Optimizing Robustness and Label Efficiency in Hybrid Quantum Binary Classifiers via Active Learning Strategy Proportion

Guilherme Alves
Physics Department
University of Coimbra
Coimbra, Portugal
guilhermemalves2003@gmail.com

Luís Silva
Informatics Engineering Department
University of Coimbra
Coimbra, Portugal
eduardosilvasantos03@gmail.com

***Abstract*—Active Learning (AL) in quantum machine learning is an ongoing field that promises significant advancements in label efficiency and noise mitigation in the NISQ era. In this paper, we systematically investigate the impact of different sampling proportion strategies using Uncertainty Sampling on the efficiency and robustness of a Hybrid Quantum Binary Classifier. We empirically demonstrate that balanced strategies, specifically the $(2P+8A)$ and the $(8P+2A)$ configurations, are highly effective. The $(2P+8A)$ strategy maximizes label efficiency, requiring up to 40% fewer samples than the passive baseline. Crucially, the $(8P+2A)$ strategy offers the best trade-off, providing high efficiency while maintaining superior robustness (lowest standard deviation, σ) for reliable training on noisy NISQ devices.**

***Index Terms*—Active Learning, Quantum Computing, Machine Learning, Noise Mitigation**

I. Introduction

A. Quantum Computing

Quantum computing, introduced by Feynman, is an emerging area in physics and computer science. Many attempts to apply machine learning in quantum computing algorithms have been made over the years, but quantum systems are computationally expensive and possess the significant disadvantage of high levels of noise when scaling to large projects.

We live in the era of NISQ (Noisy intermediate-scale quantum), introduced by Preskill [1], where even the most advanced quantum computers in the world are unstable when scaling. This instability is due to quantum gate noise, we have 2 approaches for a solution:

- Improve Hardware;
- Improve Algorithms.

When using quantum algorithms, data must be very simple: images and other types of "difficult" data must be processed into quantum states, and that requires many qubits to accomplish it. With that in mind, we can be sure of one thing: image classification is hard and expensive.

We can see those efforts, for example, in image processing and edge detection algorithms, with low resolution images requiring a significant number of qubits to achieve suboptimal results. A great paper by Mastriani [2] introduces the phenomenon "Quantum Hoax," whereby many papers do not take into account the necessary number of qubits for the algorithm, concealing significant disadvantages in this area, such as:

- Noise into quantum images destroys its information;
- Large operations makes decoherence impossible to treat

This motivates the use of hybrid algorithms. In this paradigm, the quantum computer serves as a co-processor, executing a specific quantum circuit and providing an expected value as an output. The classic computer runs the optimizer, which changes the weights and parameters of the quantum circuit. This type of hybrid algorithm is called Variational Quantum Algorithms (VQA), first introduced by McClean et al [3].

Specifically for classifiers, VQA uses parametric quantum circuits, mostly referred to as Quantum Neural Networks (QNN), introduced by Benedetti [4]. They are composed of two blocks: A Feature map that encodes classical data into quantum states and a trainable ansatz $U(\theta)$ which applies a series of parametric rotations. The objective of these classifiers is to find optimal θ parameters that transform the encoded state into a state whose expected value corresponds to the correct classification.

Despite the premise, QNNs in the NISQ era have two big problems:

- **Cost to measure**: Measuring the expected value by a reliable way needs to run the quantum circuit thousands of times;
- **Barren Plateaus**: Introduced by McClean [5], in deep circuits, gradients decay exponentially with the number of qubits, making the optimizer unable to find a descent direction thus preventing training.

Many attempts were made to mitigate noise and reduce the number of gates used in quantum systems, most of them hardware-based solutions. Improving and optimizing software can be very benefitial too. We propose the use of active learning in quantum machine learning systems.

