IOP Publishing Mach. Learn.: Sci. Technol. 6 (2025) 045033 https://doi.org/10.1088/2632-2153/ae16fc

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
27 July 2025

REVISED
20 October 2025

ACCEPTED FOR PUBLICATION
23 October 2025

PUBLISHED
4 November 2025

PAPER

QEMOS: a scalable quantum error mitigation method to overcome qubit sensitivity

Zheng Tu, Jinchen Xu, Xin Zhou, Yu Zhu, Yi Liu, Qiming Du, Hang Lian and Bei Zhou*

Information Engineering University, Zhengzhou 450001, People's Republic of China
* Author to whom any correspondence should be addressed.

E-mail: beibei_0812@126.com

Keywords: quantum noise, quantum error mitigation, machine learning

Abstract

Compared with traditional computers, quantum computers can provide exponential acceleration for certain critical fields. However, the coupling of quantum systems with the environment, along with the intrinsic characteristics of quantum systems, has collectively introduced quantum noise, which has emerged as a significant impediment to the development of quantum computing. Quantum error mitigation (QEM) has been proposed as an alternative solution in the noisy intermediate-scale quantum era. In recent years, with the rise of artificial intelligence, machine learning-based QEM technology has received attention from the industry. However, the latest machine learning-based QEM techniques have limitations, especially their inability to mitigate errors in the quantum circuits whose number of qubits exceeds the number of qubits in the training set, and their tendency to amplify noise when constructing feature sets. This paper proposes QEMOS, a novel random forest-based machine learning model that utilizes a new feature dataset incorporating quantum computer backend properties, with feature dimensionality reduction enabling decoupling from the number of qubits. The model is trained and tested using six different simulators from Qiskit and a real quantum computer tianyan-176. It is worth noting that this model overcomes the limitation of sensitivity to the number of qubits, which was the main problem of previous methods. When trained on 5–9 qubit circuits, the model achieves a probability of correct mitigation of 86.38% on 2–13 qubit circuits, though this efficacy is observed primarily for circuits exhibiting high-probability outputs and decreases as all output probabilities approach zero. Compared to the baseline, the model demonstrates a 31.74% error reduction on test sets with more qubits than the training set. On real quantum computer, testing shows an average error reduction of 67.5%.

1. Introduction

Quantum noise, stemming from phenomena such as qubit decoherence [1], gate errors, and measurement inaccuracies, can lead to significant deviations between the output of quantum computer and their theoretical predictions [2]. In the era of noisy intermediate-scale quantum (NISQ) [3] devices, quantum noise is inherently present even in medium-to-low-scale quantum circuits. At present, quantum noise has become one of the main reasons that hinder the development of quantum computing [4–7].To mitigate the quantum noise, various methods have been developed, including zero-noise extrapolation (ZNE) [8], probabilistic error cancellation (PEC) [9] and measurement error mitigation [10]. These techniques often require substantial classical and quantum resources [11], with the resource requirements increasing exponentially with the number of qubits in the circuit, thereby limiting the scalability of current quantum error mitigation (QEM) methods to larger-scale quantum circuits. As machine learning techniques find applications in various fields, they have also begun to be applied to QEM. Machine

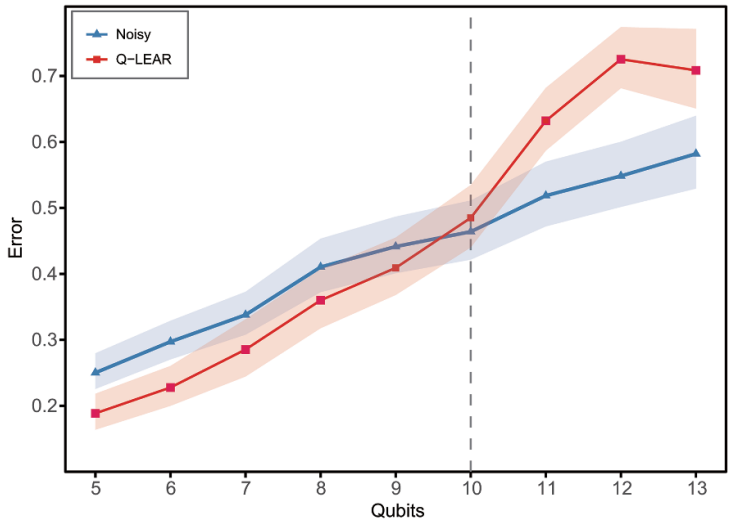

Figure 1. Testing the Q-LEAR model on circuits with 5–13 qubits, which is trained on circuits with 5–9 qubit. Circuit errors are statistically averaged by grouping according to the number of qubits, and the shaded area in the figure represents the 95% confidence interval.

learning-based QEM aim to reduce quantum noise through post-processing of circuit output, thereby significantly reducing the quantum resource overhead once the model is trained, and the classical processing time is on the order of microseconds.

However, recent machine learning-based QEM methods, such as Q-LEAR [12], exhibit limitations. Q-LEAR employs a multilayer perceptron (MLP) to construct an error mitigation model, utilizing depth-cut program error (Dpe), generated through circuit segmentation and reversed segment integration, as machine learning features. While Q-LEAR performs well on circuits with the same number of qubits as those in the training set, it fails to generalize effectively to circuits with a larger number of qubits, as illustrated in figure 1. This restricts their applicability to circuits that cannot be classically simulated, which is insufficient for NISQ-era devices, where the scale of qubit can reach levels that are classically intractable [3, 13].

