth of entanglement. Further analysis could be useful to find if this is true for other noise models.

IV. APPLICATIONS

A. Optimization of Heisenberg model ground state

Our first application is an analysis of whether and how noise affects variational algorithms. For this, we create a variational model for the VQE, and try to optimize it to find the ground state of the transverse field Heisenberg model

While there has been a number of articles on the effect of noise on VQE [7, 8], and especially how it can be damaging to the results [9], this effect does seem to depend heavily on the type of noise applied. In this work, we choose an error model described by the Kraus operators:

$$\begin{aligned} K_1 &= R_x(s), \\ K_2 &= R_y(s) \\ K_3 &= R_z(s) \end{aligned} \tag{4}$$

This model is simulated by stochastically applying the three different rotation gates with probability p and a "strength" of the noise s, i.e. the angle of the rotation. This is done by generating a random number ϵ between 0 and 1 before each possible rotation. If $\epsilon < p$, then that rotation is applied. This is performed independently for each rotation after each CNOT gate in the model.

For the model itself, two layers of encoding are used, each consisting of a layer of parameterized R_y gates, CNOT gates, and parameterized R_z gates, shown in figure 5 for six qubits without noise, and figure 6 with $p = 0.2$.

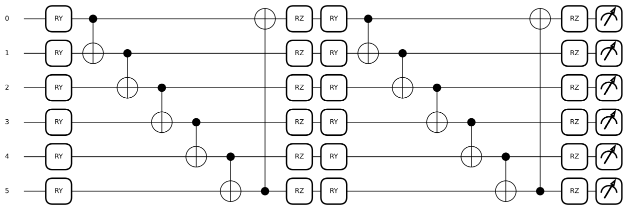

FIG. 5. Noiseless encoding model for a 6 qubit circuit for VQE.

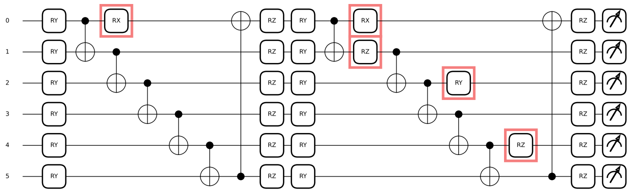

FIG. 6. Noisy encoding model for 6 qubit circuit for VQE with p=0.2. The randomly applied gates are marked in red.

To investigate the effect of noise, the model was optimized using the PennyLane `AdamOptimizer` until it converges or reaches a maximum number of optimization loops. This routine was done for the Heisenberg model with $J_x = 1, J_y = 2, J_z = 1, h = 0.5$, with a noise strength of $s = 0.2$ and multiple different noise probabilities p. To achieve larger systems of 10 qubits, the `lightning.qubit` backend was used on an NVIDIA A100 on the Cyxtera/run:ai cluster. The results of this can be seen in Fig. 7.

As expected, the noiseless model is able to approach the true eigenstate energy (calculated by exact diagonalization). The interesting part is however, that while high level of noise lead to an unstable behaviour, intermediate levels, in this case especially p=0.05, can lead to a faster convergence to the ground state compared to the noiseless run. As an additional remark, the model for p=1 applies the noise at every single CNOT gate, and therefor essentially describes a different model entirely. This shows, that certain types of noise can effectively be used as a resource to improve VQE algorithms.

B. Quantum simulations in noisy circuits

The study of closed-quantum systems is fully governed by Schrödinger's equation, and therefore it encompasses a deterministic process. On the other hand, the evolution of open quantum systems, i.e. systems that interact with their environment through noise, is inherently of stochastic nature. Simulators based on Tensor Networks are regarded as the state of the art for closed quantum systems [10]. However, simulating evolution under the influence of noise is still predominated by density matrix-based al-

