RESOURCE-EFFICIENT VARIATIONAL QUANTUM CLASSIFIER

PETR PTÁČEK[1,2,*], PAULINA LEWANDOWSKA[2], AND RYSZARD KUKULSKI[2]

1. ABSTRACT

Quantum computing promises a revolution in information processing, with significant potential for machine learning and classification tasks. However, achieving this potential requires overcoming several fundamental challenges. One key limitation arises at the prediction stage, where the intrinsic randomness of quantum model outputs necessitates repeated executions, resulting in substantial overhead. To overcome this, we propose a novel measurement strategy for a variational quantum classifier that allows us to define the *unambiguous quantum classifier*. This strategy achieves near-deterministic predictions while maintaining competitive classification accuracy in noisy environments, all with significantly fewer quantum circuit executions. Although this approach entails a slight reduction in performance, it represents a favorable trade-off for improved resource efficiency. We further validate our theoretical model with supporting experimental results.

Keywords: quantum machine learning, variational quantum classifier, binary classification, quantum circuit, resource-efficient

2. INTRODUCTION

Quantum computing is a rapidly evolving field, with advances continually pushing the boundaries of classical computational power. This progress has generated increasing interest in quantum machine learning (QML) [1, 2, 3], the intersection of machine learning and quantum computing, where the crucial question is whether quantum resources can enable learning models that outperform their classical counterparts. Among the approaches explored in QML, hybrid variational algorithms have become especially prominent due to their compatibility with today's NISQ

[1] Department of Applied Mathematics, Faculty of Electrical Engineering and Computer science, VSB-Technical University of Ostrava, 17. listopadu 2172/15, 708 33 Ostrava, Czech Republic

[2]IT4Innovations, VSB - Technical University of Ostrava, 17. listopadu 2172/15, 708 33 Ostrava, Czech Republic

[*] *Corresponding author. E-mail:* petr.ptacek@vsb.cz

devices [4]. A particularly important example is the variational quantum classifier (VQC) [5, 6, 7], which combines parameterized quantum circuits with classical optimization to perform supervised learning tasks.

The general quantum classification typically involves encoding classical input data into quantum states known as *feature map*, transformation of quantum state to process the information *ansatz*, and finally, *measurements* are performed to extract classical outcomes that determine the predicted label. Each part of this process can contain trainable parameters, allowing the classifier to adapt to the specific dataset and classification task.

An encoding procedure, known as a feature map, serves to embed classical data into quantum states and to map classically non-separable data points into a higher-dimensional Hilbert space where they can become linearly separable. The degree of separation of quantum states has a significant impact on classifier performance, as discussed, for instance, in [8], where the authors investigate trainable feature maps and quantum random access coding for efficient and expressive quantum feature representations.

Once classical data are encoded into a quantum state, the quantum binary classifier can apply a parameterized quantum circuit, called ansatz, to transform the embedded data. The ansatz consists of alternating layers of parametrized single-qubit rotations and entangling gates. The choice and structure of the ansatz affect the expressivity of the classifier, influence the required circuit depth, and impact the efficiency of optimization [9]. By adjusting the parameters iteratively through classical optimization routines, the ansatz can shape the quantum state in Hilbert space so that data points from different classes become even more distinguishable under measurement. The optimization routines used in QML can be broadly categorized into gradient-based methods, such as Adam [10, 11, 12], RMSProp [10, 11], and many modifications of the Gradient Descent [10, 11, 12, 13], and gradient-free stochastic approaches, including SPSA [5, 12, 14], COBYLA [12], and Nelder–Mead [12, 15, 16].

After the ansatz processes the encoded data, the qubits are measured in a chosen basis. Each measurement yields a classical outcome in the form of sample (a bitstring representing the state of measured qubits) drawn from the quantum state. A post-processing procedure, such as majority voting, parity checks, thresholding, or any method that determines the predicted label for a given measurement sample, must be applied. By repeating the measurement and postprocessing procedures multiple times—each repetition referred to as a shot—an empirical probability distribution over the possible class labels can be estimated.

Single-shot quantum classifier. While much prior research has emphasized difficulties in the training phase, such as barren plateaus [17], the decision stage also

introduces its own complications. Because measurement outcomes in quantum mechanics are inherently probabilistic, producing deterministic predictions requires repeated circuit executions. For classification tasks, this measurement overhead can become substantial, raising questions about the practical efficiency of VQCs algorithms in the presence of fault-tolerant quantum hardware [18, 19].

To handle this, there are many modifications of general classification setup. Taking into account the reduction of quantum resources, a paper [20] was recently published. Authors have introduced the *single-shot quantum classifier* for binary classification that the label of a data point is inferred from a single execution of the quantum circuit. To do this, several modifications to the postprocessing of measurement results were introduced, along with the design of more sophisticated trainable feature maps. As finally stated, achieving quantum learning models cannot be single-shot in a generic way and trainable at the same time. Authors also noted that achieving near-deterministic classification requires that input data be embedded so that samples from different classes are sufficiently separated along mutually orthogonal directions in Hilbert space which is really hard to achieve. Otherwise the classifier will not return the expected results. Hence, its performance is highly dependent on the input data and the specific classification problem. While such schemes have a potential to offer a resource-efficient methods, it sadly entails with one more significant disadvantage that is the lack of noise robustness.

Unambiguous quantum classifier. To address these challenges, we propose a novel measurement strategy for variational quantum classification (VQC) tasks. This approach enables us to define an unambiguous quantum classifier for binary classification by appropriately adjusting the measurement basis and the postprocessing procedure.

The postprocessing procedure operates as follows. We perform the binary classification task once, and the possible measurement outcomes are labeled as class ”0”, class ”1”, or ”I don’t know”. If the outcome is 0 or 1, we classify the data according to the corresponding label. However, if the outcome is ”I don’t know”, we repeat the experiment, increasing the number of measurement shots by one. This process continues until a definite outcome of 0 or 1 is obtained.

This innovative strategy is capable of making near-deterministic predictions maintaining adequate classification accuracy in noisy environment while requiring significantly fewer quantum circuit executions. The slight reduction in performance represents a favorable trade-off for resource efficiency. We further validate the theoretical model with supporting experimental results. The experiments show that with sufficient embedding, proper data preprocessing, postprocessing and training strategies (e.g., use of data re-uploading [21]), and proper choice of cost function, the trained *unambiguous quantum classifier* can achieve near-deterministic predictions with a reasonable trade-off. The slight decrease of average accuracy in noiseless setups was

around 6.25 %, and in the setups introducing noise, the decrease was around 7.2 %, indicating that the proposed method can be usable in noisy environments such as real quantum hardware.

This paper is organized as follows. Section 3 presents the mathematical description of the classification task with the general setup of the classifier. We determine the probability of successful classification for noiseless and noisy environments. Next, in Section 4 we present the unambiguous quantum classifier scenario, again first for a noiseless environment and next including the noise model. Section 5 is dedicated to experimental results that support theoretical considerations. Concluding remarks are presented in the final Section 6.