In this paper, we propose a QEM method based on random forest model, called QEMOS, designed to address the limitations of current machine learning-based approaches. Our method combines features related to the fidelity of the quantum backend, gate set fidelity, and measurement error to estimate the noise in quantum circuits, and achieves decoupling from the number of qubits when constructing the feature data. Unlike Q-LEAR, our method does not introduce additional quantum noise into the circuits during the estimation process, thus requiring small amount of quantum resources. By leveraging a random forest model and incorporating fidelity and other feature data of qubits number decoupling, QEMOS effectively learns the feature of noise and generalizes well to the circuits whose number of qubits greater than those in the training set.

We trained and tested the QEMOS model using six different noisy simulators, and a quantum computer called tianyan-176. Our testing indicates that when generating random quantum circuits with 5–9 qubits in the training set, the model can achieve an average probability of 86.38% for correctly mitigating noise on circuits with 2–13 qubits. The noise mitigation performance on the quantum circuits whose number of qubits exceeds the number of qubits in the training set surpasses that of the Q-LEAR model by 31.74%. On real quantum computers, the average noise mitigation efficacy reaches 67.5%. These findings demonstrate that QEMOS can effectively mitigate quantum noise in circuits, including those with the number of qubits outside the training range.

2. Background

2.1. Fundamentals of quantum computing

2.1.1. Quantum bit

The quantum bit, or qubit, represents the smallest unit in quantum computing. Unlike classical bits in traditional computers, which can only be in the states of 0 or 1, qubits in quantum computing can exist in the states of 0, 1, or a superposition of both. In quantum computing, the state of qubits are typically

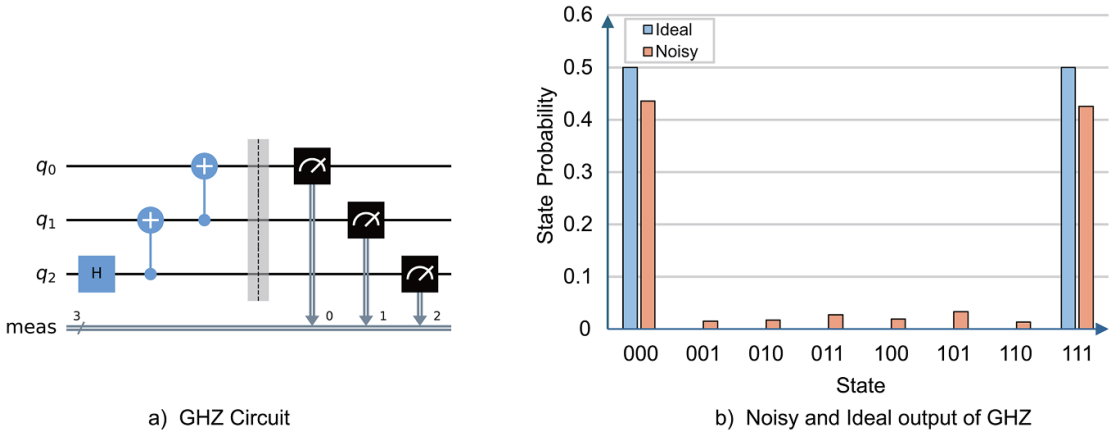

Figure 2. The quantum circuit and circuit output of three-qubit GHZ [14] quantum algorithms. (a) The quantum circuit diagram for the GHZ state preparation. (b) The theoretical (ideal) and actual (noisy) output probability distributions.

denoted by $|0\rangle$ and $|1\rangle$, and the state of a single qubit can be expressed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$,The expression $|\alpha|$ denotes the modulus of α.

2.1.2. Quantum gate

A quantum gate refers to an operation that alters the state of a qubit. These operations are categorized into single-qubit gates, two-qubit gates, and multi-qubit gates. For instance, the action of a Hadamard gate (H gate) is to transform a qubit in the ground state (GS) to a superposition state.

2.1.3. Quantum circuit

A quantum circuit is composed of qubits, a series of quantum gates, and a final measurement operation. The quantum gates act on the qubits in sequence, following the order of appearance from left to right, and the quantum circuit's result is obtained through measurement gates. The initial state of all qubits is usually $|0\rangle$.The quantum circuit ultimately yields a result in the form of a dictionary. An illustrative of a three-qubit quantum circuit is depicted in figure 2(a).

2.2. Quantum compiling

Quantum compiling refers to the intricate process of translating a sophisticated quantum algorithm or quantum program language description into an efficient and physically executable sequence of instructions on a quantum computer [15]. Due to the distinctive qubit topology and supported gate sets of different quantum computer backends, the qasm quantum circuit must be tailored to the specific quantum computer in use, allowing the circuit to operate on the target machine. It's important to note that the physical topology, qubit quality, and environmental couplings vary among different quantum devices, resulting in distinct noise characteristics for quantum circuits on different quantum computers [16, 17]. As such, it is arduous to identify a universal model that can directly mitigate quantum noise across all machines. Instead, it is necessary to model and learn for each individual quantum machine, or for a model to be trained using several quantum computers. When utilized, it should only be tailored to the quantum computers involved in the training.