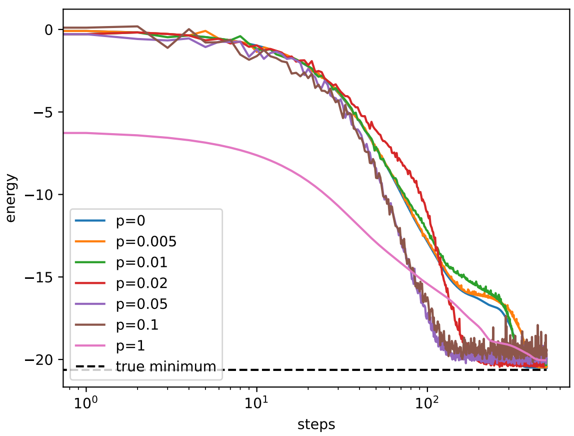

FIG. 7. Convergence rate of VQE approach for a 10-site transverse field Heisenberg model with J_x=1, J_y=2, J_z=1 and h=0.5. Different noise probabilities are shown, with a strength parameter of $s = 0.2$. While very low noise levels barely affect the behaviour, and high noise levels lead to unstable behaviour without convergence, intermediate levels significantly improve the rate the model approaches the eigenstate

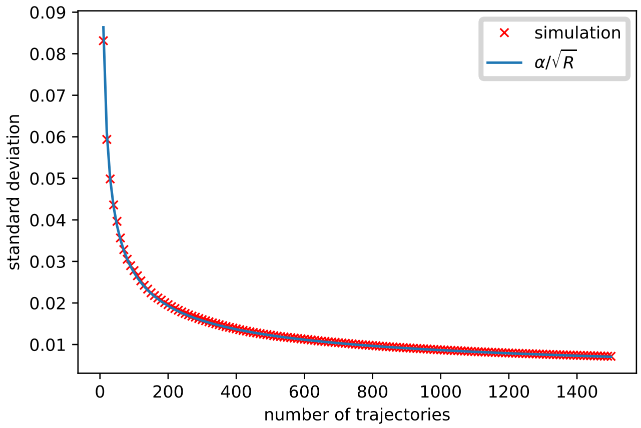

FIG. 8. Standard deviation against number of trajectories in a 10-qubit system, for the calculation of fidelity between stochastic-noise method and exact density matrix simulation. $\alpha = 3.66$

gorithms [11]. This second scenario is of particular interest as we are currently in the NISQ era where quantum computers are still widely affected by noise.

The problem with this approach arises when attempting to study systems of higher dimension, as the classical simulation becomes prohibitely-expensive due to increase of the memory required as d^{2n} where n is the number of sites in our system and d is the local dimension ($d = 2$ for qubits). A naive implementation would then make the computational cost grow exponentially with the number of qubits, making it impossible to study systems of interest.

In this work we employ a method based on the stochastic application of noise operators to our quantum circuit. Through independent runs (trajectories) of this process, we obtain ensembles of state-vectors that can be used to obtain an approximate final noisy state. This work is based on the paper by W. Berquist et.al [12], and adapted to the simulation of state evolving under the Heisenberg Hamiltonian model, under the influence of depolarizing noise. The computation is offloaded to the backend provided by NVIDIA's cuQuantum through PennyLane's Lightning-GPU plugin. For all the simulations we employed a noise parameter $p = 0.0033$, in accordance to the simulations performed in [5].

As a first step to test the validity of our approach, we studied the convergence of the algorithm in dependence of the number of trajectories. Since this stochastic application of noise is in essence a Monte Carlo method, we used the pertinent analysis of the standard deviation [13]. The results for a 10-qubit system shown in Fig. 8 show the convergence of our algorithm with the expected $1/\sqrt{R}$ behaviour, with R the number of trajectories.

In order to make sense of our simulation and verify its correctness, the fidelity for the final state of the noisy circuit was calculated using the statevector and exact density matrix approaches. This was done for different number of qubits n, a circuit with 10 Trotter steps, normalized time of $T = 1$, coupling parameters as mentioned in III, and a noise probability $p = 0.0033$ as we motivated above. To calculate the fidelity we first took the absolute value squared of the final state-vector to convert into a probability vector ψ^r

$$\psi^r = \sum_i^{2^n} |\psi_i|^2 e_i \tag{5}$$

with the computational basis states e_j. We repeat this process R times and we add the results for each trajectory to obtain an average probability vector.

$$\psi_{avg} = \frac{1}{R}\sum_r^R \psi^r \tag{6}$$

On the other hand, from the density matrix ρ obtained in the exact simulation we retrieve the probabilities encoded in the diagonal entries and store them in another probability vector

$$\psi_{exact} = \sum_i^{2^n} \rho_{i,i} e_i \tag{7}$$

Finally, we calculate the fidelity as

$$F(\psi_{avg}, \psi_{exact}) = |\langle\sqrt{\psi_{avg}}, \sqrt{\psi_{exact}}\rangle|^2 \tag{8}$$

where we took the square root of the probability vectors as the inner product $\langle \cdot, \cdot \rangle$ is defined for state-vectors. Figure 9 shows the fidelity plot against the number of trajectories which provides us with insight on the number of independent runs that needed to obtain a desired accuracy.