3. GENERAL CLASSIFIER SETUP

In this section, we will analyze the classification performance after training phase for different classifier setups.

Let us consider a N-qubit quantum system. Without loss of generality, let $N = 2k + 1$, $k \in \mathbb{N}$. Consider the quantum circuit after training phase. Let a data point that is of class $'0'$ be described by N-qubit quantum state σ. In general classification setup, the binary measurement which determines the class is taken as $\{|0\rangle\langle 0| \otimes \mathbb{1}, |1\rangle\langle 1| \otimes \mathbb{1}\}$ which measures the value of the first qubit. Then, the probability of successful classification in one-shot scenario equals

$$p_{succ} = \frac{1+\delta}{2}, \tag{1}$$

where δ defines the level of data separation after training. We will consider a noisy environment, where a noise channel $\mathcal{E}$ is defined as $\mathcal{E}(\rho) = (1-\epsilon)\rho + \epsilon\rho_*$, where ρ_* is the completely mixed state and $\mathcal{E}$ acts on each qubits. Then, to determine the impact of noise for classification task, we calculate

$$\mathrm{tr}\left(\mathcal{E}^{\otimes N}(\sigma)\,|0\rangle\langle 0| \otimes \mathbb{1}\right) = \langle 0|\,\mathcal{E}(\mathrm{tr}_{-1}(\sigma))\,|0\rangle = \frac{1+\delta}{2} - \frac{\delta}{2}\epsilon, \tag{2}$$

where $\mathrm{tr}_{-t}(\cdot)$ denotes the partial trace on all qubits without t-th qubit. Then, the minimal probability of successful classification with the level of separation δ is expressed by

$$p^{\epsilon}_{succ} = \frac{1+\delta}{2} - \frac{\delta}{2}\epsilon. \tag{3}$$

In this work, we consider a different setup by changing the measurement strategy. For now we will use the final measurement defined by

$$\Pi_r = \sum_{b:\rho_H(b,0)\leq N-r} |b\rangle\langle b|\,, \tag{4}$$

where $\rho_H(b,0)$ is the Hamming distance between bitstring b to all-zero string. Then, let us observe that the probability of successful classification can be calculated as

$$\mathrm{tr}\left(\mathcal{E}^{\otimes N}(\sigma)\Pi_r\right) = \mathrm{tr}\left(\Delta\mathcal{E}^{\otimes N}(\sigma)\Pi_r\right) = \mathrm{tr}\left(\mathcal{E}^{\otimes N}(\Delta(\sigma))\Pi_r\right), \tag{5}$$

where Δ is the completely dephasing channel. Observe that, opposite the original setup, the success probability is strictly related with the diagonal elements of σ.

Lets fix $\rho \coloneqq \Delta(\sigma)$. Imagine the scenario when during the training phase the algorithm puts the same weight for each bitstring, hence all bitstrings are treated as equally likely. In that case, the representation of the state to be classified is given as

$$\rho = \frac{1+\delta}{2}\frac{\Pi_{k+1}}{\mathrm{tr}(\Pi_{k+1})} + \frac{1-\delta}{2}\frac{\mathbb{1}-\Pi_{k+1}}{\mathrm{tr}(\mathbb{1}-\Pi_{k+1})}. \tag{6}$$

We achieve the probability of successful classification with the level of separation δ given by

$$p^{\epsilon}_{succ} = \frac{1+\delta}{2} - \frac{\binom{N}{k}(k+1)\delta}{2^N}\epsilon + \mathcal{O}(\epsilon^2). \tag{7}$$

Using Stirling's approximation [22], we have

$$p^{\epsilon}_{succ} \simeq \frac{1+\delta}{2} - \frac{\sqrt{k}\delta}{\sqrt{\pi}}\epsilon + \mathcal{O}(\epsilon^2). \tag{8}$$

During the learning process it is possible to achieve a marginal bitstrings when the structure of the problem is better caught by the algorithm. Due to this, we can receive the lifted probability of successful classification p^{ϵ}_{max}, which is equal to

$$p^{\epsilon}_{max} = \frac{1+\delta}{2} + \mathcal{O}(\epsilon^2), \tag{9}$$

and achieved for the quantum state

$$\rho = \frac{1+\delta}{2}\left|0,\ldots,0\right\rangle\!\left\langle 0,\ldots,0\right| + \frac{1-\delta}{2}\left|1,\ldots,1\right\rangle\!\left\langle 1,\ldots,1\right|. \tag{10}$$

Finally if we have access to T shots of given data point, then, for the multiple-shot scenario, the minimal success probability is

$$p^{T}_{succ} = \Theta\left(\sqrt{T}\frac{p_{succ}-\frac{1}{2}}{\sqrt{p_{succ}(1-p_{succ})}}\right), \tag{11}$$

where Θ is the cdf of the normal distribution. Analogically, we define $p^{T,\epsilon}_{succ}$ and $p^{T,\epsilon}_{max}$.

4. UNAMBIGUOUS QUANTUM CLASSIFIER

Recall that the unambiguous quantum classifier setup operates as follows. We perform the binary classification task using the final measurement described in Eq. (4). Possible measurement outcomes are labeled as class „0", class „1" or „I don't know". If the outcome is 0 or 1, we classify the data according to the corresponding label. However, if the outcome is „I don't know", we repeat the experiment, increasing the number of measurement shots by one. This process continues until a definite outcome of 0 or 1 is obtained.

Assume the following probabilities: p_0 for obtaining the label 0, p_1 for obtaining the label 1 and p_x as the probability of rejection of the label x and repeat the experiment. In this way, we define the unambiguous success probability p^u as $p^u = \frac{p_0}{p_0+p_1}$. Let T represent the number of shots necessary to accept the unambiguous procedure. The expectation of T is given by $\mathbb{E}(T) = \frac{1}{p_0+p_1}$.