2.3. Quantum noise

In the NISQ era, the presence of quantum noise may lead to substantial deviations between the operational outcomes of quantum circuits and the anticipated results of researchers. This represents a principal impediment to the expansion of quantum computing to larger scales. The quantum noise is illustrated as depicted in figure 2(b). Quantum noise primarily stems from the following causes.

2.3.1. Environmental perturbations

Tian *et al* [17] quantum systems may become coupled to the external environment, giving rise to phenomena such as decoherence and relaxation. For instance, when an external quantum system interacts with a qubit, it transfers environmental noise to the qubit, leading to decoherence and relaxation of the qubit.

2.3.2. Disturbances between qubits

Resch and Karpuzcu [2] interactions among qubits can result in minute perturbations to the states of the qubits, consequently giving rise to deviations in the outcomes.

2.3.3. Imperfections in qubits and gates

Dobrovitski *et al* [18] superconducting qubits employ microwave pulses for gate operations, necessitating highly precise control signals. However, achieving perfection in practical implementation of such microwave pulse control methods is exceedingly challenging. As of now, there is no flawless approach to fabricate qubits, and the fidelity of qubits is not absolute. Moreover, the coherence time of qubits is finite, and over time, the quality of qubits deteriorates.

The noise in quantum circuits accumulates with increasing circuit depth and interactions between qubits. Consequently, as the quantum circuit grows deeper, involving more qubits and a greater number of quantum gates, the noise amplifies.

In the realm of classical computing and networking, noise is a persistent challenge. Typically, diverse error-correcting codes or transmission protocols are employed for error rectification and verification. Analogizing to classical computing, quantum computing has also cultivated error correction codes, initially denoted as surface codes [19]. Recently, IBM has proposed a novel error correction code termed LDPC code [20]. IBM's experiments have shown that utilizing surface codes for a 12-qubit circuit necessitates an additional 3000 qubits, an unattainable feat at present. Even LDPC codes require 288 qubits [20]. Therefore, in the NISQ era, researchers are seeking QEM as an alternative approach to achieve accurate quantum program outputs. QEM, to some extent, alleviates noise while permitting a certain level of noise without impeding researchers from obtaining the desired correct results.

2.4. Key terms

Below are explanations of several foundational concepts essential for understanding the methodologies outlined in this paper.

Circuit output state (State): this refers to the resultant state of a quantum circuit after execution. For instance, a three-qubit quantum circuit may yield eight potential states: 000, 001, 010, 011, 100, 101, 110, 111. Each of these eight states is associated with a corresponding probability, denoted as Pro. The sum of these probability values equates to 1.

Output probability distribution (Pro): this refers to the probability distribution corresponding to the circuit output state. Each State is linked to its respective probability of occurrence, and the total probability distribution for a given circuit equals 1.

Circuit error: deviations exist between the output generated by execute a quantum circuit on a noisy quantum computer (either real hardware or a simulator) and the theoretical values, as depicted in the table in figure 2(b). We employ the Hellinger distance as the metric for assessing circuit error [12], and this method is used to evaluate errors for both the unmitigated noisy results and the mitigated results. This statistical measure is utilized to quantify differences between probability distributions, enabling the comparison of similarity between two probability distributions [21]. A Hellinger distance closer to 0 signifies greater similarity between two probability distributions. Assuming the probability distribution of the noisy outcome state is represented as $\overrightarrow{pro}_{\text{noisy}}$ (or the mitigated outcome $\overrightarrow{pro}_{\text{mit}}$)and the ideal outcome's probability distribution as $\overrightarrow{pro}_{\text{ideal}}$, the formula for evaluating circuit error using Hellinger distance is:

$$\text{HLD}\left(\overrightarrow{pro}_{\text{noisy}}, \overrightarrow{pro}_{\text{ideal}}\right) = \frac{1}{\sqrt{2}}\left|\left|\sqrt{\overrightarrow{pro}_{\text{noisy}}} - \sqrt{\overrightarrow{pro}_{\text{ideal}}}\right|\right| \tag{1}$$

State error: this refers to the absolute difference between the probability with noise, denoted as pro_{noisy}, associated with a particular state solution State, and the ideal probability pro_{ideal}:

$$error_{\text{state}} = \left|pro_{\text{noisy}} - pro_{\text{ideal}}\right|. \tag{2}$$

3. Related work and their limitations

QEM techniques are broadly classified into two categories: weak error mitigation and strong error mitigation [22]. Weak error mitigation targets the expectation values of quantum observables and represents the most common approach currently, with examples including ZNE and PEC. Strong error mitigation directly corrects the probability distribution of measurement outcomes in the computational basis.

So far, the methods claimed to be effective for strong error mitigation include Qraft and its improved version Q-LEAR. Although measurement error mitigation can also be used for strong error mitigation, the field currently primarily employs it for weak error mitigation. The subsequent analysis focuses on ZNE [8], PEC [9], and Q-LEAR [12].

3.1. Prevalent weak error mitigation

3.1.1. ZNE

ZNE believes that the expectation value of a quantum computation can be modeled as a multi-parameter function of the fault rate γ. By running the circuit at different noise levels and fitting the data to this function, one can extrapolate to the zero-noise limit. Many studies have extended the ZNE method using different approaches [23–26]. However, this method has several limitations:

Resource intensity: ZNE requires multiple executions of the circuit under varying noise conditions, which demands significant quantum resources.
Time dependence: ZNE assumes time-independent noise, which is not always the case in real-world quantum devices, making it difficult to apply historical noise models to mitigate current noise [27].

