

Evaluating the Impact of Noise on Variational Quantum Circuits in NISQ Era Devices

Bikram Khanal 

*Department of Computer Science
Baylor University
Waco, TX
bikram_khanal1@baylor.edu*

Pablo Rivas , Senior, IEEE

*Department of Computer Science
Baylor University
Waco, TX
pablo_rivas@baylor.edu*

Abstract—The limited supply of qubits and significant quantum noise impose limitations on the capability of quantum algorithms in the Noisy Intermediate-Scale Quantum (NISQ) era. NISQ devices have a variety of applications, such as Variational Quantum Circuit (VQC), which provides answers to difficult optimization and machine learning problems. This paper presents a thorough investigation of quantum variational classification in the NISQ context, with a focus on comprehending noise’s impact on various feature maps and VQCs. We evaluate the effectiveness of quantum classifiers using a variety of datasets, ranging from straightforward binary classification problems to more complex tasks. Our results reveal the critical role that feature maps and variational circuit selection play in mitigating the effect of noise, identifying specific quantum circuit designs that exhibit robustness even in noisy situations. In order to highlight the potential of quantum machine learning in solving complex problems within the NISQ setting, this study emphasizes the delicate interaction between feature map selection, variational circuit design, dataset complexity, and quantum noise.

Index Terms—Variational Quantum Circuits, NISQ Devices, Noise, Feature Map, Quantum Classifiers.

I. INTRODUCTION

The emergence of Noisy Intermediate-Scale Quantum (NISQ) devices has significantly advanced quantum computing. Current quantum processors, although characterized by a limited number of qubits and lack of full fault-tolerance, have the potential to revolutionize computational sciences by tackling problems deemed intractable for classical computers [1]–[3]. These quantum devices’ nascent stage and high noise levels present a unique set of challenges and opportunities [1].

One of the promising applications of NISQ devices lies in machine learning and optimization, facilitated by variational quantum circuits [4]. Hybrid quantum-classical algorithms, such as the Variational Quantum Eigensolver (VQE) [5] and the Quantum Approximate Optimization Algorithm (QAOA) [6], exemplify this approach by leveraging the advantages of quantum computation and mitigating potential losses through classical optimization routines. These algorithms play a crucial role in the construction of quantum machine learning (QML) models.

The rapidly evolving field of QML aims to exploit the unique properties of quantum mechanics, including superposition, entanglement, and interference, to enhance machine learning models [7], [8]. Typically, the generation of QML models

involves training quantum circuits using quantum algorithms in a hybrid quantum-classical approach, which are then optimized using various techniques such as quantum gradient descent [9], variational quantum eigensolver [5], quantum natural gradient [10], and quantum annealing [11]. These techniques are designed to optimize the efficiency and accuracy of model training on quantum hardware [12]. A central component of these algorithms is the variational circuit, a parameterized quantum circuit where parameters are iteratively updated using classical optimization methods to minimize a predefined cost function [13].

Parameterized Quantum Circuits (PQCs) [6], essentially Quantum Neural Networks (QNNs) in the context of machine learning [14], and Quantum Kernel Methods (QKMs) [8] are two prevalent approaches for training QML models. QML algorithms, such as Quantum Support Vector Machine (QSVM) [15], Quantum Principal Component Analysis [16], and Quantum K-means [17], have demonstrated quantum speedups over classical methods. Despite the potential of QML to offer considerable computational speedup, it remains unclear whether QML can consistently outperform classical methods across various learning tasks. Ref. [18]–[21] have probed the performance of QML algorithms on diverse classification and regression problems. While these studies suggest that QML can outperform classical methods in certain instances, there are also cases where it underperforms [4], [22].

Another significant challenge in QML is the noise inherent in quantum hardware, which can substantially affect model performance and accuracy [12], [23]. The performance of QSVM on real-world datasets, for instance, has been found to be lower than its theoretical promise due to quantum hardware noise [24]. For QNNs, training on large datasets poses a challenge due to the high computational complexity of quantum circuits [3].

Several techniques, such as error mitigation [25], quantum error correction [26], variational quantum thermalizing algorithm [27], zero noise extrapolation [28], and randomized circuit resampling [28], have shown theoretical promise in enhancing the accuracy and robustness of QML models in the presence of noise. It is not only of theoretical significance but also of practical importance to study quantum variational classification in the presence of noise. As NISQ devices

continue to evolve, understanding the noise effects on quantum classifiers becomes crucial for efficient quantum algorithm and hardware design [29].