Taking into account the lifted training, we can improve the success probability. Hence, the maximal success classification probabilities that can be achieved are $p_0 = \frac{1+\delta}{2} + \mathcal{O}(\epsilon^2)$ and $p_1 = \frac{1-\delta}{2} + \mathcal{O}(\epsilon^2)$ and then

$$p^{u,\epsilon}_{max} = \frac{1+\delta}{2} + \mathcal{O}(\epsilon^2), \tag{12}$$

with the expectation value $\mathbb{E}(T) = 1 + \mathcal{O}(\epsilon^2)$. It means that for lifted training we can achieve the same probability for as for general classification model in one shot scenario. However, it should be noted that the lifted training is hard to achieve using quantum technology, so let us now analyze the average case. Let l be a parameter $k+1 < l < N$, then

$$\begin{aligned} p_0 &= \frac{1+\delta}{2^N} \operatorname{tr}\left(\mathcal{E}^{\otimes N}(\Pi_{k+1})\Pi_l\right) + \frac{1-\delta}{2^N}\left(\mathcal{E}^{\otimes N}\left(\mathbb{1} - \Pi_{k+1}\right)\Pi_l\right) \\ &= \frac{1+\delta}{2^N}\operatorname{tr}(\Pi_l) + \mathcal{O}(\epsilon^2). \end{aligned} \tag{13}$$

Analogously, we calculate

$$p_1 = \frac{1-\delta}{2^N}\operatorname{tr}(\Pi_l) + \mathcal{O}(\epsilon^2). \tag{14}$$

Finally, we obtain the unambiguous probability of successful classification equals

$$p^{u,\epsilon}_{succ} = \frac{1+\delta}{2} + \mathcal{O}(\epsilon^2) \tag{15}$$

with the expectation value $\mathbb{E}(T) = \frac{2^{N-1}}{\operatorname{tr}(\Pi_l)} + \mathcal{O}(\epsilon^2)$. Observe that

$$\mathbb{E}(T) = \frac{2^{N-1}}{\operatorname{tr}(\Pi_l)} + \mathcal{O}(\epsilon^2) = \frac{2^{N-1}}{2^{N-1} - \binom{N}{k+1} - \binom{N}{k+2} - \ldots - \binom{N}{l-1}} + \mathcal{O}(\epsilon^2) \leq 2 + \mathcal{O}(\epsilon^2), \tag{16}$$

when l remains a relatively small constant distance from $k+1$. In summary, it means that we can successfully complete the classification procedure with the expected number of shots equals two.

5. EXPERIMENTAL RESULTS

In this section, we present the results of the experiment supporting the theoretical consideration of the unambiguous quantum classifier. The implementation is publicly available on GitHub at `https://github.com/Weliras/RESOURCE-EFFICIENT-VARIATIONAL-QUANTUM-CLASSIFIER`. All of our experiments are implemented in the Python PennyLane framework version 0.40.0 [10], which is commonly used in the field of QML. We utilized the C++ *lightning.qubit* performance-optimized state-vector simulator for training our QML models, while the *default.mixed* simulator was used for testing under both noiseless and noisy conditions. The implementation is carried out in a series of *Jupyter Notebooks*, which provide an effective environment for experimentation and data visualization.

To compare and evaluate our models against results reported in previous studies, we use three publicly available datasets. These datasets are chosen because they collectively represent a common use case that supports the timely diagnosis of severe health conditions. Specifically, ***Dataset A*** is designed for predicting heart failure, ***Dataset B*** focuses on diabetes prediction, and ***Dataset C*** is used for forecasting breast cancer. The selection of these datasets is further motivated by their suitability for modeling problems formulated as binary classification tasks, which are well aligned with the implementation of VQC models. The datasets used in this study are available at the following links: `https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data`, `https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database` and `https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic`.

We follow the preprocessing strategy described in [6]. Briefly, we isolate the target column, scale the original input features to a normalized range using sklearn's *StandardScaler*, and apply Principal Component Analysis (PCA) to select the most relevant features while reducing the dimensionality of the input data. These components represent linear combinations of the original features that capture the greatest variance within the dataset. The resulting PCA components are scaled to the range $[0, 1]$ using sklearn's *MinMaxScaler*. Since the original datasets contain many features and we use an angle embedding method which requires one qubit per encoded feature, we limit the number of features accordingly. The train and test datasets are split in 80:20 ratio, and the same preprocessing technique is applied to all datasets.

5.1. **Circuit's architecture.**

In all subsequent circuits and models, the number of qubits used will be denoted as k. The number k is set equal to the number of input data features $x_i = (x_i^{(0)}, \ldots, x_i^{(k-1)})$, where x_i is one data point from the whole

dataset. In the quantum circuits of our models, we utilize a layered ZZ-entangling non-trainable feature map $U_{FM}(x_i)$ as we can see in Fig. 1.

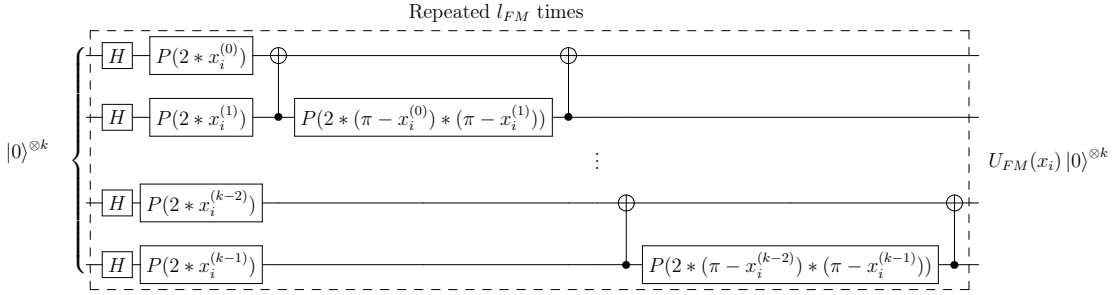

FIGURE 1. Quantum circuit of ZZ-entangling feature map $U_{FM}(x_i)$ for data point x_i of k data features. At the input, we consider a tensor product state of k qubits, each initialized in an default $|0\rangle$ quantum state. After single-qubits rotations $P(\varphi)$ of angle φ and two-qubits entangled gates CNOT we obtain the output $U_{FM}(x_i)\,|0\rangle^{\otimes k}$ with encoded data point x_i.

The feature map $U_{FM}(x_i)$ itself is consisting of l_{FM} repeating layers, allowing for a sufficiently deep embedding to effectively separate data points belonging to different classes. A study by Tomono and Natsubori [23] have highlighted the effectiveness of utilizing ZZ feature maps, demonstrating competitive performance scores in classification tasks compared to other types of feature maps, namely Y, Z-ZZ. The $U_{FM}(x_i)$ feature map applies controlled-NOT (CNOT) entangling gates between neighboring qubits interleaved with single-qubit rotations, allowing for effective nonlinear mappings of classical features into the Hilbert space.

We employ a layered trainable ansatz with reverse linear entangling, commonly used in variational algorithms for classification or chemistry applications. This ansatz, a heuristic trial wave function, is implemented in Qiskit as the *RealAmplitudes* class. The variational ansatz consists of l_A alternating layers of Y-rotations and CNOT entangling gates. Chandrasekhar et al. [24] have demonstrated the efficiency of this ansatz combined with the ZZ feature map, motivating our choice of the same ansatz. We can see the structure of used ansatz in Fig. 2.

5.2. **VQC classifier.** The models employ a hybrid parameterized quantum circuit composed of $U_{FM}(x_i)$ feature map and $U_A(x_i, \theta)$ ansatz. The resulting circuit $U_{VQC}(x_i, \theta)$ is depicted on Fig. 3.