B. Quantum Active Learning

Active learning is a sub-field of machine learning focused on label efficiency. There are different methods of active learning with each own advantages and disadvantages. An overview into different active learning methods were made by Xueying Zhan et al [6] stating that for binary classifiers, methods such as **Uncertainty Sampling** have good results. Beside the results, since in the quantum realm noise can diverge solutions, we want to see how robust this method is.

Uncertainty Sampling is to select, actively, the label that the model has the most uncertainty. This will provoke a great change in the parameters and can provoke chaos in the system but can also improve the efficiency.

Quantum Active learning is an already well-studied field, according to Ding et al [7], the authors took off 93% of the samples in a sample pool and got the same precision as with the whole pool.

While previous works in Quantum Active Learning (QAL) have primarily focused on demonstrating label efficiency, the impact of the sampling strategy proportion on the robustness of training in NISQ environments has not been systematically studied. This paper seeks to fill that gap, focusing not only on efficiency but on the stability and reproducibility (robustness) of the training process.

C. *Active Learning and HCAI Alignment*

Active Learning (AL) establishes a critical bridge between Quantum AI and the principles of Human-Centered Artificial Intelligence (HCAI), particularly in NISQ environments where reliability and sustainability are paramount [8]. HCAI systems must prioritize trustworthiness and human control. AL directly impacts these pillars by focusing on label efficiency and robustness.

Firstly, the **Robustness** of AL (low variance in results) is essential for human trust, ensuring the Quantum AI system operates predictably and consistently despite quantum noise, a core requirement for Trustworthy Quantum Machine Learning [9]. Secondly, the **Label Efficiency** of AL reduces the time and cost of training, making development and experimentation more accessible. This facilitates greater human control over the development process [10] and transitions VQA from an expensive research concept to a more sustainable and viable solution for real-world applications.

D. *Objectives*

Our goal in this paper is to quantify active learning efficiency in a binary classifier using the **MNIST** dataset. We decided to use a hybrid quantum classifier, where the quantum layer will process and classify the data and the classical layer will change the weights based on the results.

We define robustness as the standard deviation, σ, across multiple random simulations. This metric is crucial because the performance variability inherent to quantum systems (where simulations can start with widely different precision percentages) must be minimized. Therefore, ensuring the reproducibility of results, specifically achieving a consistent number of samples required to reach the target precision is a fundamental requirement for a robust system.

II. Materials and Methods

Our methodology consists on:

- Process data into quantum states;
- classify the data;
- based on the results, change the weights, w, and angles, θ;
- Repeat.

We will give 10 samples into the system for each step, making small moves to fine-tune. Also, 11 simulations will be done with the same samples:

1) 10 passive samples + 0 active samples
2) 9 passive samples + 1 active sample
3) 8 passive samples + 2 active samples ...
4) 1 passive sample + 9 active samples
5) 0 passive samples + 10 active samples

The simulations were done using a deprecated function **Estimator** from qiskit-aer library which only exists for Python 3.9 as the time of writing.

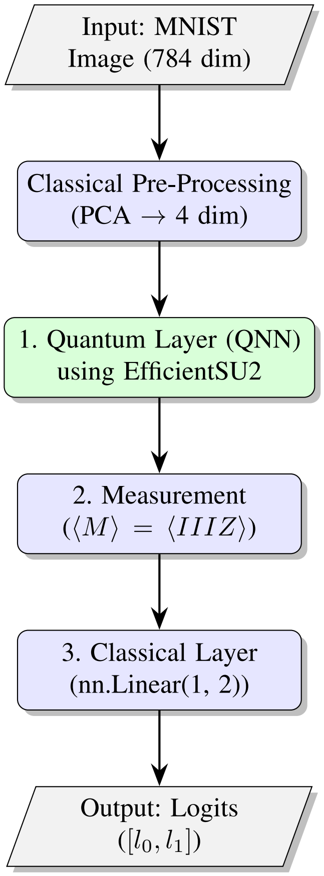
Input: MNIST Image (784 dim)
Classical Pre-Processing (PCA → 4 dim)
1. Quantum Layer (QNN) using EfficientSU2
2. Measurement ($\langle M \rangle = \langle IIIZ \rangle$)
3. Classical Layer (nn.Linear(1, 2))
Output: Logits ($[l_0, l_1]$)

Fig. 1. Diagram of a one step into the Hybrid Quantum binary classifier

A. *Data Processing*

Since our objective is to make a binary classifier, we decided to only use the following digits:

$$[0, 1]$$

MNIST samples are images of hand-written digits made of $28 \times 28 = 784$ pixels.