Recognizing the need to understand quantum classifiers' resilience under noise, this paper undertakes a comprehensive study of quantum variational classification in the NISQ era. We evaluate the impact of noise on different feature maps, variational circuits, and their performance across diverse datasets, including iris [30], [31], Two Interleaving Half Circles, Concentric Circles, Binary Classification, and Isotropic Gaussian Blobs from scikit-learn¹, and a synthetic dataset following [29]. The effects of bitflip and thermal noise on the performance of variational quantum circuits on these datasets are thoroughly investigated. Our approach involves establishing a baseline performance by implementing a variational quantum circuit model for a specific optimization problem on a noiseless quantum simulator. Subsequently, we simulate the effect of noise by running the circuit on two different noisy simulators. The first simulator introduces bitflip noise, modeled by the Pauli-X error channel with a probability of p [9]. The second simulator introduces thermal noise, modeled by the amplitude damping channel with a damping rate of γ [32]. The performance of the circuit is then evaluated under these noisy conditions and compared to the baseline established with the noiseless simulator.

II. VARIATION QUANTUM CIRCUITS AND FEATURE MAPS

A. Variational Quantum Circuits

At the heart of variational quantum algorithms are Variational Quantum Circuits (VQCs), which are paramount in the design and performance of quantum classifiers. VQCs are parameterized quantum circuits with parameters that undergo iterative optimization to minimize a designated cost function [33]. A VQC can be generally formulated as follows:

$$U(x, \theta) = U_s(x) \prod_i U_w(\theta_i), \quad (1)$$

In this equation, $U(x, \theta)$ signifies the unitary transformation embodied by the variational circuit. The feature map, $U_s(x) = |\phi(x)\rangle^{\otimes n} : \phi : \mathcal{X} \mapsto \mathcal{H}$, maps from the input space \mathcal{X} to the quantum Hilbert space \mathcal{H} , with n representing the number of qubits. $U_w(\theta_i)$ refers to the parameterized single-qubit or multi-qubit gates, while $\theta = \{\theta_1, \theta_2, \dots, \theta_m\}$, $m, n \in \mathbb{Z}_+$, are the parameters of the circuit [34].

A VQC-based machine learning model can be construed as a quantum circuit [7], [35]. The QML model is derived by measuring a Hermitian operator \mathcal{M} in Equation 1, yielding:

$$f_\theta(x) = \langle U(x, \theta) | \mathcal{M} | U(x, \theta) \rangle, \quad (2)$$

The performance of Equation 2 in terms of learning can be enhanced via training and minimizing the cost function. In the realm of a quantum classifier, the cost function could be defined as:

$$C(\theta) = \frac{1}{N} \sum_{i=1}^N |y_i - f_\theta(x_i)|^2 \quad (3)$$

¹<https://scikit-learn.org/stable/datasets.html>, version 1.2.2

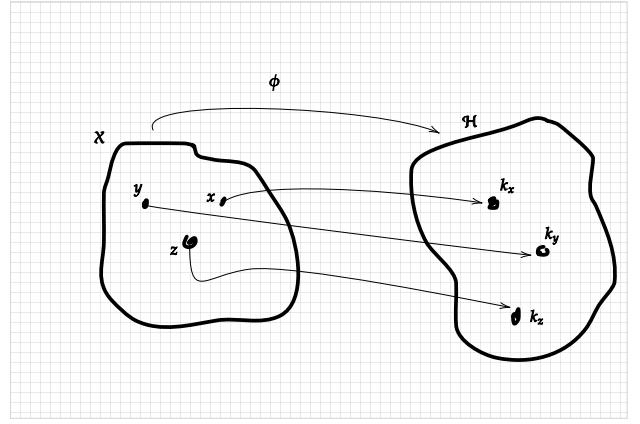


Fig. 1. Mapping of input from input space \mathcal{X} to a quantum Hilbert space \mathcal{H} via a mapping function $\phi(\cdot)$. $k(\cdot)$ is a vector on \mathcal{H} .

In this equation, N is the total number of training examples, y_i are the true labels, x_i are the input data, and $f_\theta(x_i)$ are the predictions made by the quantum classifier for each input x_i .

The parameters θ of Equation 1 are updated using classical optimization algorithms, such as gradient descent, in order to minimize Equation 3.

The update rule for the parameters in the case of gradient descent is:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \quad (4)$$

In this equation, η is the learning rate, and $\nabla L(\theta_t)$ represents the gradient of the loss function, $L(\theta) = \sum_i |y_i - \hat{y}_i|^2$, with respect to the parameters at the t_{th} iteration [36].