5.3. **Postprocessing methods.** The output of the model can be interpreted in two ways. First we can sample each qubit in the Z-basis and obtain bitstrings, or we can estimate the expectation value of Pauli-Z operator on each qubit. For our models, both measurement methods are acceptable, and after postprocessing, they

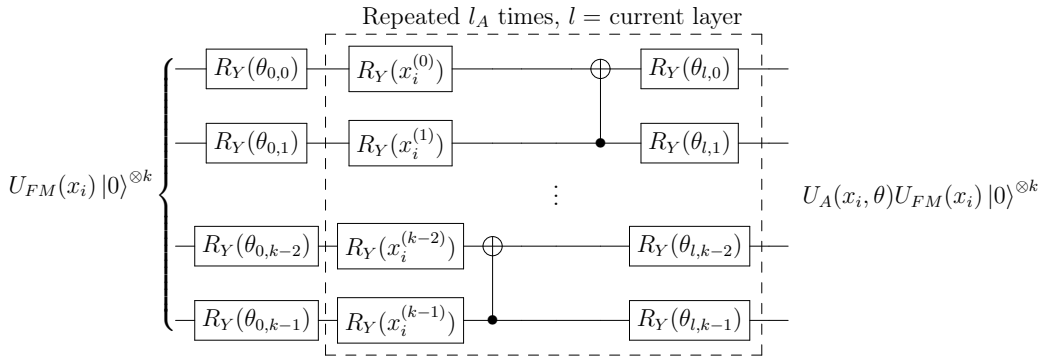

FIGURE 2. Parametrized circuit (ansatz) $U_A(x_i, \theta)$ with reverse-entangling and data re-uploading for trainable parameters $\theta_{l,i}$ for current layer l on i-th qubit. This ansatz implements data re-uploading [21], introducing the data point x_i to each layer. At the input, we consider an entangled state $U_{FM}(x_i)\,|0\rangle^{\otimes k}$. At the output we already have $U_A(x_i, \theta)U_{FM}(x_i)\,|0\rangle^{\otimes k}$, with processed quantum state by ansatz.

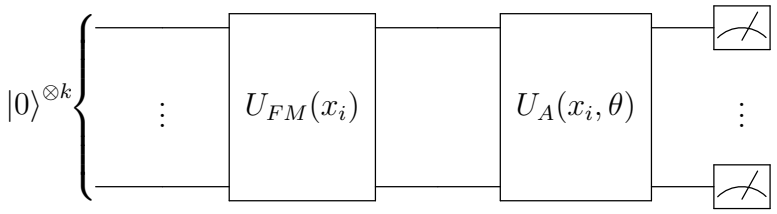

FIGURE 3. VQC binary classifier circuit $U_{VQC}(x_i, \theta)$ composed of layered feature map $U_{FM}(x_i)$ with l_{FM} layers and ansatz $U_A(x_i, \theta)$ with l_A layers. Input data point is depicted as x_i and trainable weights as θ. The output is obtained either by sampling each qubit in the Z-basis or by estimating the expectation value, which enables the use of gradient-based classical optimizers during training.

provide equivalent information for classification. However, measuring expectation values offers a major advantage over sampling bitstrings, as it provides a smoother cost function, enabling the use of gradient-based optimizers within the PennyLane framework. In that case, we are going to work with measured expectation values on each qubit, which are single float values in the range $[-1, 1]$.

In our experiments, we implement and evaluate three VQC models that use the same quantum circuit $U_{VQC}(x_i, \theta)$. Each model differs mainly in the implementation of the postprocessing procedure.

(A) **Many-Shots model (*M1*)**: In this model we take measured value only from the first qubit. As we aim to model a local cost and compare it

with subsequent models, the measurement result—when sampling bitstrings (0/1)—directly corresponds to the predicted label. Considering classification based on the expectation value, we include sign of the expectation value, where negative values correspond to class 0. The measurement procedure is repeated T times (where T is the number of shots) for each data point to estimate the probability of each label.

(B) **Many-Shots model (*M2*)**: In contrast to the previous model, measurements are taken from all qubits, allowing the approach to capture the global cost of the circuit. In this model, the measurement results—when sampling bitstrings—are mapped to predicted labels using the Hamming distance $\rho_H(b, 0)$ between the measured bitstring and the all-zero bitstring. When using expectation values, we sum the values from all qubits, and classification is determined based on the sign of this sum. The measurement procedure is repeated T times (where T is the number of shots) for each data point to estimate the probability of each label.

(C) **Unambiguous model (*M3*)**: In our novel approach we need to measure all qubits. The measurement results—whether obtained by sampling bitstrings or estimating expectation values—are interpreted in the same manner as in model *M2*. The main difference in this model is the introduction of an acceptance threshold, which must be satisfied by the circuit's results. The acceptance threshold specifies the number of identical values that must be obtained in a single shot attempt for it to be considered accepted. It means that if either the resulting Hamming distance or the sum of expectation values does not reach this threshold, then the measurement is repeated with the same setup. The procedure is repeated at most T_c times and $T_c \ll T$ until the unambiguous procedure accepts the measurement result and the label is determined by the first accepted measurement result and its predicted label. In the best-case scenario, the result is accepted on the first try, for the worst-case scenario, we define an additional parameter T_c that limits the maximum number of attempts. The expectation value, however, is upper-bounded $\mathbb{T} \leq 2 + \mathcal{O}(\epsilon^2)$ which is shown in Section 4.

5.4. **Training description.** Having described the full data preprocessing pipeline and model setup, we now turn to the description of our model training procedure. We trained the weights for models *M2* and *M3* using the postprocessing strategy of model *M2*, whereas the weights for model *M1* were trained using its corresponding *M1* postprocessing strategy. We explore a variety of different classical optimizers such as PennyLane native ones: *AdagradOptimizer*, *AdamOptimizer*, *RMSPropOptimizer*, *NesterovMomentumOptimizer* but also *SPSAOptimizer*. In the earliest experiments, we investigate the performance of non-native PennyLane optimizers, including Nelder–Mead and Particle Swarm Optimization, and confirm that they are

also applicable. In the end, we choose to use the *AdamOptimizer*. We use *Binary Cross Entropy* as the cost function. The number of training epochs is capped at 300, with early stopping applied once the cost function has converged. Within each epoch, the cost function is computed on batches of 8 input train data points. The entire training process is logged for subsequent analysis.

Following the training of model *M2* on the training data, its learned weights are applied to both models *M2* and *M3*, whereas the trained weights of model *M1* are used exclusively for model *M1*. This approach ensures a fair comparison of classification performance under different postprocessing strategies. Each model is evaluated and compared across all three datasets under both noiseless and noisy conditions. For each evaluation, 25 classification runs are performed on the full test dataset to collect statistics on evaluation metrics from each run, including the required number of quantum circuit executions, accuracy, precision, recall, F1 score, and ROC-AUC. The classification performed by the *unambiguous quantum classifier* (model *M3*) uses the following parameters: a maximum number of repetitions $T_C = 50$, and an acceptance threshold equal to the number of qubits, meaning that all measured values must be identical to be accepted.