Having a low number of Qubits, we decided to apply PCA to lower the 784 features to only 4 features in order to use only 4 qubits.

Since we only want to classify and not interpret any details of each pixel, PCA (Principal Component Analysis) was the perfect approach. Despite this, PCA can introduce more noise into the system, a great active learning strategy must be robust.

After applying PCA we got a classic vector $\vec{x}$:

$$\vec{x} = (x_0, x_1, x_2, x_3)$$

To transform $|0\rangle^{\otimes 4}$ into the codified quantum state $|\psi(\vec{x})\rangle$ we need to apply the ansatz **ZFeatureMap**, introduced by Havlicek et al [11], associated with the classical vector, $U_\Phi(\vec{x})$:

$$U_\Phi(\vec{x}) = \left[\left(\prod_{i=0}^{3} U_{ZZ}(\vec{x})_{i,i+1}\right)\left(\bigotimes_{i=0}^{3} R_Z(2x_i)\right)\right] \cdot \left(\bigotimes_{i=0}^{3} H\right) \tag{1}$$

Where:

- $\bigotimes_{i=0}^{3} H$: Is the Hadamard gate, which creates a uniform superposition between the 16 states;
- $\bigotimes_{i=0}^{3} R_Z(2x_i)$: Is the Z-Rotation gate, which maps the feature x_i into the rotation angle $2x_i$.
- $\prod_{i=0}^{3} U_{ZZ}(\vec{x})_{i,i+1}$: Is the entanglement layer, applied to neighboring pairs (q_0-q_1, q_1-q_2, q_2-q_3), where U_{ZZ} is the gate R_{ZZ} with angle $\phi_{i,j}(x_i, x_j)$.

this will make the following transformation:

$$U_\Phi(\vec{x})|0\rangle^{\otimes 4} = |\psi(x)\rangle$$

B. Learning and Processing Layer

In this paper, we will use the **EfficientSU2** function, $U(\theta)$ for the system to separate 0's from 1's in the Hilbert Space.

This whole Anstaz is defined by the following operation:

$$U(\theta) = \prod_{l=1}^{L} E_l R_l(\theta_l) \tag{2}$$

where L is the number of layers defined by the user. In our simulations we decided:

$$L = 2$$

R_l which is the parametric rotation is defined by:

$$\mathbf{R}_l(\theta_l) = \bigotimes_{i=0}^{3} \left[R_Y(\theta_i^{(l)}) R_Z(\phi_i^{(l)})\right] \tag{3}$$

in which:

$$R_Y(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \tag{4}$$

$$R_Z(\phi) = \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix} \tag{5}$$

The $\mathbf{E}_l$ is defined by:

$$\mathbf{E}_l = CNOT_{2,3} \cdot CNOT_{1,2} \cdot CNOT_{0,1} \tag{6}$$

The classification layer can be visualized as:

$$|\psi(x)\rangle \to |\psi_{\textbf{final}}(x, \theta)\rangle$$

C. Measurement

To measure, we will use operator M:

$$M = I \otimes I \otimes I \otimes Z \tag{7}$$

in which:

- I is the identity operator;
- Z is the Z-Pauli operator.