3.1.2. PEC

PEC models noise as a set of additional operations that precede the ideal quantum operations, which can be represented by a noise matrix. The noise effects are then mitigated by inverting this matrix. Over the past few years, significant research efforts have been devoted to developing PEC techniques [28–30]. However, this approach suffers from several key limitations:

Resource intensity: PEC requires extensive classical and quantum resources to learn the noise model, limiting its application to small-scale circuits.
Complete noise understanding: PEC requires a complete understanding and modeling of the noise, which may be impractical in complex quantum systems.

3.2. Machine learning-based strong error mitigation

Recent advancements in machine learning have led to the development of techniques specifically designed for QEM [31]. IBM introduced four machine learning models in 2023 for weak error mitigation [32]. However, these methods are limited in their applicability to specific types of quantum algorithms, such as the Bernstein–Vazirani algorithm [33], the Deutsch–Jozsa algorithm [34], and Grover's algorithm [35], which require the state probability distribution rather than just the expectation values of quantum observables. Quantum algorithms that target the probability distribution of output states are extensively utilized in finance, healthcare, biology, and information security [36, 37]. Prominent examples include quantum Monte Carlo methods, discrete logarithm finding, and large integer factorization [38, 39], among others. These algorithms are characterized by circuit output states containing high-probability solutions. Consequently, the future development of quantum algorithms with high-probability state may indeed be the primary avenue through which quantum algorithms transition into practical applications. This necessitates the implementation of strong error mitigation method.

The latest machine learning-based strong error mitigation method, Q-LEAR [12], estimates the noise in quantum circuits by cutting the circuit into segments and adding reversed segments to form Dpe. However, this approach has the following limitations:

Sensitivity to the number of qubits: Q-LEAR performs well on circuits with the same number of qubits as those in the training set but does not generalize well to circuits with a different number of qubits, as illustrated in figure 1. This limitation restricts its applicability to circuits of varying sizes.

Noise amplification: Q-LEAR introduces new noise by appending reversed segments to the original circuit, leading to an inaccurate estimation of the noise. This can degrade the quality of the noise mitigation.
Training resource requirements: Q-LEAR uses MLP that may require substantial computational resources for training, especially with large datasets. This can limit the scalability of the approach.

To overcome the limitations of existing methods, this work introduces a novel machine learning-based quantum noise mitigation method. Our method addresses the sensitivity to the number of qubits issue and avoids introducing new noise by leveraging the fidelity of the quantum backend and other relevant features of qubits number decoupling. Additionally, we employ a random forest model, which is computationally efficient and can handle large-scale circuits. The fifth chapter of this paper will comprehensively expound upon our approach, while the sixth chapter will empirically demonstrate the superiority of our methodology over existing ones.

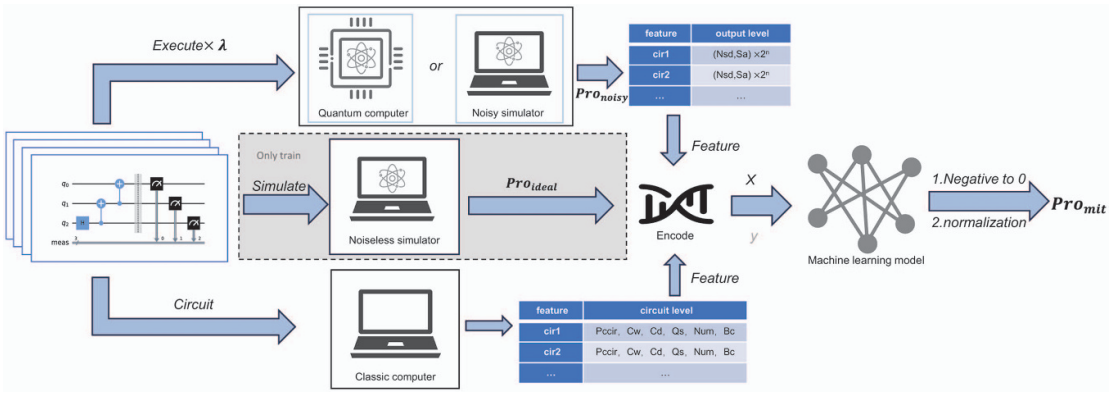

Figure 3. Overview of the fundamental process of machine learning-based quantum error mitigation. The process involves executing quantum circuits on noisy and ideal backends to obtain noisy outputs and ideal outputs, extracting circuit-level and output-level features (e.g. Pccir, Nsd), and using a classical random forest model to mitigate quantum errors.

Table 1. The selected algorithm from the MQT bench.

Quantum algorithm	Abbreviation
Amplitude estimation	AE [41]
GHZ State	GHZ [14]
Ground State	GS [42]
Quantum approximation optimization algorithm	QAOA [43]
Quantum phase estimation exact	QPEexact [44]
Quantum phase estimation inexact	QPEinexact [44]
Variational quantum eigensolver	VQE [45]
W-State	WS [46]

Source: www.cda.cit.tum.de/mqtbench/.