5.5. **Noise model description.** We model realistic quantum noise by explicitly introducing standard quantum error channels. The noise is introduced using the PennyLane transform function applied to a quantum circuit, ensuring that the noise channels were inserted immediately after each quantum operation. Specifically, for each qubit on which any operation is done, we sequentially applied:

(1) Depolarizing noise (in PennyLane: *DepolarizingChannel* class) with probability 0.02, modeling the loss of information due to random flips in the quantum state.
(2) Amplitude damping (in PennyLane: *AmplitudeDamping* class) with probability 0.05, capturing energy relaxation processes in which the qubit state decays from the excited state $|1\rangle$ to the ground state $|0\rangle$.
(3) Phase damping (in PennyLane: *PhaseDamping* class) with probability 0.03, representing decoherence that reduces off-diagonal elements of the quantum state density matrix without energy loss.

Formally, if $\mathcal{E}_{\text{depol}}$, $\mathcal{E}_{\text{amp}}$, and $\mathcal{E}_{\text{phase}}$ denote the respective noise channels, for each operation O acting on qubit i, the noisy operation sequence becomes

$$O_i \longrightarrow \mathcal{E}^{(i)}_{\text{depol}} \longrightarrow \mathcal{E}^{(i)}_{\text{amp}} \longrightarrow \mathcal{E}^{(i)}_{\text{phase}}. \tag{17}$$

5.6. **Results.** In Fig. 4 and Fig. 5, we compare the models in terms of validation metrics and the number of quantum circuit executions required to make a prediction for a single test data point using 3 and 5 qubits, respectively. These figures illustrate results for *Dataset C*, while the result's comparition for all datasets are provided in Appendix A. The figures present statistics obtained from 25 classification runs per

model in both noiseless and noisy scenarios. The first row of Fig. 4 and Fig. 5 compares models classification performance using scoring metrics such accuracy, precision, recall. The second row, whereas, compares the number of executions each model requires to make a near-deterministic prediction. The three columns represent the results per individual models, respectively. The solid curves depict outcomes from the noiseless environment, whereas the dashed curves illustrate results obtained under noisy conditions.

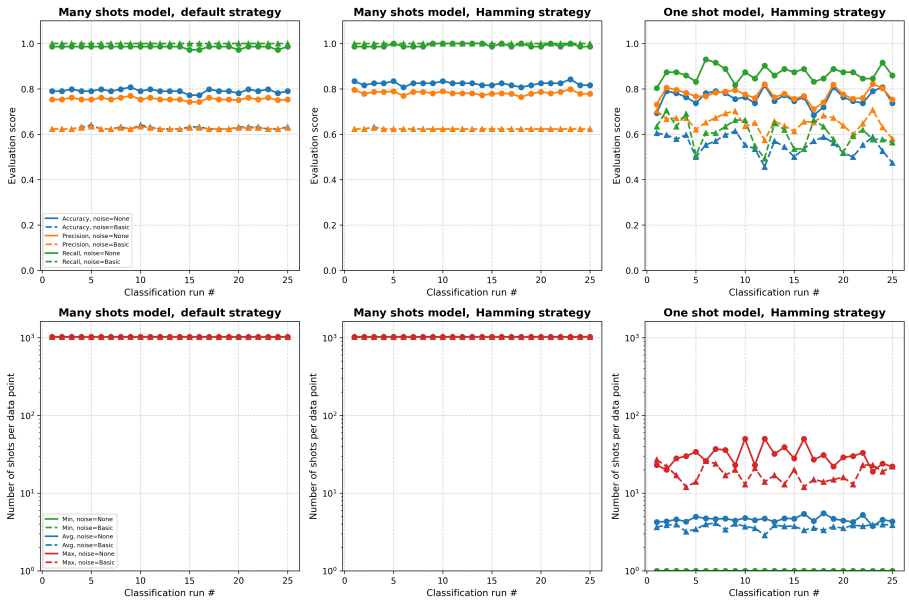

FIGURE 4. Comparison of classification performance and the number of executions required to achieve a near-deterministic prediction for each model using a circuit with 3 qubits, 2 ansatz layers, and 1 feature map layer on *Dataset C*.

From Fig. 4 (3 qubits), we can see that in the noiseless setup, model *M2* achieves the highest average accuracy of 82.25 %, model *M1* achieves an average accuracy of 79.05 %, while our *unambiguous quantum classifier*, model *M3*, attains a slightly lower accuracy of 76 %, representing a decrease of only 6.25 % compared to model *M2*. Model *M1* is performing worse than *M2*, confirming our our theoretical prediction that the global cost function performs better, as more data are obtained from measurements and available for postprocessing. All while models *M1* and *M2* needs 1024 executions for prediction, while the model *M3* needs on average only 4.61 executions. For *Dataset C*, this means that models *M1* and *M2* require a total of 116,736 executions to make predictions for the entire test set of 114 records, whereas model *M3* requires only approximately 525.54 executions. This means that

our *unambiguous quantum classifier* requires approximately 222× fewer executions than models *M1* and *M2*, while incurring only a 6.25 % drop in accuracy compared to model *M2*.

In the noisy setup, the accuracy of the best-performing model, *M2*, drops to 62.32 %, representing a decrease of nearly 20 % due to noise. This significant reduction can be attributed to the use of highly noisy channels with large error probabilities. The accuracy of model *M3* decreases by a similar margin of approximately 20 %. When comparing models *M2* and *M3*, the difference in accuracy is 7.20 %; however, model *M3* requires far fewer executions—only 3.69 on average. Overall, model *M3* performs approximately 277.5× fewer executions than models *M1* and *M2*.

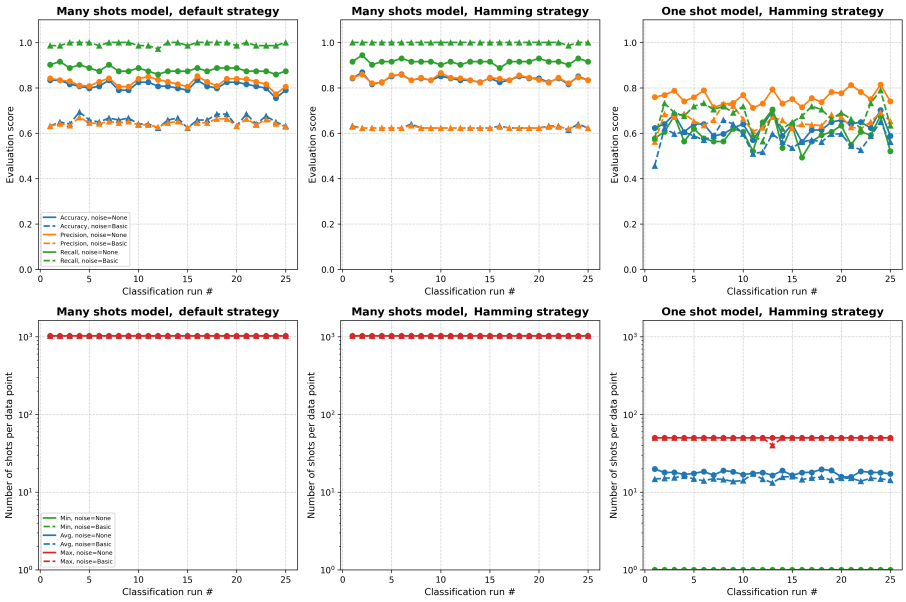

FIGURE 5. Comparison of classification performance and the number of executions required to achieve a near-deterministic prediction for each model using a circuit with 5 qubits, 2 ansatz layers, and 1 feature map layer on *Dataset C*.