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

the measurement will be the expected value of M, $\langle M \rangle$:

$$\langle M \rangle = \langle \psi_{\text{final}}(x, \theta)|M|\psi_{\text{final}}(x, \theta)\rangle \tag{8}$$

This operator will force the ansatz $U(\theta)$ to concentrate its information into the last qubit making the expected values always be in the interval:

$$[-1, 1]$$

Positive values will be highly associated with one class and negatives with the other. Values close to 0 will have maximum uncertainty (which we will use for the active learning process).

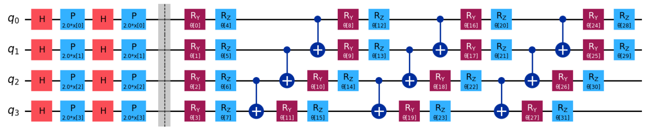
q_0
q_1
q_2
q_3
H
P
2.0*x[0]
2.0*x[1]
2.0*x[2]
2.0*x[3]
R_Y θ[0] θ[1] θ[2] θ[3]
R_Z θ[4] θ[5] θ[6] θ[7]
R_Y θ[11] θ[10] θ[9] θ[8]
R_Z θ[15] θ[14] θ[13] θ[12]
R_Y θ[19] θ[18] θ[17] θ[16]
R_Z θ[23] θ[22] θ[21] θ[20]
R_Y θ[27] θ[26] θ[25] θ[24]
R_Z θ[31] θ[30] θ[29] θ[28]

Fig. 2. Quantum circuit associated with the quantum layer, the enconding and the classificator are divided by a qc.barrier

D. Classical Post-Processing

$\langle M \rangle$ will be in the input into PyTorch.

Using a linear classical layer, **nn.Linear**:

$$logits = \begin{pmatrix} l_0 \\ l_1 \end{pmatrix} = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \langle M \rangle + \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \tag{9}$$

The logits are transformed into probabilities, p_k, using the function **Softmax** defined by:

$$p_k = \frac{e^{l_k}}{\sum_{j=0}^{1} e^{l_j}} \tag{10}$$

This probabilities are used to calculate the Cross-Entropy Loss, $\mathcal{L}$, which is the model's objective to minimize:

$$\mathcal{L} = -\sum_{k=0}^{1} y_k \cdot \log(p_k) \tag{11}$$

Which y_k is the real label.

E. Optimization

Our objective now is to find the parameters that minimize $\mathcal{L}$, consisting of Θ:

$$\Theta = \arg\min \mathcal{L} = \{ \underbrace{\theta}_{\text{Quantum (EfficientSU)}}, \underbrace{w_0, w_1, b_0, b_1}_{\text{Classical (nn.Linear)}} \}$$

For the quantum circuit parameters θ, the gradients $\partial\langle M\rangle/\partial\theta$ are analytically computed using the Parameter-Shift Rule [9]:

$$\frac{\partial\langle M\rangle}{\partial\theta_i} = \frac{1}{2}[\langle M\rangle(\theta_i + \frac{\pi}{2}) - \langle M\rangle(\theta_i - \frac{\pi}{2})] \qquad (12)$$

In contrast, the classical parameters (w, b) are optimized using standard automatic differentiation provided by the PyTorch autograd engine. The hybrid model is treated as a single differentiable computational graph, allowing gradients to flow from the classical loss function back through the linear layer via the chain rule:

$$\frac{\partial\mathcal{L}}{\partial w} = \frac{\partial\mathcal{L}}{\partial p} \cdot \frac{\partial p}{\partial \text{logits}} \cdot \frac{\partial \text{logits}}{\partial w} \qquad (13)$$

Finally, the Adam optimizer updates both quantum (θ) and classical (w, b) parameters simultaneously based on the full gradient vector $\nabla_\Theta\mathcal{L}$:

$$\Theta_{new} = \Theta_{old} - \eta \times Adam(\nabla_\Theta\mathcal{L}) \qquad (14)$$

where η is the learning rate, in which in our simulations we defined by:

$$\eta = 0.01$$

This value was selected to ensure slow updates between iterations. Rapid parameter changes can cause the system to diverge.