4. Approach

This paper presents a QEM method based on random forests model, which has the capability to reduce error in circuits whose number of qubits exceeds the number qubits of the training set. The novelty of this paper lies in the adoption of a method for estimating quantum noise that integrates quantum backend fidelity with quantum circuit gate and qubit information. This noise estimation does not disrupt the operation of the quantum circuit. We encode this noise estimation as one of the feature data for machine learning, denoted as Pccir. Beyond Pccir, the model also integrates additional novel features extracted at both the circuit and output levels. Feature dimensionality reduction is applied to decouple these features from the number of qubits in the quantum circuit, thereby overcoming the model's sensitivity to qubit count. We observed that the random forest model, within this characteristic dataset, demonstrated the most effective mitigation of quantum noise in the circuits whose number of qubits exceeds the number of qubits in the training set, among all the tested models. Figure 3 provides a schematic overview of the model's training and testing processes. As illustrated in the figure 3, the quantum computer is solely employed for executing quantum circuits during the acquisition of noisy result data, with the remainder including feature processing and model training being classical processes.

In order to assess and evaluate the effectiveness of the model's feature data and its applicability to quantum algorithms, we have selected a subset of quantum algorithms from MQT Bench [40] for subsequent experimental test, as detailed in table 1.

Next, we will proceed to delineate the methodology for acquiring circuit-level Feature and output-level Feature of this model.

4.1. Circuit-level Features

The model's feature data encompasses common statistical feature of quantum circuits, including circuit width (Cw), circuit depth (Cd), as well as the counts of *CX* gates, *X* gates, *SX* gates, and *RZ* gates.

The following introduces novel circuit-level features employed in this research.

Quantum gate parameters statistics (Qs): the parameters of parameterized quantum gates within the circuit, referred to as gate parameters, range from -2π to 2π. The interval $[-2\pi, 2\pi]$ is divided into eight sections at intervals of 0.5π. This setup is an empirically driven choice: an excessive number of sections may give rise to model overfitting, whereas an insufficient number of sections could lead to inadequate model resolution. The counts of gate parameters within each of these intervals are tallied and then input as feature data into the model. These feature data provide a sideways reflection of the circuit's structure, enabling the model to capture circuit feature and thus discern different circuits.

Quantum error evaluated by properties combined with circuit information (Pccir): for instance, in a three-qubit quantum circuit, the occurrences of cx gates on specific qubit pairs are tallied. As the error rates for the same gate on different qubits or qubit pairs vary, the backend property information is queried to retrieve the error rates for these gates on the specified qubit pairs. The summation of the error rates for all single-qubit and two-qubit gates, along with the summation of the readout error rates for qubits used, constitutes an estimate of the noise for the current circuit. Pccir is calculated as follows:

$$Pccir = \sum_{i,j} \mu_{ij} G_{ij}^{\text{error}} + \sum_{i} \mu_i G_i^{\text{error}} + \sum_{i} M_i^{\text{error}} \tag{3}$$

where μ denotes the number of quantum gates on the corresponding qubit or qubit pairs on the quantum circuit, G^{error} denotes the error rate of quantum gates on the corresponding qubit or qubit pairs on the quantum circuit, and M^{error} denotes the readout error on the corresponding qubit on the quantum circuit.

Quantum backend coding (Bc): the model utilizes six noisy simulated backends to generate training and test data. In order to enable the model to discern the backend used for the current circuit, six training backends are numbered from 1 to 6, and the backend code is incorporated as a circuit-level feature for the current circuit. However, when training the model using a real quantum computer, as only one real machine is available, the model does not need to distinguish the current device, and this feature data is excluded.

In the field of machine learning, an effective feature data must exhibit a strong correlation with the target values [47, 48]. To explore the potential association between this feature data and quantum noise, the algorithms in the MQT test set are partitioned based on the number of qubits. We delves into the Pearson correlation and Spearman correlation between circuit-level feature and circuit error, figure 4 illustrates the correlation between Pccir and circuit error, quantifying the linear relationship between Pccir and circuit error. The correlation measures the strength of the linear relationship between two random variables: a value of 1 indicates a strong positive fit, 0 denotes no linear relationship, and -1 signifies a strong negative fit. All the features in the figure exhibit a strong correlation with circuit errors, especially Pccir, which clearly exhibits a clear functional relationship. Therefore, these features can all be used as feature data for model training.

4.2. Output-level features

The output-level feature includes the probability values of the state distribution corresponding to the noisy output of the circuit, denoted as Pro_{noisy}, which is the primary mitigation objective of our approach. The following are the novel output-level features proposed by this model.