When examining Fig. 5 (5 qubits), we can observe that model *M3* performs worse on 5 qubits compared to the 3-qubit configuration. Its average accuracy reaches 62.84 % in the noiseless setup and 57.58 % under noisy conditions. Table 1 presents the results in a tabular format, while the corresponding tables for the other datasets are provided in the appendix.

We compared our results with those reported in [6] and found that our models achieve even better performance than their VQC model, referred to as QNN. The comparison was conducted on the same datasets, using the identical feature map

Model	**Qubits**	Noise	**Avg. executions**	**ACC**	**PRE**	**REC**	**F1**
M1	3	NO	1024	0.790526	0.754565	0.983662	0.854008
M2	3	NO	1024	0.822456	0.584507	0.249600	0.348995
M3	3	NO	4.612982	0.760000	0.498914	0.208000	0.290289
M1	3	YES	1024	0.627368	0.625681	1.000000	0.769740
M2	3	YES	1024	0.623158	0.421821	1.000000	0.593325
M3	3	YES	3.685614	0.551228	0.410103	0.580800	0.479864
M1	5	NO	1024	0.808421	0.823892	0.881127	0.851451
M2	5	NO	1024	0.838246	0.584507	0.249600	0.348995
M3	5	NO	17.768772	0.628421	0.498914	0.208000	0.290289
M1	5	YES	1024	0.654386	0.644549	0.993239	0.781728
M2	5	YES	1024	0.625263	0.421821	1.000000	0.593325
M3	5	YES	14.897544	0.575789	0.410103	0.580800	0.479864

TABLE 1. Comparison of Model Performance on *Dataset C* with respect to average number of executions need to do prediction, Accuracy(ACC), Precision(PRE), Recall(REC) and F1 score.

and a similar ansatz configuration. This improvement can likely be attributed to the use of data re-uploading techniques, as well as experimentation with different optimizers, their hyperparameters, and batch sizes.

We also went further and trained our *unambiguous quantum classifier* to evaluate how well it could optimize the weights under the constraint of using as few executions as possible. We compared it with the training that we normally employ on model *M2* and the comparison can be seen in Fig. 6 and Fig. 7. For this experiment, we used the PennyLane *SPSAOptimizer*, as training model *M3* involves significant noise due to the inherent randomness of quantum measurements and the constraint on the number of executions. We can see that model *M3* is capable of training itself; however, this approach introduces noise into the training process, causing the cost function to oscillate. Despite this, the optimizer successfully minimizes the cost and achieves convergence.

6. CONCLUSION AND DISCUSSION

In this work, we introduced the *unambiguous quantum classifier*, a measurement strategy designed to mitigate the stochastic overhead inherent in quantum classification. By enforcing an acceptance threshold on measurement outcomes, our approach achieves near-deterministic predictions while drastically reducing the number of required quantum circuit executions. Experimental validation across three publicly available datasets demonstrated that, in a 3-qubit noiseless setup, the unambiguous classifier attains an average accuracy of 76%, only 6.25% below the best-performing

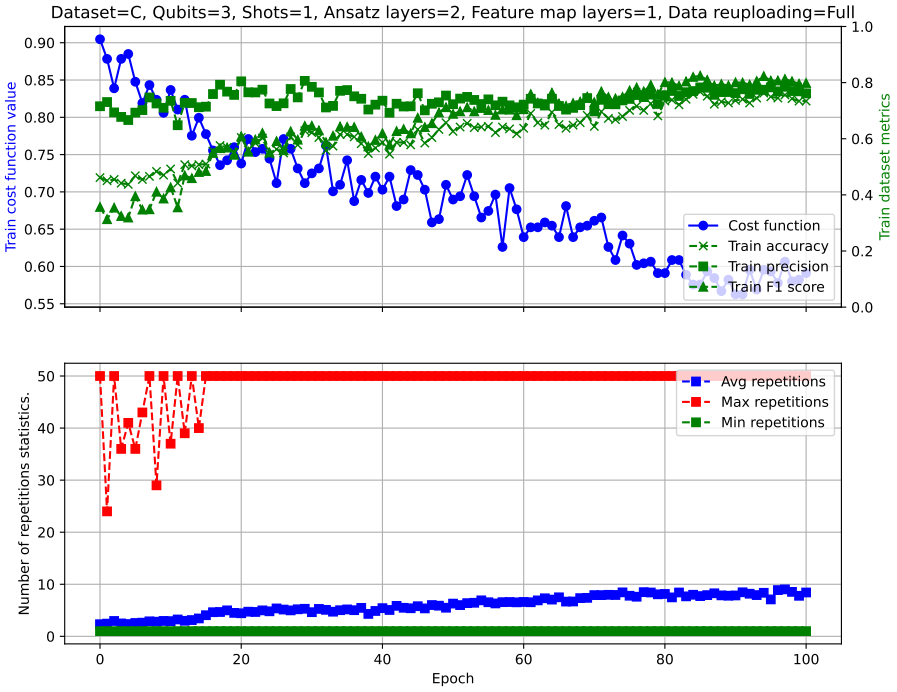

FIGURE 6. Training history using *Dataset C* and number of executions in each epoch for model *M3* on 3 qubits with 2 ansatz layers, 1 feature map layer.

baseline (model *M2* at 82.25%), while requiring approximately 222× fewer circuit executions on average. Under noisy conditions, the performance gap remains modest, around 7%, while the reduction in required executions remains comparable, at roughly 277×.

The use of expectation-value-based measurements enables smoother cost functions and efficient gradient-based optimization. Additionally, the incorporation of data reuploading facilitates faster convergence during training. Although the unambiguous classifier introduces a slight trade-off in accuracy, it provides substantial gains in computational efficiency, making it a compelling approach for near-term quantum hardware where circuit execution cost is the dominant constraint. Furthermore, we have shown that the model can self-optimize within its own constraints using stochastic optimizers such as SPSA, achieving convergence even under higher noise levels.