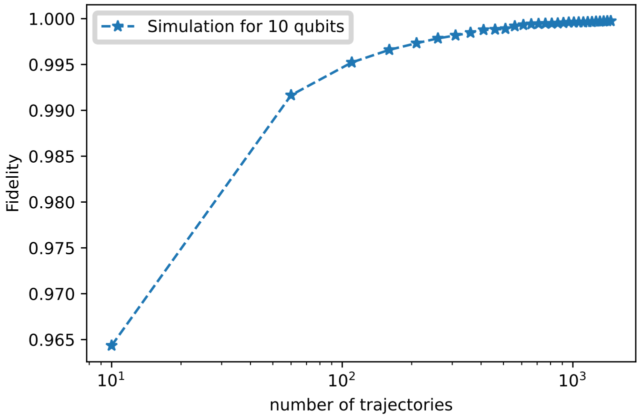

FIG. 9. Fidelity of stochastic noise algorithm with respect to the exact density matrix simulation against number of trajectories. Parameters defined in the main text

In order to perform runtime comparisons between our statevector-based method and the density matrix one, we first needed to find a suitable number of trajectories that would achieve a high accuracy, and still result in a speed-up for a range of system sizes. Figure 10 shows a comparison of fidelities across system sizes of up to 10 qubits for different number of trajectories. Aided by this comparison, we decided to perform the runtime benchmarks with 200 trajectories.

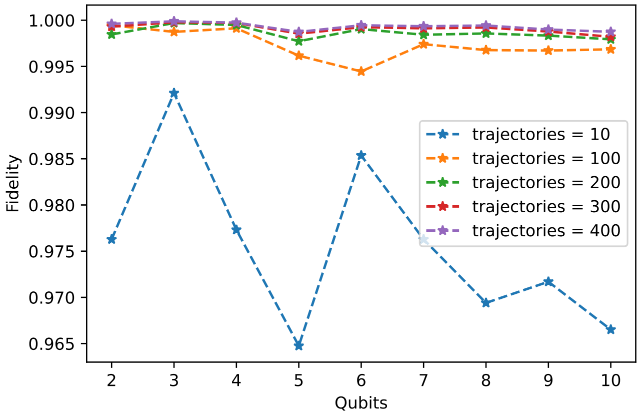

FIG. 10. Fidelity of stochastic noise algorithm with respect to the exact density matrix simulation against number of trajectories. Parameters defined in the main text

In addition, and as one of the highlights of our comparisons, we benchmarked the performance of the stochastic algorithm by comparing the wall-time needed to run our simulation using three different backends provided by PennyLane:

- `default.mixed` : Exact density matrix simulation
- `lightning.qubit` : CPU-based statevector simulation
- `lightning.gpu` : GPU-assisted statevector simulation

The results shown in 11 depict that while the density matrix method yields a better runtime for small systems(less than roughly 9 qubits), the exponential nature of this method becomes evident already for systems above 10 qubits, with runtimes of the order of 10^2 seconds. On the other hand the statevector simulators show an initially poor performance due to the overhead caused by the calculation of the individual trajectories. However, both `lightning.qubit` and `lightning.gpu` performed better than the density matrix simulator for larger systems. Furthermore, the GPU-assisted simulation outperformed the CPU-based one, allowing to simulate the time evolution of systems up to 25 qubits in the same time the density matrix method could only simulate 10 qubits.