Noisy standard deviation (Nsd): Nsd is determined by running the circuit on a noisy quantum machine or simulator λ times and calculating the standard deviation of probability for a certain circuit output state. Can Nsd effectively assess the magnitude of state noise? We selected several circuits from the MQT benchmark test set, as shown in figure 5, and ran these circuits on the Qiskit noisy simulator, FakeGuadalupe, to obtain the noise values for each state and calculate the correlation between the state error and Nsd of the noisy results, as depicted in figure 5. It can be observed that, apart from the GS circuit, Nsd exhibits a high degree of correlation with State Error. This is because for a given state, the greater the error influence it is subject to, the greater the randomness in its probability value, which manifests as a larger standard deviation. In contrast, the GS algorithm circuit is relatively simple with a small circuit depth, leading to lower inherent errors and consequently showing a lower correlation. In

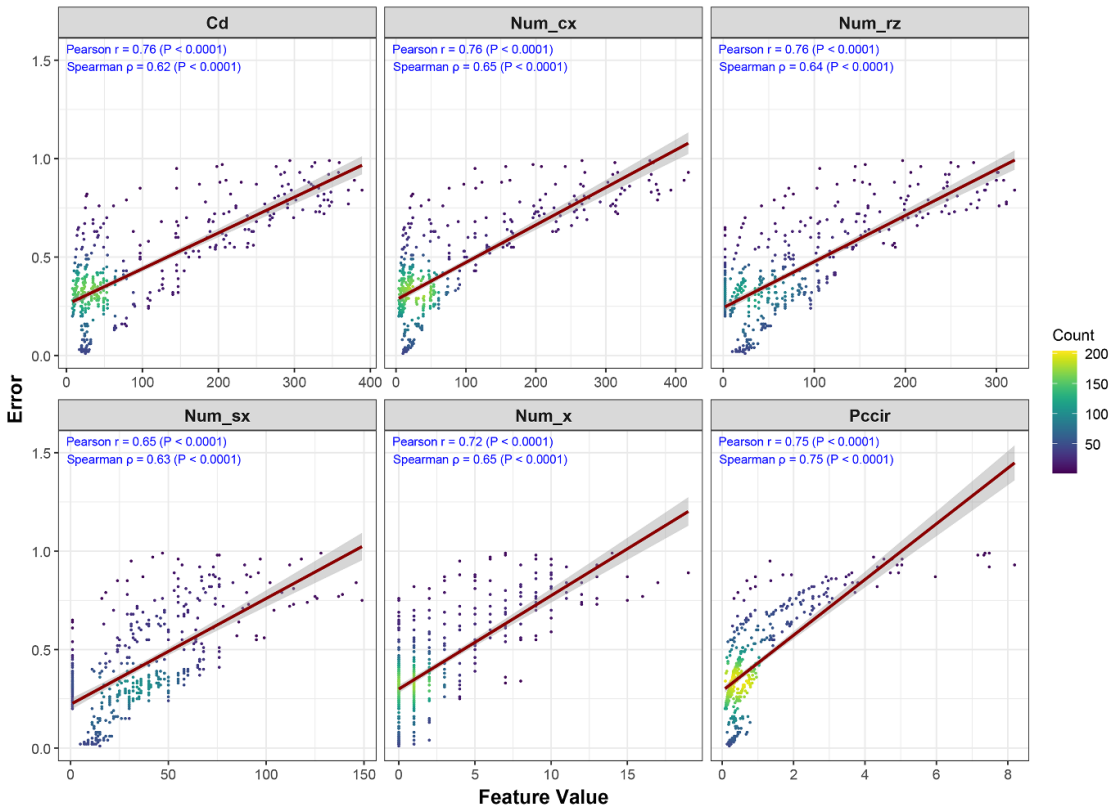

Figure 4. Pearson and Spearman correlation coefficients between circuit-level features and circuit error. Each subplot illustrates the correlation between a specific type of feature data and circuit error, where a monotonic increase indicates a positive correlation, and the gray shaded area represents the 95% confidence interval.

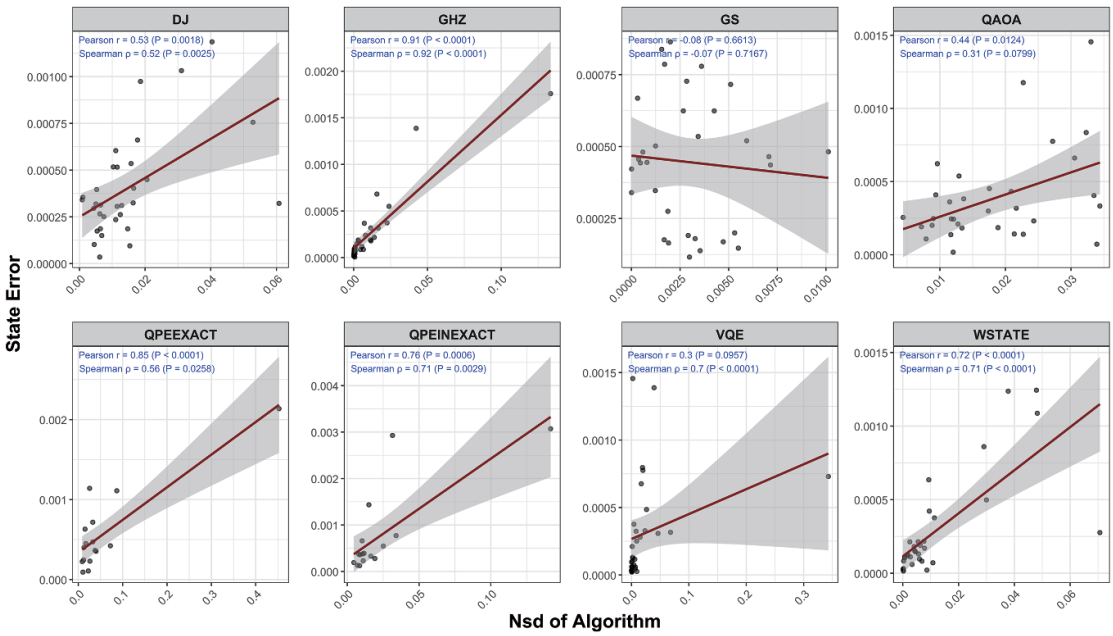

Figure 5. Pearson and Spearman correlation between the Nsd of the MQT bench test circuits and the state error. Each subplot illustrates the correlation between the noisy standard deviation (NSd) of results from a quantum algorithm and its corresponding state error, where a monotonic increase indicates a positive correlation, a monotonic decrease indicates a negative correlation, and the shaded area represents the 95% confidence interval.