To further evaluate the significance of our approach, we compared our models' performance with results reported in recent studies, such as the VQC model from [6]. Across the same datasets and comparable circuit configurations, our models consistently achieved superior performance. This improvement can be attributed to the

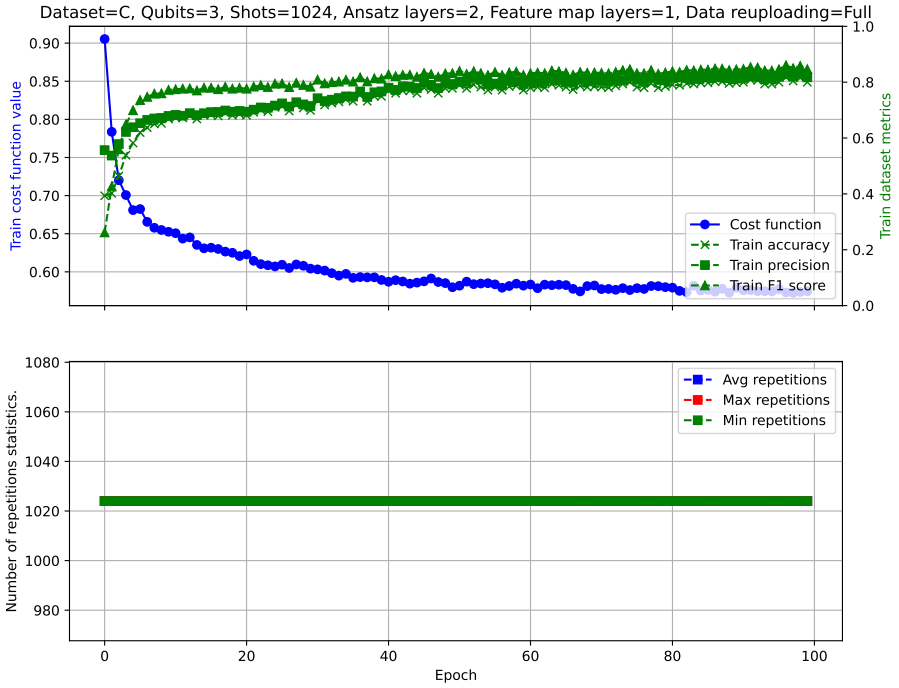

FIGURE 7. Training history using *Dataset C* and number of executions in each epoch for model *M2* on 2 qubits, with 2 ansatz layers, 1 feature map layer and 1024 shots.

data re-uploading strategy, optimized hyperparameters, and the use of alternative optimizers.

Overall, the proposed *unambiguous quantum classifier* demonstrates that it is possible to significantly reduce the stochastic overhead inherent to quantum measurements without severely compromising accuracy. By introducing a measurement strategy that yields near-deterministic outcomes, this approach provides a practical pathway toward more resource-efficient quantum machine learning models. The balance it achieves between performance and execution cost highlights its potential to enhance the scalability and applicability of variational quantum classifiers in realistic, noisy quantum computing environments.

Future research could focus on extending the unambiguous quantum classifier to larger qubit systems and more complex datasets to evaluate its scalability and robustness in higher-dimensional Hilbert spaces. Investigating adaptive acceptance thresholds that dynamically adjust based on measurement statistics could further improve efficiency and accuracy. Additionally, integrating error mitigation and

noise-aware training strategies may enhance performance on current noisy intermediate-scale quantum (NISQ) devices. Finally, exploring hybrid classical–quantum architectures that combine the unambiguous classifier with classical post-processing or feature selection techniques could offer further gains in both interpretability and computational efficiency.

DATA AVAILABILITY STATEMENT

The data that support the findings of study are openly available in Github repository at `https://github.com/Weliras/RESOURCE-EFFICIENT-VARIATIONAL-QUANTUM-CLASSIFIER`.

ACKNOWLEDGMENTS

PP is supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254) and by a Grant of SGS No. SP2025/049, VŠB - Technical University of Ostrava, Czech Republic

PL is supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254), with the financial support of the European Union under the REFRESH – Research Excellence For REgionSustainability and High-tech Industries project number `CZ.10.03.01/00/22\_003/0000048` via the Operational Programme Just Transition.

RK is supported by the European Union under the Quantum error correction codes enhanced by reinforcement learning dedicated for the Ising model-based optimization, contract nr. 01906/2025/RRC via the Operational Programme Just Transition and Moravian-Silesian Region.

CONFLICT OF INTEREST

The authors declare no conflict of interest in this research.

AUTHOR CONTRIBUTION

All authors made equal contributions to the preparation of this paper. All authors wrote the manuscript. PP performed the methodology for the simulations and developed the software, as well as conducting visualization. PL and RK conceptualized the study, performed formal analysis, and supervised the project.

References

[1] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. Quantum machine learning. Nature. 2017;549(7671):195-202.

[2] Schuld M, Killoran N. Quantum machine learning in feature Hilbert spaces. Physical Review Letters. 2019;122(4):040504.

[3] Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, et al. Supervised learning with quantum-enhanced feature spaces. Nature. 2019;567(7747):209-12.
[4] Preskill J. Quantum computing in the NISQ era and beyond. Quantum. 2018;2:79.
[5] Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, et al. Supervised learning with quantum-enhanced feature spaces. Nature. 2019;567(7747):209-12.
[6] Ajibosin SS, Cetinkaya D. Implementation and performance evaluation of quantum machine learning algorithms for binary classification. Software. 2024;3(4):498-513.
[7] Yu Z. Analyzing SARS CoV-2 Patient Data Using Quantum Supervised Machine Learning. bioRxiv. 2021:2021-10.
[8] Thumwanit N, Lortaraprasert C, Yano H, Raymond R. Trainable discrete feature embeddings for variational quantum classifier. arXiv preprint arXiv:210609415. 2021.
[9] Wu A, Li G, Wang Y, Feng B, Ding Y, Xie Y. Towards efficient ansatz architecture for variational quantum algorithms. arXiv preprint arXiv:211113730. 2021.
[10] Bergholm V, Izaac J, Schuld M, Gogolin C, Ahmed S, Ajith V, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. arXiv preprint arXiv:181104968. 2018.
[11] Qiu PH, Chen XG, Shi YW. Solving quantum channel discrimination problem with quantum networks and quantum neural networks. IEEE Access. 2019;7:50214-22.
[12] Pellow-Jarman A, Sinayskiy I, Pillay A, Petruccione F. A comparison of various classical optimizers for a variational quantum linear solver. Quantum Information Processing. 2021;20(6):202.
[13] Suzuki Y, Yano H, Raymond R, Yamamoto N. Normalized gradient descent for variational quantum algorithms. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE; 2021. p. 1-9.
[14] Malik AJ, Verma CS. On quantum computing and geometry optimization. bioRxiv. 2023:2023-03.
[15] Wiersema R, Zhou C, Carrasquilla JF, Kim YB. Measurement-induced entanglement phase transitions in variational quantum circuits. SciPost Physics. 2023;14(6):147.
[16] Cerezo M, Sone A, Volkoff T, Cincio L, Coles PJ. Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nature Communications. 2021;12(1):1791.
[17] McClean JR, Boixo S, Smelyanskiy VN, Babbush R, Neven H. Barren plateaus in quantum neural network training landscapes. Nature Communications. 2018;9(1):4812.
[18] Preskill J. Fault-tolerant quantum computation. Introduction to Quantum Computation and Information. 1998;213.
[19] Katabarwa A, Gratsea K, Caesura A, Johnson PD. Early fault-tolerant quantum computing. PRX Quantum. 2024;5(2):020101.
[20] Recio-Armengol E, Eisert J, Meyer JJ. Single-shot quantum machine learning. Physical Review A. 2025;111(4):042420.
[21] Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E, Latorre JI. Data re-uploading for a universal quantum classifier. Quantum. 2020;4:226.
[22] Leubner C. Generalised Stirling approximations to N! European Journal of Physics. 1985;6(4):299.
[23] Tomono T, Natsubori S. Shipping inspection trial of quantum machine learning toward sustainable quantum factory. In: PHM Society Asia-Pacific Conference. vol. 4; 2023. .
[24] Chandrasekhar S, Pokhrel SR, Singh N. Adapting Quantum Machine Learning for Energy Dissociation of Bonds. arXiv preprint arXiv:251006563. 2025.