B. Feature Maps

Feature maps function as the bridge that encodes classical data into quantum states, thus forming an integral component of quantum machine learning architectures. They map input data $x = (x_1, x_2, \dots, x_d)$ from a d -dimensional space to a quantum state $|\phi(x)\rangle$ in the Hilbert space \mathcal{H} [4].

The general form of a quantum feature map, ϕ , can be depicted as:

$$|\Psi(x)\rangle = \phi(x) |0\rangle = U(x) |0\rangle \quad (5)$$

In this equation, $|0\rangle$ represents the initial state of the qubits, and $U(x)$ is the unitary transformation that is dependent on the input data x [8], [29]. Fig. 1 provides an illustration of this mapping between two feature spaces.

Different types of feature maps can be employed for encoding. The distinguishing factor among these feature maps lies in their data encoding methods and the transformations applied to the initial state. Various feature maps suitable for a wide range of QML problems have been proposed by different authors [4], [29], [37]–[39]. However, in this paper, we confine our discussion to several derivatives of the encoding strategy proposed in [29]. For a circuit with d depth, a unitary operator

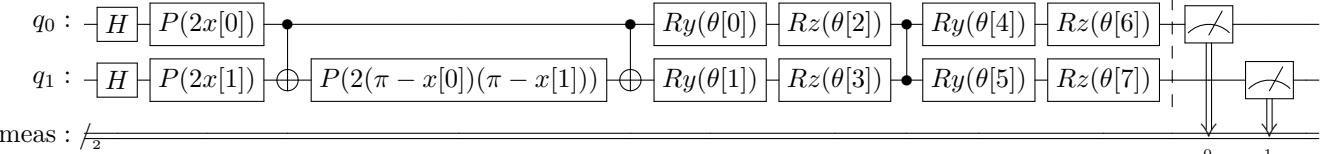


Fig. 2. Quantum circuit diagram representing a two-qubit quantum operation consist of 2 dimensional feature vector x and 8 learnable parameter vector θ . The circuit consists of Hadamard (H) gates, phase (P) gates, controlled-NOT (CNOT) gates, and rotation gates (R_y and R_z) parameterized by angles from the vectors x and θ . The gates are applied in a specific sequence to qubits q_0 and q_1 , followed by measurement (indicated by the meter symbols) on a computational basis. The phase gates include a parameterized phase shift, with one of the phase gates incorporating a product of angles. The rotation gates are applied with different parameters from the θ vector.

defined by the following equation can execute a classically intractable feature map:

$$U_\theta(x) = \prod_d \exp \left(i \sum_{j \subseteq 1, 2, \dots, n} \phi_j(x) \prod_{k \in j} P_k \right) H^{\otimes n} \quad (6)$$

In this equation, the number of qubits is represented by $n = |x|$, $\phi_j(x)$ denotes the coefficient encoding for x , and $P_k \in \{I, X, Y, Z\}$ can be extended as Pauli-gates, while H symbolizes the Hadamard gate.

These feature maps, serving as the intermediary between classical data and quantum computations, play a pivotal role in the performance of quantum classifiers under different types of noise. Therefore, understanding their behavior and resilience in the face of noise is vitally important in designing and applying quantum variational classifiers [33].

III. METHODOLOGY

A. Problem Setup

We initiate our discussion by defining the binary classification learning problem for the datasets specified in Section I. Let's assume that $\mathcal{X} \subseteq \mathbb{R}^d$ is a feature space of d dimensions. Given a dataset $D \in \mathcal{X}$ comprising N training examples, $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where $x_i \in \mathbb{R}^d$ denotes a d-dimensional feature vector and $y_i \in \{0, 1\}$ indicates the corresponding binary label, the goal of a binary classifier is to learn a function $f : \mathbb{R}^d \rightarrow \{0, 1\}$ that can accurately predict the label y for new, unseen feature vectors x [40].

In the context of quantum machine learning, this learning problem is mapped onto a quantum system. Here, the feature vectors are encoded into quantum states using a feature map, as described in Section II-B. Thus, the quantum classifier can be expressed as follows:

$$f(x) = \langle \Psi(x) | U^\dagger(\theta) M U(\theta) | \Psi(x) \rangle \quad (7)$$

In this equation, $|\Psi(x)\rangle = \Phi(x)|0\rangle$ is the quantum state obtained by applying the feature map $\Phi(x)$ to the initial state $|0\rangle$, $U(\theta)$ is the variational quantum circuit parameterized by θ , M is the measurement operator, and $\langle \Psi(x) | U^\dagger(\theta) M U(\theta) | \Psi(x) \rangle$ is the expected value of the measurement on the state obtained by applying $U(\theta)$ to $|\Psi(x)\rangle$ [35].