We initially tested various objectives, including comparing the number of samples required for each strategy to reach 90% precision. However, this objective proved unviable, as each simulation would consume an average of 16 hours. Given the high computational demands of the entire system, we decided that comparing the samples needed to reach 80% precision offered the optimal balance between precision and computational time (with each simulation averaging 40 minutes).

III. Results and Discussion

A. Results

Doing 6 simulations, using random seeds, we got the following results:

TABLE I
Results of the Simulation 1

Strategy (P + A)	Samples to reach 80%	(vs. Baseline)
(10P + 0A)	310	0 (0.0%)
(9P + 1A)	210	100 (32.3%)
(8P + 2A)	230	80 (25.8%)
(7P + 3A)	230	80 (25.8%)
(6P + 4A)	140	170 (54.8%)
(5P + 5A)	**100**	**210 (67.7%)**
(4P + 6A)	**100**	**210 (67.7%)**
(3P + 7A)	110	200 (64.5%)
(2P + 8A)	120	190 (61.3%)
(1P + 9A)	190	120 (38.7%)
(0P + 10A)	140	170 (54.8%)

TABLE II
Results of the Simulation 2

Strategy (P + A)	Samples to reach 80%	(vs. Baseline)
(10P + 0A)	380	0 (0.0%)
(9P + 1A)	**680**	**-300 (-78.9%)**
(8P + 2A)	240	140 (36.8%)
(7P + 3A)	400	-20 (-5.3%)
(6P + 4A)	220	160 (42.1%)
(5P + 5A)	220	160 (42.1%)
(4P + 6A)	390	-10 (-2.6%)
(3P + 7A)	270	110 (28.9%)
(2P + 8A)	220	160 (42.1%)
(1P + 9A)	160	220 (57.9%)
(0P + 10A)	**150**	**230 (60.5%)**

TABLE III
Results of Simulation 3

Strategy (P + A)	Samples to reach 80%	(vs. Baseline)
(10P + 0A)	370	0 (0.0%)
(9P + 1A)	120	250 (67.6%)
(8P + 2A)	260	110 (29.7%)
(7P + 3A)	140	230 (62.2%)
(6P + 4A)	160	210 (56.8%)
(5P + 5A)	220	150 (40.5%)
(4P + 6A)	**110**	**260 (70.3%)**
(3P + 7A)	210	160 (43.2%)
(2P + 8A)	190	180 (48.6%)
(1P + 9A)	**110**	**260 (70.3%)**
(0P + 10A)	120	250 (67.6%)

TABLE IV
Results of Simulation 4

Strategy (P + A)	Sample to reach 80%	(vs. Baseline)
(10P + 0A)	270	0 (0.0%)
(9P + 1A)	410	-140 (-51.9%)
(8P + 2A)	260	10 (3.7%)
(7P + 3A)	320	-50 (-18.5%)
(6P + 4A)	430	-160 (-59.3%)
(5P + 5A)	380	-110 (-40.7%)
(4P + 6A)	310	-40 (-14.8%)
(3P + 7A)	350	-80 (-29.6%)
(2P + 8A)	**200**	**70 (25.9%)**
(1P + 9A)	490	-220 (-81.5%)
(0P + 10A)	330	-60 (-22.2%)

TABLE V
RESULTS OF SIMULATION 5

Strategy (P + A)	Samples to reach 80%	(vs. Baseline)
(10P + 0A)	270	0 (0.0%)
(9P + 1A)	280	-10 (-3.7%)
(8P + 2A)	270	0 (0.0%)
(7P + 3A)	340	-70 (-25.9%)
(6P + 4A)	**160**	**110 (40.7%)**
(5P + 5A)	330	-60 (-22.2%)
(4P + 6A)	260	10 (3.7%)
(3P + 7A)	320	-50 (-18.5%)
(2P + 8A)	270	0 (0.0%)
(1P + 9A)	230	40 (14.8%)
(0P + 10A)	**160**	**110 (40.7%)**