Algorithm 1. QEMOS training workflow.

```
Input: Quantum circuit dataset
Output: QEMOS model
X ← [[]]                                                        //Feature data set
y ← [[]]                                                        //Target data set
for each QC in Circuit_Dataset do
    for k ← 1 to λ do
        Noisy_res ← ExecuteNoisy(QC)   // Executing on noisy backend or real quantum computer (iteration k)
    end
    Ideal_res ← ExecuteIdeal(QC)                                //Executing on ideal backend
    Circuit_Features ← ExtractFeature (QC)                      //Extracting circuit features
    for each Noisy_res_i in Noisy_res do
        Output_Feature_i ← ExtractFeatureNoisy_res              //Extracting output features
        X_i ← [Output_Feature_i,Circuit_Features,Noisy_res_i]
        y_i ← Ideal_resi
    end
    X ← X ∪ X_i
    y ← y ∪ y_i
end
Model ←TrainModelX,y
```

Algorithm 2. QEMOS inference workflow.

```
Input: Quantum circuits (QCs)
Output: Mitigated result
for each QC in QCs do                                   // mitigated for each circuit
    X ← GenerateFeature (QC);                           // same as the training Workflow
    Pre ← Predict(X);                                   // QEMOS model prediction
    Pre ← NegativeToZero(Pre);                          // Negative to zero
    Mitigated_res ← Normalization(Pre);
end
```

practical quantum algorithms, quantum circuits are typically more complex and have a greater depth. Therefore, Nsd can be used to evaluate the range of these fluctuations.

State assignment (Sa): Sa involves converting the binary numbers corresponding to the states of the circuit into their respective decimal numbers. For example, if a particular state corresponds to the binary string 01 101, it is converted to the decimal number 13. By combining Sa with circuit-level features such as Qnum, the 0–1 bit string for the current state can be inferred. We consider this feature data essential for the model because the noise in different states of different circuits varies. The model needs to differentiate between different circuits based on circuit-level features and also distinguish between different states based on the state assignment in the output-level features. This approach enables dimensionality reduction of qubit-dependent features into a fixed two-dimensional space, thereby achieving decoupling from the qubit number.

4.3. Model training and testing

The model was based on the random forest machine learning algorithm, which entailed the creation of a random forest comprising 300 decision trees. The trees were constructed in a top-down manner using a recursive and greedy algorithm. The splitting criterion for the decision trees employed mean squared error reduction for regression. At least 2 feature data points were required for the splitting of each decision tree. This configuration was chosen through extensive testing to balance model stability and computational cost. Using MSE helped the model identify nonlinear patterns in the data. The workflow for model training and Inference is detailed in algorithms 1 and 2.

5. Evaluation and analysis

The present study endeavors to address the efficacy and superiority of the QEMOS model in mitigating quantum noise by providing insights into the following four key questions:
Research question 1: what justifies the selection of the random forest model?
Research question 2: is the performance of QEMOS superior to Q-LEAR?
Research question 3: what factors contribute to the efficacy of QEMOS?
Research question 4: can QEMOS operate effectively on real quantum computers?

5.1. Experiment setting

5.1.1. Training data

The model's training data comprises randomly generated circuits from six noisy simulated backends of Qiskit. These simulated backends include FakeGuadalupe, FakeMontreal, FakeCairo, FakeMumbai, FakeSydney and FakeToronto. The noise of these backends is derived from real quantum computers. In situations where quantum computing resources are difficult to access, noisy simulated backends serve as a crucial means and approach for studying quantum computation. The number of qubits in the training circuits ranges from 5 to 9, with depth randomly determined between 1 and 100. Each backend has 80 circuits for each number of qubits, resulting in a total of 2400 training circuits, yielding an extensive training feature data set of up to 480 000 rows. For the real quantum computer used in this study, the number of qubits in training set still range from 5 to 9, with 80 randomly generated circuits for each.

5.1.2. Test data

The test set comprises randomly generated circuits from six noisy simulated backends of Qiskit, with the number of qubits in these circuits ranging from 2 to 13 and random circuit depths. Each backend has 20 circuits for each number of qubits. Additionally, the test set includes some MQT test sets compiled using these six backends, as detailed in table 1. The number of qubits in the circuit sets for each quantum algorithm in the test varies from 5 to 13. For the real quantum computer, the number of qubits in the test set ranges from 2 to 13 qubits, with 20 randomly generated circuits for each.

In the proposed experimental configuration, the quantum circuit is executed on noisy backend repeatedly λ times, with its value simply set to the minimum of 3 in this test. The repetition count λ can be incrementally adjusted to achieve enhanced noise characterization fidelity, though this improvement comes at the expense of increased computational resources.