In our setup, we choose the measurement operator, M , to be the Pauli operator. This results in an outcome within the $\{-1, +1\}$ range. For any given feature vector x , we assign a label as $y = \text{sign}(f(x))$. The parameters θ of the variational

quantum circuit are optimized to minimize the cost function defined in Section II-A, thereby learning the function f that best classifies the training examples [13].

B. Experimental Setup

This study employed a quantum variational approach to optimize a parameterized quantum circuit using a classical optimizer in a 2 qubits system. The optimization procedure is carried out under noisy conditions, examining the impacts of various noise models. The experimental setup, including the quantum circuit architecture, the optimization algorithm, the loss function, and the noise models, is detailed in the following subsections.

1) *Quantum Circuit Architecture*: We developed two random variational quantum circuits inspired by qiskit [41] variational circuits. Fig. 2 depicts one such quantum circuit for a 2-dimensional feature vector and 8 learnable parameters.

- Two-Local-Inspired Circuit: This configuration comprises alternating rotation and entanglement layers. The rotation layers include single-qubit gates independently applied to each qubit, while the entanglement layers encompass two-qubit gates that entangle the qubits according to a pre-determined strategy. The specific gates used in the rotation and entanglement layers are randomly selected from a set of candidate gates, such as Pauli rotation gates and controlled gates.
- EfficientSU(2)-Inspired Circuit: This configuration consists of layers of single-qubit operations spanned by the special unitary group SU(2) and CNOT entanglements. The circuit is designed to prepare trial wave functions for variational quantum algorithms or to function as a classification circuit for quantum machine learning. The specific single-qubit gates are randomly selected from the SU(2) group, which represents 2x2 unitary matrices with determinant 1, such as the Pauli rotation gates.

2) *Loss Function and Optimization*: The binary cross-entropy loss function, as defined in equation 8, was employed in the optimization process.

$$L(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \quad (8)$$

In this equation, y is the true label, and $\hat{y} = f(x)$ is the predicted probability. This loss function measures the divergence between the predicted probabilities and the true labels in binary classification tasks. [42].

The optimization of the parameterized quantum circuit was performed using the Simultaneous Perturbation Stochastic Approximation (SPSA) optimizer [43], [44]. SPSA is a well-regarded optimization method, particularly suitable for noisy optimization problems. Its efficiency in terms of function evaluations is due to the fact that it only requires two function measurements to compute an estimate of the gradient, regardless of the dimension of the optimization problem.

The SPSA algorithm follows these steps when given a loss function $L(\theta)$ to minimize, where $\theta \in \mathbb{R}^n$ denotes the parameters of the model:

- (i) Initialize the parameters $\theta^{(0)}$ and set the iteration number $k = 0$.
- (ii) At each iteration k , generate a random perturbation $\Delta^{(k)}$ with i.i.d. entries drawn from a symmetric Bernoulli ± 1 distribution.
- (iii) Estimate the gradient $g^{(k)}$ of the loss function at $\theta^{(k)}$ using the symmetric difference quotient:

$$g^{(k)} = \frac{L(\theta^{(k)} + c_k \Delta^{(k)}) - L(\theta^{(k)} - c_k \Delta^{(k)})}{2c_k \Delta^{(k)}} \quad (9)$$

where c_k is a positive step size sequence.

- (iv) Update the parameters using the estimated gradient:

$$\theta^{(k+1)} = \theta^{(k)} - a_k g^{(k)} \quad (10)$$

where a_k is a positive gain sequence.

- (v) Set $k = k + 1$ and repeat steps (ii) – (iv) until a stopping criterion is met.

The step size and gain sequences c_k and a_k are usually chosen to satisfy certain conditions for convergence, such as $a_k = a/(k + A)^\alpha$ and $c_k = c/(k + 1)^\gamma$, where a, A, α, c, γ are constants.

This algorithm provides a simple and efficient method for optimizing the parameters of variational quantum circuits, particularly in the presence of noise.

C. Experimental Procedure

Our variational quantum circuit model was trained for a total of 300 steps, with initial parameters randomly initialized from a uniform distribution using the numpy library in Python. We leveraged standard machine learning libraries alongside Qiskit for quantum computing functionalities.

Three distinct experimental conditions were set up for the study: (1) a noiseless simulation, (2) a simulation with bit-flip noise, and (3) a simulation with thermal noise. The noiseless simulation was conducted using Qiskit's Aer module “qasm_simulator” backend with 1024 shots for each circuit execution. To ensure reproducibility, the random seeds for the simulator and the transpiler were set to 42.