TABLE VI
RESULTS OF SIMULATION 6

Strategy (P + A)	Samples to reach 80%	vs. Baseline)
(10P + 0A)	360	0 (0.0%)
(9P + 1A)	570	-210 (-58.3%)
(8P + 2A)	**310**	**50 (13.9%)**
(7P + 3A)	470	-110 (-30.6%)
(6P + 4A)	640	-280 (-77.8%)
(5P + 5A)	630	-270 (-75.0%)
(4P + 6A)	190	170 (47.2%)
(3P + 7A)	120	240 (66.7%)
(2P + 8A)	160	200 (55.6%)
(1P + 9A)	140	220 (61.1%)
(0P + 10A)	270	90 (25.0%)

TABLE VII
EFFICIENCY AND ROBUSTNESS ON EACH STRATEGY (AVERAGE OF THE 6 SIMULATIONS)

Strategy (P + A)	Samples to reach 80%	Standard Deviation
(10P + 0A)	326.7	45.7
(9P + 1A)	378.3	197.3
(8P + 2A)	261.7	25.4
(7P + 3A)	316.7	107.8
(6P + 4A)	291.7	184.1
(5P + 5A)	313.3	167.3
(4P + 6A)	226.7	104.7
(3P + 7A)	230.0	92.2
(2P + 8A)	193.3	46.8
(1P + 9A)	220.0	126.5
(0P + 10A)	195.0	77.2

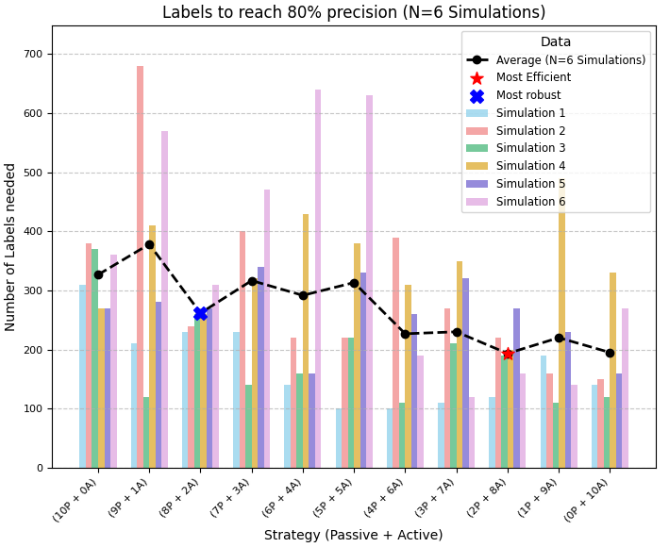

Fig. 3. Average Efficiency comparison across strategies highlighting the trade-off between passive and active learning.

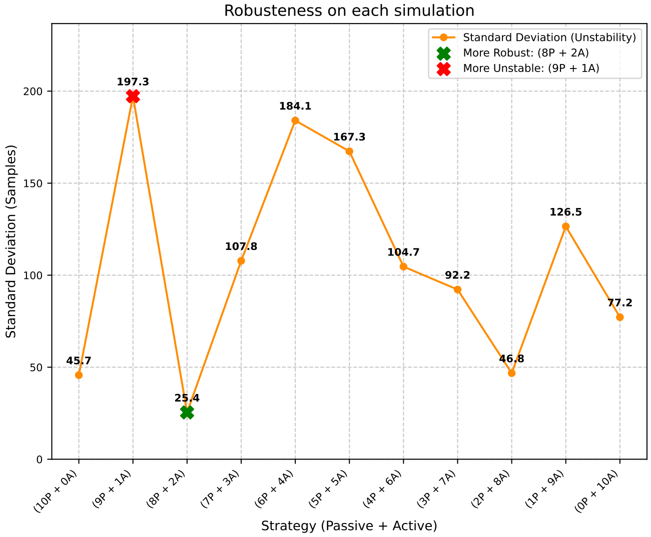

Fig. 4. Robustness of each strategy during 6 simulations

B. Discussion

1) Why does adding just 1 active sample (or 1 passive one) destabilize the training so much?: With 9 passive samples, most of them will likely be ”easy” to classify, so the average gradient will be relatively small and stable. The 1 active sample will be the hardest sample the model found. This one generates a very large gradient (and possibly in an opposing directly to the average of the other 9 samples).