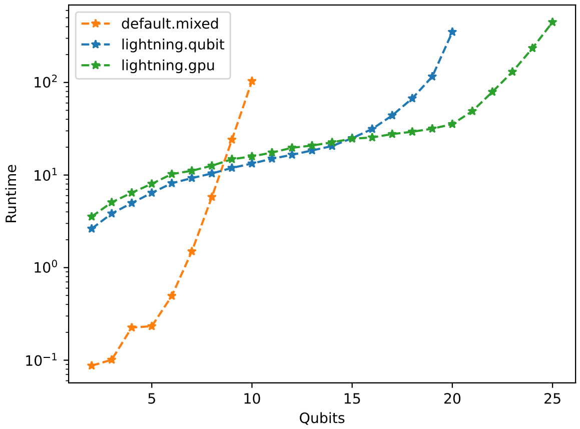

FIG. 11. Runtime against number of qubits in Heisenberg time evolution quantum circuit, performed for three different backends available in Pennylane. Statevector simulations (`lightning.qubit` and `lightning.gpu`) were performed with 200 trajectories. All simulations used 10 Trotter steps, normalized time of $T = 1$, coupling parameters as mentioned in III, and a noise probability $p = 0.0033$

V. CONCLUSION

In this project for QuHack 2023, we have utilized Pennylane and NVIDIA GPUs to determine the effect of noise on quantum simulations. We have demonstrated that GPU-assisted simulations provide a considerable improvement in runtime over density matrix and CPU-based statevector simulations, while still eventually following the same exponential scaling that limits classical simulation. We have used this to be able to simulate more qubits with faster runtime for each our use cases. This is an essential requirement for studying current noisy NISQ devices and simulating QPUs as they continue to grow in system size.

We have found that there is a threshold after which stronger noise destroys the fidelity of a quantum system. This shows a tolerance to noise which is dependant on the ratio between noise strength and the interaction parameters of the Hamiltonian being simulated. We have also presented the effect of noise on the entanglement entropy of the system, in particular that depolarising noise adds to the system's entropy. These findings have been applied to density matrix simulations of open quantum systems and to optimization algorithms for finding the ground state energy where we see in both cases that a small addition of noise assists both applications by increasing both speed and accuracy. This research therefore directly supports the field of quantum simulation.

[1] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. Sohaib Alam, G. Alonso-Linaje, B. Akash-Narayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. Di Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isacsson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R. A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J. J. Meyer, J. A. Montañez-Barrera, R. Moyard, Z. Niu, L. J. O'Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R. A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, and N. Killoran, PennyLane: Automatic differentiation of hybrid quantum-classical computations, arXiv e-prints , arXiv:1811.04968 (2018), arXiv:1811.04968 [quant-ph].

[2] N. cuQuantum team, cuquantum.

[3] G. Vidal, Efficient simulation of one-dimensional quantum many-body systems, Physical Review Letters **93**, 040502 (2004), arXiv:quant-ph/0310089.

[4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[5] Y.-T. Chen, C. Farquhar, and R. M. Parrish, Low-rank density-matrix evolution for noisy quantum circuits (2021).

[6] F. Verstraete, J. I. Cirac, and V. Murg, Matrix Product States, Projected Entangled Pair States, and variational renormalization group methods for quantum spin systems, Advances in Physics **57**, 143 (2008), arXiv:0907.2796 [cond-mat, physics:math-ph, physics:quant-ph].

[7] J. Zeng, Z. Wu, C. Cao, C. Zhang, S.-Y. Hou, P. Xu, and B. Zeng, Simulating noisy variational quantum eigensolver with local noise models, Quantum Engineering **3**, 10.1002/que2.77 (2021).

[8] C. Xue, Z.-Y. Chen, Y.-C. Wu, and G.-P. Guo, Effects of quantum noise on quantum approximate optimization algorithm (2019).

[9] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, Noise-induced barren plateaus in variational quantum algorithms, Nature Communications **12**, 10.1038/s41467-021-27045-6 (2021).

[10] I. L. Markov and Y. Shi, Simulating quantum computation by contracting tensor networks, SIAM Journal on Computing **38**, 963 (2008).

[11] D. Manzano, A short introduction to the lindblad master equation, AIP Advances **10**, 025106 (2020).

[12] W. Berquist, D. Lykov, M. Liu, and Y. Alexeev, Stochastic approach for simulating quantum noise using tensor networks (2022).

[13] H.-P. Breuer and F. Petruccione, *The theory of Open Quantum Systems* (Clarendon, 2010).