The simulation incorporating bit-flip noise utilized the AerSimulator backend with a noise model replicating bit-flip errors. The noise model was configured to flip a single qubit gate with a probability of 0.05 when a single qubit gate was applied, and a probability of 0.1 was set to flip the state of a qubit during measurement. A probability of 0.05 was set for resetting a qubit to state $|1\rangle$ instead of $|0\rangle$. Correlated bit-flip

errors were introduced to two-qubit gates via a mixed Pauli error channel within Qiskit. The noisy backend for bit-flip was set up with the noise model, 1024 shots per circuit execution, and random seeds of 42 to maintain reproducibility.

For the simulation with thermal noise, the AerSimulator backend was supplied with a noise model that mimicked thermal errors. This model introduced thermal relaxation errors, determined by the relaxation time constant T_1 and the dephasing time constant T_2 (where $T_2 \leq 2T_1$) for each qubit. Instruction times for reset and measurement were set to 1 microsecond, with single-qubit gates represented by Pauli gates and two-qubit gates by CNOT. The noisy backend for thermal noise simulation shared the same settings as the bit-flip configuration.

We conducted the optimization process under each of these three conditions. The results obtained from these simulations were subsequently analyzed to evaluate the impact of quantum noise on the performance of the variational quantum circuit.

IV. RESULTS AND DISCUSSION

A. Result

Our analysis reveals noteworthy insights about the interaction between quantum noise and the performance of various datasets in a quantum machine learning framework. Six datasets—Isotropic Gaussian Blobs, Concentric Circles, Iris, Adhoc, Binary Classifications, and Two Interleaving Half Circles—training and test accuracies were examined under three distinct conditions: Simulator, Bitflip noise, and Thermal noise. The scatter plot for these datasets is presented in Fig. 3 for visualization. A summary of the training and testing accuracies for all datasets and noise conditions is presented in Table I.

For the “Isotropic Gaussian Blobs” dataset, we observed a significant dip in training and test accuracies when exposed to Bitflip and Thermal noise compared to the Simulator. The training accuracy decreased from 51.25% with no noise to 37.5% and 40% with Bitflip and Thermal noise, respectively. The test accuracy exhibited a similar trend, dropping from 50% to 20% and 35% under Bitflip and Thermal noise conditions, respectively. The lower performance could be attributed to the dataset’s inherent structure, which might be more susceptible to quantum noise, highlighting the need for more robust quantum error correction techniques for such datasets. The “Concentric Circles” dataset revealed a different image. While there was a noticeable drop in performance while using Bitflip and Thermal noise, it was not as severe as when using Isotropic Gaussian Blobs. The training accuracy dropped from 90% in the Simulator to 72.5% and 70% in Bitflip and Thermal noise, respectively. The test accuracy followed a similar pattern, dropping from a considerable 100% in the simulated environment to 55% and 65% under Bitflip and Thermal noise, respectively. The low testing accuracy in the presence of Bitflip noise suggests that this type of noise might be more disruptive to the model’s generalization capabilities. The “Iris” dataset exhibited exceptional performance across all noise conditions, with its lowest test accuracy at 90% under Bitflip noise. It displayed a perfect test accuracy of 100% under