2) Statistical Validation: Due to computational constraints limiting the study to $N = 6$ simulations per strategy, non-parametric tests were employed. A Kruskal-Wallis test showed no global difference across all 11 strategies ($p = 0.2221$), likely due to high variance in unstable configurations (e.g., 9P+1A). However, Mann-Whitney U pairwise comparisons revealed significant findings:

- **Efficiency**: The $(2P + 8A)$ strategy requires significantly fewer labels than the Baseline $(10P + 0A)$ ($p = 0.0039$, unilateral).
- **Robustness**: The $(8P + 2A)$ strategy also outperforms the Baseline in efficiency ($p = 0.0117$) while maintaining stability.
- **Differentiation**: A bilateral test ($p = 0.0367$) confirms that $(2P + 8A)$ and $(8P + 2A)$ are statistically distinct, supporting a trade-off between aggressive efficiency and stability.
- **Noise Confirmation**: The $(9P + 1A)$ strategy showed no statistical difference from the Baseline ($p = 0.8099$), confirming that single-sample active learning introduces variance without efficiency gains.

C. Limitations

The development of this work was confronted by the following major limitations:

*1) **Hardware**:*

- The Real Quantum Hardware available at our university offered a capacity of only 3 qubits, making it impossible to implement the algorithm;
- Options such as Qiskit Runtime (connecting to real hardware via API) were not viable, as the free tier only provided 10 hours of computation each month for free

For these reasons, we opted to test the algorithm via simulation. Despite this, as noted by Piskor [12], these simulations are reliable. The main disadvantage of using simulations is the requirement for computational power and time.

*2) **Time constraints**:* Having limited time for developing the paper, we had a short time to develop more than 6 simulations.

*3) **PCA**:* To make this work viable, the use of Principal Component Analysis was fundamental. This introduced problems:

- Loss of Information
- Noise introduction

In an ideal situation, PCA would not be used, but this ideal situation is far from the reality of NISQ devices. From our point of view, this limitation turned out to be the perfect justification to use Active Learning.

D. Future Work

In the future, we propose the same active method using a hybrid QNN in different conditions:

- Perform more than 6 simulations (typically, thirty is the industry standard);
- Test the algorithm for higher precision targets;
- Evaluate the algorithm using different datasets.

IV. Conclusion

This study addressed the dual challenge of label efficiency and training stability in Hybrid Quantum Binary Classifiers within the NISQ era. Our investigation into active learning strategy proportions reveals that the choice of sampling strategy is not merely an optimization of speed, but a fundamental control mechanism for mitigating quantum noise.

A. The Efficiency-Robustness Trade-off

Our empirical results identify a distinct trade-off between aggressive learning and system stability. Statistical analysis ($p < 0.05$) confirms that mixed strategies significantly outperform the passive baseline.

- **Maximum Efficiency**: The $(2P+8A)$ strategy proved to be the most label-efficient, reducing the sample requirement by approximately 40% compared to the baseline. This makes it the ideal candidate for scenarios where labeling costs are prohibitive, provided that a higher variance in training trajectory is acceptable.
- **Superior Robustness**: Conversely, the $(8P+2A)$ strategy emerged as the optimal configuration for trustworthy AI. It maintains a statistically significant efficiency gain ($p = 0.0117$) while achieving the lowest standard deviation (σ) across simulations. This stability is critical for deploying quantum algorithms on real, noisy hardware, where reproducibility is paramount.