Appendix A. Result's comparison

In this appendix, we describe result's comparison for *Dataset B* followed by the convention presented in the main text. Fig. 8 presents the results obtained from *Dataset B - 3 qubits*, whereas Fig. 9 describes the experiment using *Dataset B - 5 qubits*. Fig. The results obtained for *Dataset A* are similar to those obtained for *Dataset B*; therefore, we decided not to include the corresponding plots and tables here.

Table 2 represent the comparison of model performance on *Dataset B*.

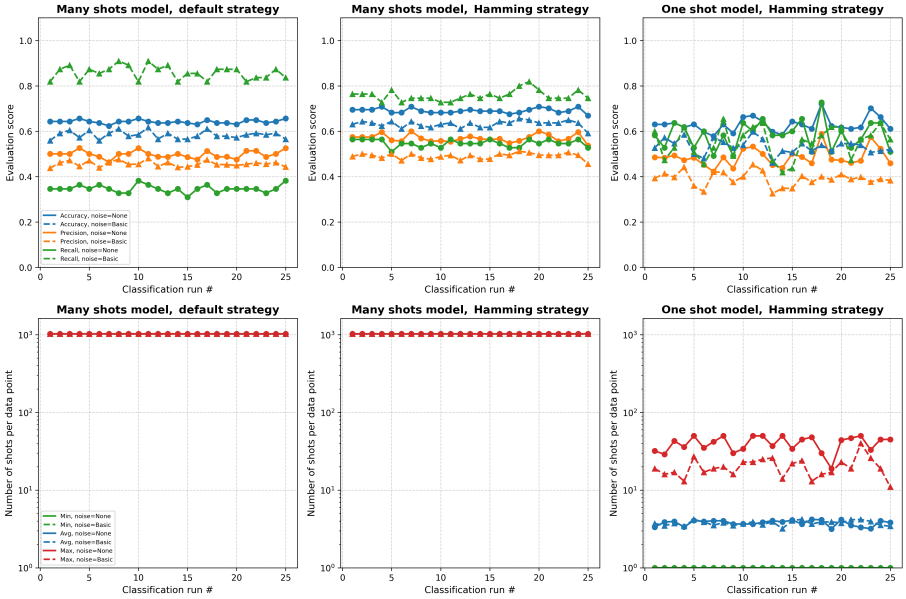

Figure 8. Comparison of classification performance and the number of executions required to achieve a near-deterministic prediction for each model using a circuit with 3 qubits, 2 ansatz layers, and 1 feature map layer on *Dataset B*.

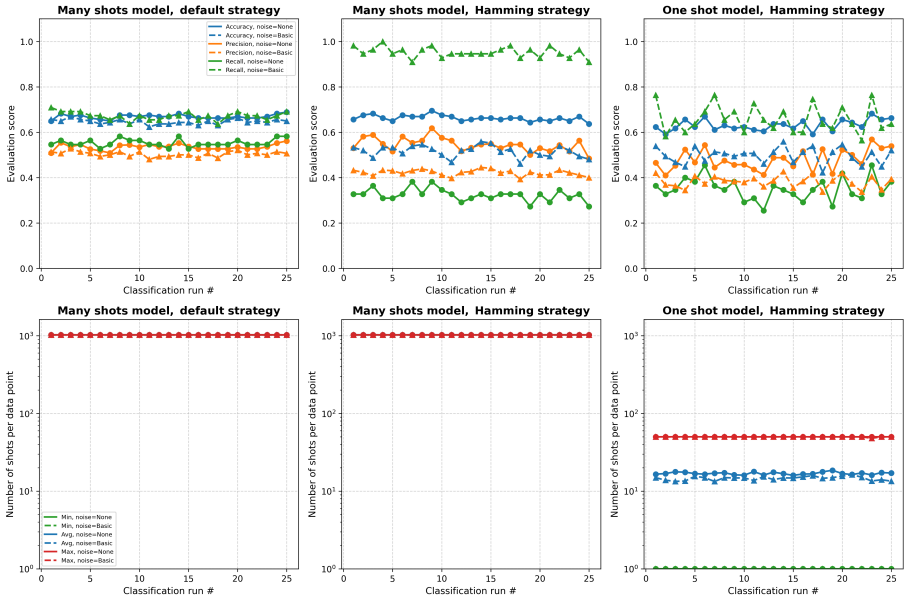

FIGURE 9. Comparison of classification performance and the number of executions required to achieve a near-deterministic prediction for each model using a circuit with 5 qubits, 2 ansatz layers, and 1 feature map layer on *Dataset B*.

Model	**Qubits**	Noise	**Avg. executions**	**ACC**	**PRE**	**REC**	**F1**
M1	3	NO	1024	0.641558	0.497477	0.346182	0.408101
M2	3	NO	1024	0.690130	0.584507	0.249600	0.348995
M3	3	NO	3.796623	0.629610	0.498914	0.208000	0.290289
M1	3	YES	1024	0.583377	0.455868	0.858182	0.595316
M2	3	YES	1024	0.631429	0.421821	1.000000	0.593325
M3	3	YES	3.782857	0.532727	0.410103	0.580800	0.479864
M1	5	NO	1024	0.668312	0.534555	0.552727	0.543404
M2	5	NO	1024	0.662078	0.584507	0.249600	0.348995
M3	5	NO	16.945195	0.631429	0.498914	0.208000	0.290289
M1	5	YES	1024	0.646234	0.503592	0.672000	0.575682
M2	5	YES	1024	0.515584	0.421821	1.000000	0.593325
M3	5	YES	14.624416	0.497403	0.410103	0.580800	0.479864

TABLE 2. Comparison of Model Performance on *Dataset B* with respect to average number of executions need to do prediction, Accuracy(ACC), Precision(PRE), Recall(REC) and F1 score.