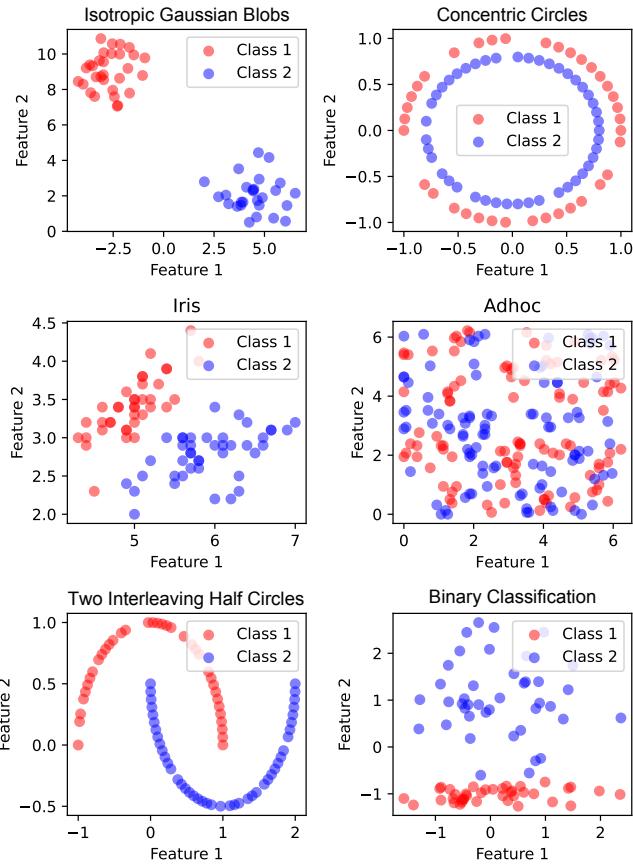


Fig. 3. Scatterplots of six distinct datasets, each preprocessed to have two features, making them suitable for binary classification and visualization. Two Interleaving Half Circles is a synthetic dataset displaying two interleaving half-circle distributions. Binary Classification is a customizable synthetic dataset generated for multi-class classification tasks. The Adhoc dataset is tailor-made for a unique task, showcasing the versatility of data handling in machine learning. The classic Iris dataset features measurements of three species of Iris flowers utilized for classification problems. Isotropic Gaussian Blobs generate isotropic Gaussian blobs ideal for clustering or classification tasks. Lastly, Concentric Circles is a synthetic dataset with two classes distributed in concentric circles. Each subplot in the figure represents a distinct data structure and classification task, highlighting the diverse challenges in machine learning.

the Simulator and Thermal noise, demonstrating the robustness of this dataset.

The “Adhoc” dataset performed exceptionally well across all noise conditions, maintaining a test accuracy of 95% and 100% under Bitflip and Simulator/Thermal noise, respectively. This high performance could be related to the inherent properties of the “Adhoc” dataset, which may be more resilient to quantum noise.

The “Two Interleaving Half Circles” dataset, similar to “Concentric Circles”, showed a moderate decrease in performance when subjected to Bitflip and Thermal noise. However, it still maintained high training and test accuracies above 90% in all conditions, suggesting its strong resilience to quantum noise.

Finally, the “Binary Classification” dataset displayed a unique trend, with test accuracy higher than training accuracy across all noise conditions. This might be indicative of the

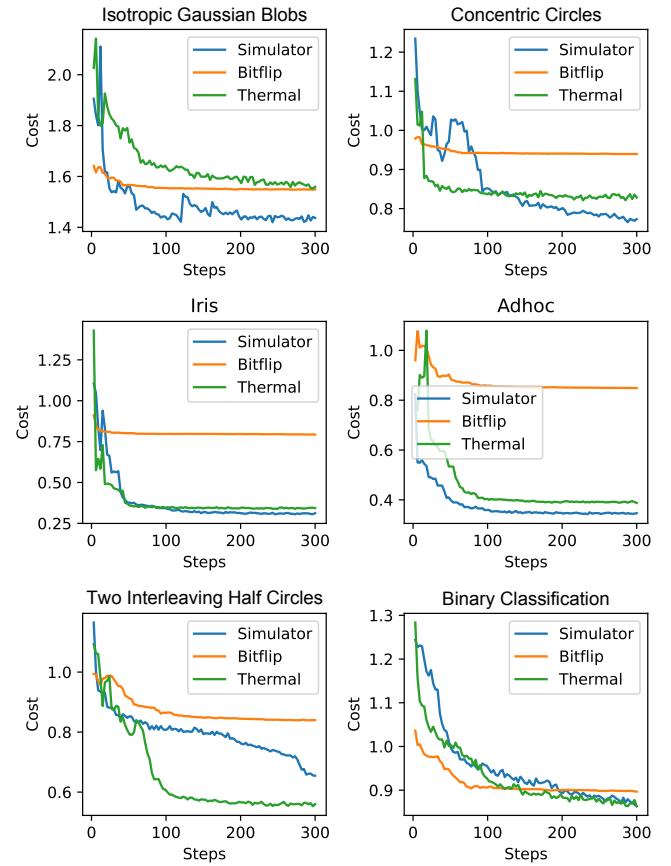


Fig. 4. Comparative analysis of cost versus training steps for six datasets in three different noise conditions: Simulator, Bitflip, and Thermal. Each subplot demonstrates the impact of quantum noise on the learning process of variational circuits, illustrating the varying degrees of noise resistance depending on the complexity of the dataset.

TABLE I
PERFORMANCE OF VARIOUS DATASETS UNDER SIMULATOR, BITFLIP, AND THERMAL NOISE CONDITIONS

Dataset	Noise	Train Accuracy	Test Accuracy
Isotropic Gaussian Blobs	Simulator	0.5125	0.5
	Bitflip	0.375	0.2
	Thermal	0.4	0.35
Concentric Circles	Simulator	0.9	1.0
	Bitflip	0.725	0.55
	Thermal	0.7	0.65
Iris	Simulator	0.9625	1.0
	Bitflip	0.8875	0.9
	Thermal	0.925	0.95
Adhoc	Simulator	0.995	1.0
	Bitflip	0.945	0.95
	Thermal	0.995	1.0
Two Interleaving Half Circles	Simulator	0.9	0.9
	Bitflip	0.9	0.85
	Thermal	0.9375	0.9
Binary Classification	Simulator	0.675	0.75
	Bitflip	0.7	0.8
	Thermal	0.7375	0.8

dataset’s properties or beneficial interaction with the quantum noise models employed.