B. Technical Insights on Optimization

A critical technical insight emerged regarding the interplay between Uncertainty Sampling and gradient-based optimization. We observed that extreme active strategies (or single-sample active batches like $(9P + 1A)$ often destabilize the training process. This occurs because Uncertainty Sampling selects ”hard” examples that generate large, often outlier gradients. In small batches, these outliers skew the average gradient direction, causing the optimizer to diverge or oscillate. Therefore, retaining a ”passive buffer” (e.g., 80% passive samples) acts as a stabilizer, smoothing the optimization landscape while still benefiting from the informational value of active samples.

C. Implications for Human-Centered AI

From a Human-Centered AI (HCAI) perspective, these findings bridge the gap between theoretical quantum advantage and practical, trustworthy application. By quantifying robustness, we provide a path to build Trust-ensuring that quantum systems behave predictably despite inherent noise.

Simultaneously, by maximizing efficiency, we enhance Viability and Control, reducing the computational barriers that currently restrict quantum machine learning to elite research environments.

In conclusion, while fully active strategies are viable for rapid prototyping, we strongly recommend balanced strategies, specifically the $(8P + 2A)$ ratio, for deployment in NISQ devices. This approach ensures that the pursuit of efficiency does not compromise the reliability required for human-critical applications.

References

[1] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018. [Online]. Available: https://doi.org/10.22331/q-2018-08-06-79

[2] M. Mastriani, “Quantum image processing: the truth, the whole truth, and nothing but the truth about its problems on internal image representation and outcomes recovering,” 2020. [Online]. Available: https://arxiv.org/abs/2002.04394

[3] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, “The theory of variational hybrid quantum-classical algorithms,” *New Journal of Physics*, vol. 18, no. 2, p. 023023, Feb. 2016. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/18/2/023023

[4] 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, Nov. 2019. [Online]. Available: http://dx.doi.org/10.1088/2058-9565/ab4eb5

[5] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, “Barren plateaus in quantum neural network training landscapes,” *Nature Communications*, vol. 9, no. 1, Nov. 2018. [Online]. Available: http://dx.doi.org/10.1038/s41467-018-07090-4

[6] X. Zhan, H. Liu, Q. Li, and A. Chan, “A comparative survey: Benchmarking for pool-based active learning,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, ser. IJCAI International Joint Conference on Artificial Intelligence, Z. Zhou, Ed. International Joint Conferences on Artificial Intelligence, Aug. 2021, pp. 4679–4686, 30th International Joint Conference on Artificial Intelligence (IJCAI 2021), IJCAI-21 ; Conference date: 19-08-2021 Through 27-08-2021. [Online]. Available: https://www.ijcai.org/proceedings/2021/,https://ijcai-21.org/

[7] Y. Ding, Y. Ban, M. Sanz, J. D. Martín-Guerrero, and X. Chen, “Quantum active learning,” 2024. [Online]. Available: https://arxiv.org/abs/2405.18230

[8] B. Shneiderman, *Human-Centered AI*. Oxford University Press, 2022.

[9] M. Papič, A. Auer, S. Pogorzalek *et al.*, “Trustworthy quantum machine learning: A roadmap for reliability, robustness, and security in the nisq era,” *arXiv preprint arXiv:2511.02602*, 2025.

[10] H. Liu, Y. Wang, W. Fan, X. Liu, Y. Li, S. Jain, Y. Liu, A. K. Jain, and J. Tang, “Trustworthy ai: A computational perspective,” *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 1, pp. 1–59, 2022.

[11] 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, p. 209–212, Mar. 2019. [Online]. Available: http://dx.doi.org/10.1038/s41586-019-0980-2

[12] T. Piskor, M. Schöndorf, M. Bauer, D. Smith, T. Ayral, S. Pogorzalek, A. Auer, and M. Papič, “Simulation and benchmarking of real quantum hardware,” 2025. [Online]. Available: https://arxiv.org/abs/2508.04483