B. Discussion

Table I and Fig. 4 present an intriguing picture of the interplay between feature maps, variational circuits, quantum noise, and the nature of the datasets in a quantum machine learning setting. Noteworthy is the better performance of models on the Iris and Adhoc datasets, which are typically more complex than the binary classification problems presented by the Isotropic Gaussian Blobs and Binary Classification datasets. This could indicate that the feature maps and variational circuits used in this experiment are more adept at capturing complex patterns in the data rather than simple binary separations.

In the context of noise, the Bitflip and Thermal noise conditions demonstrate a varying impact on the training cost of the variational circuits. As the step count increases, the training cost tends to decrease more for Simulator conditions than their Bitflip and Thermal counterparts, indicating that quantum noise can slow the learning process or even halt it. The Bitflip noise, for instance, consistently causes the cost to remain almost constant across all datasets, signaling that this kind of quantum noise could seriously impede learning in a NISQ-era quantum machine learning model.

The performance drop in Isotropic Gaussian Blobs and Binary Classification could be because the quantum circuits' high capacity is not as advantageous in these simpler problems and could even be detrimental due to overfitting. The less complex the problem, the more the model is susceptible to noise, as evidenced in the Bitflip and Thermal noise conditions for Isotropic Gaussian Blobs. On the other hand, for more complex problems like the Adhoc and Iris datasets, the model seems more resistant to noise, indicating that the right choice of feature map and variational circuit can mitigate noise's impact.

The importance of choosing the right feature map for a given dataset comes into sharp relief. The high performance on the Adhoc dataset suggests that if a feature map perfectly captures the data structure, noise is less of an issue, maintaining high accuracy even under noise conditions.

This study's limitation to two qubits is an important factor to consider. Quantum systems gain their power from entanglement, and with more qubits, there's a higher potential for entanglement and hence more computational power. If more qubits were used, the models might perform better on simpler datasets as well, as they could take advantage of this increased capacity to better handle noise and possibly prevent overfitting.

V. CONCLUSION

This research has illuminated the intricate interplay between feature maps, variational circuits, quantum noise, and dataset complexity in quantum machine learning. Notably, our models showcased superior performance on more complex datasets such as Iris and Adhoc, implicating that the chosen feature maps and variational circuits are adept at deciphering intricate data structures.

However, quantum noise, notably Bitflip and Thermal, impeded the optimization of variational circuits, underlining its adverse effects in the NISQ era. Interestingly, the careful choice of feature maps and variational circuits could mitigate

the impact of noise, a phenomenon evident with the Adhoc and Iris datasets.

Our research was constricted to two-qubit systems, limiting the potential for entanglement, a crucial resource in quantum computation. Would the inclusion of more qubits improve performance on simpler datasets by leveraging increased entanglement, even in noisy environments?

In conclusion, our findings underscore the importance of the careful selection of dataset complexity, feature map, and variational circuit design in the face of quantum noise in quantum machine learning. A salient question for future research is: How can we develop more noise-resistant feature maps and variational circuits? Furthermore, how might we strategize noise management in larger quantum systems? Addressing these questions will be instrumental in advancing quantum machine learning in the NISQ era and beyond.

ACKNOWLEDGMENT

The work presented here was supported by the Baylor AI lab in Baylor University's Department of Computer Science. This work is also partially supported by the National Science Foundation under Grant 2136961.

REFERENCES

- [1] M. Bandic, S. Feld, and C. G. Almudever, "Full-stack quantum computing systems in the nisq era: algorithm-driven and hardware-aware compilation techniques," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 1–6.
- [2] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.
- [3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.
- [4] M. Schuld and F. Petruccione, *Supervised learning with quantum computers*. Springer, 2018, vol. 17.
- [5] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, no. 1, p. 4213, 2014.
- [6] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.
- [7] P. Wittek, *Quantum machine learning: what quantum computing means to data mining*. Academic Press, 2014.
- [8] M. Schuld, "Supervised quantum machine learning models are kernel methods," *arXiv preprint arXiv:2101.11020*, 2021.
- [9] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.
- [10] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032331, 2019.
- [11] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse ising model," *Physical Review E*, vol. 58, no. 5, p. 5355, 1998.
- [12] B. Khanal, J. Ordzu, P. Rivas, and E. Baker, "Supercomputing leverages quantum machine learning and grover's algorithm," *The Journal of Supercomputing*, pp. 1–23, 2022.
- [13] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, p. 043001, 2019.
- [14] S. C. Kak, "Quantum neural computing," *Advances in imaging and electron physics*, vol. 94, pp. 259–313, 1995.
- [15] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical review letters*, vol. 113, no. 13, p. 130503, 2014.
- [16] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 9, pp. 631–633, 2014.

- [17] S. Lloyd, P. Rebentrost, and M. Mohseni, “Quantum algorithms for supervised and unsupervised machine learning,” *arXiv preprint arXiv:1307.0411*, 2013.
- [18] N. Schetakis, D. Aghamalyan, P. Griffin, and M. Boguslavsky, “Review of some existing qml frameworks and novel hybrid classical-quantum neural networks realising binary classification for the noisy datasets,” *Scientific Reports*, vol. 12, no. 1, pp. 1–12, 2022.
- [19] S. Kavitha and N. Kaulgud, “Quantum machine learning for support vector machine classification,” *Evolutionary Intelligence*, pp. 1–10, 2022.
- [20] H. Yamasaki and S. Sonoda, “Exponential error convergence in data classification with optimized random features: Acceleration by quantum machine learning,” *arXiv preprint arXiv:2106.09028*, 2021.
- [21] B. Khanal, P. Rivas, and J. Orduz, “Kernels and quantum machine learning,” in *International Conference on Emergent and Quantum Technologies (ICEQT 2022)*, 2022, pp. 1–15.
- [22] B. Duan, J. Yuan, Y. Liu, and D. Li, “Quantum algorithm for support matrix machines,” *Physical Review A*, vol. 96, no. 3, p. 032301, 2017.
- [23] B. Khanal, P. Rivas, J. Orduz, and A. Zhakubayev, “Quantum machine learning: A case study of grover’s algorithm,” in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2021, pp. 79–84.
- [24] E. Farhi and H. Neven, “Classification with quantum neural networks on near term processors,” *arXiv preprint arXiv:1802.06002*, 2018.
- [25] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, “Error mitigation extends the computational reach of a noisy quantum processor,” *Nature*, vol. 567, no. 7749, pp. 491–495, 2019.
- [26] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Physical review A*, vol. 52, no. 4, p. R2493, 1995.
- [27] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, “Variational ansatz-based quantum simulation of imaginary time evolution,” *npj Quantum Information*, vol. 5, no. 1, p. 75, 2019.
- [28] K. Temme, S. Bravyi, and J. M. Gambetta, “Error mitigation for short-depth quantum circuits,” *Physical review letters*, vol. 119, no. 18, p. 180509, 2017.
- [29] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.
- [30] E. Anderson, “The species problem in iris,” *Annals of the Missouri Botanical Garden*, vol. 23, no. 3, pp. 457–509, 1936.
- [31] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [32] H. M. Wiseman and G. J. Milburn, *Quantum measurement and control*. Cambridge university press, 2009.
- [33] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, “Variational quantum algorithms,” *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.
- [34] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, M. Gu, M.-J. Zhu, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, “Experimental quantum computing to solve systems of linear equations,” *Physical review letters*, vol. 110, no. 23, p. 230501, 2013.
- [35] M. Schuld and F. Petruccione, *Machine learning with quantum computers*. Springer, 2021.
- [36] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.
- [37] Y. Liu, S. Arunachalam, and K. Temme, “A rigorous and robust quantum speed-up in supervised machine learning,” *Nature Physics*, vol. 17, no. 9, pp. 1013–1017, 2021.
- [38] M. Schuld and N. Killoran, “Quantum machine learning in feature hilbert spaces,” *Physical review letters*, vol. 122, no. 4, p. 040504, 2019.
- [39] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, “Quantum embeddings for machine learning,” *arXiv preprint arXiv:2001.03622*, 2020.
- [40] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.
- [41] Qiskit contributors, “Qiskit: An open-source framework for quantum computing,” 2023.
- [42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [43] J. C. Spall, “An overview of the simultaneous perturbation method for efficient optimization,” *Johns Hopkins apl technical digest*, vol. 19, no. 4, pp. 482–492, 1998.
- [44] J. Spall, “Implementation of the simultaneous perturbation algorithm for stochastic optimization,” *IEEE Transactions on aerospace and electronic systems*, vol. 34, no. 3, pp. 817–823, 1998.