For strong error mitigation, compared with weak error mitigation, its resource consumption grows exponentially, and it requires higher precision of methods. As a study on a strong error mitigation method, this research has achieved significant breakthroughs in scalability compared with the state-of-the-art methods. However, limited by the difficulties of strong error mitigation tasks themselves and the limitations of computing resources, in this study, the scale of the quantum circuit was only set to 2–13 qubits during testing. Nevertheless, we still provided test results for four 25-qubit circuits as a reference to verify the scalability of the model, and all these circuits yield high-probability state outputs without exception.

5.2. Results and analyses

5.2.1. RQ1: What justifies the selection of the random forest model?

For RQ1, we employed a comparative experimental approach to assess the performance of five machine learning models on the feature data in this study. These models include random forest, linear regression (Linear), ridge regression (Ridge), support vector regression (SVR), and K-nearest neighbor regression (KNN). The variable in the experiment is the model used, while the feature data remains constant. The lower the noise after a machine learning model mitigation, the better the model is in the feature set. Specifically, after training each model using the same dataset, we tested the models using a uniform test dataset. The test results are grouped by the number of qubits of the circuits. Specifically, we group together the circuit error obtained by testing circuits with the same number of qubits using the corresponding model, and calculate their average. This is to identify models capable of mitigating circuits whose number of qubits exceeds that of the training set. The evaluation of circuit error utilized the Hellinger distance as mentioned earlier, and the test results are depicted in figure 6.

It is evident that some of the models we employed lack noise mitigation capabilities, such as SVR and KNN. While certain models demonstrate the ability to mitigate noise, they can only do so within the number of qubits range of the training set, as seen in the cases of Linear and Ridge. Among the tested models, only the random forest model exhibits noise mitigation capabilities both within and beyond the range of the number of qubits in the training set. This distinction arises from the nature of our

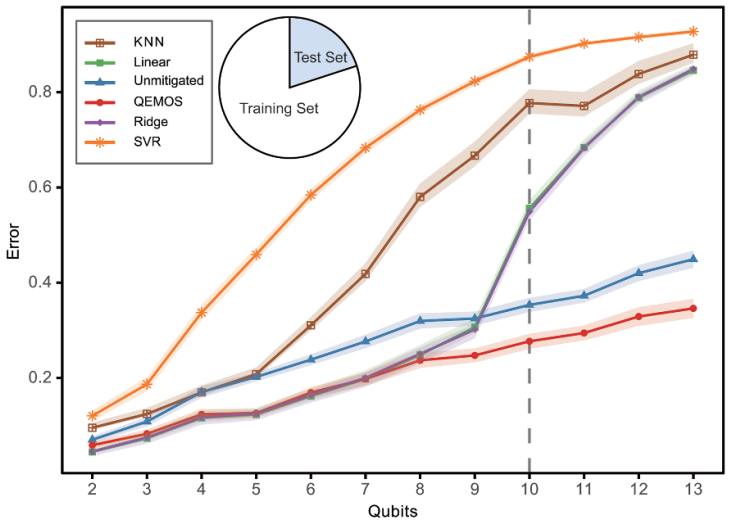

Figure 6. Comparison of the performance of different machine learning models on the feature data in this study. The pie chart illustrates the proportional distribution of the model's training and test sets, while the dashed line indicates that the number of qubits in the test set circuits begins to exceed that of the training set starting from 10 qubits.

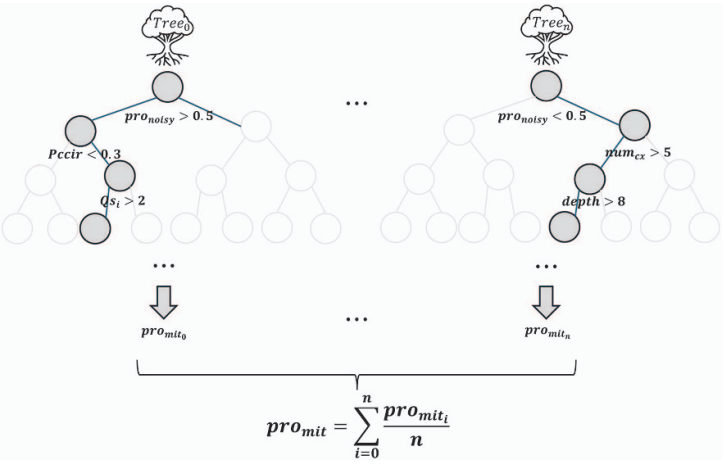

Figure 7. The random forest model mitigate quantum error. The random forest model operates by splitting nodes of decision trees based on feature data, and then averaging the results of all decision trees to produce the final output.

task objective. QEM tasks are, in essence, neither classification tasks nor regression tasks. This implies that the model's output is not a fixed value and does not involve straightforward regression prediction tasks. Rather, it involves introducing a certain bias to the noisy circuit output based on the feature data, steering the solutions towards theoretical values. Given that quantum noise accounts for a relatively small proportion of the circuit output probabilities, this indicates that the bias we need to introduce is minimal. Moreover, there is no clear functional relationship between this bias and the feature data, leading to the limitation of certain models to perform well only within the qubit range of the training set. It is worth noting that even when experiments are conducted on larger-scale datasets, the performance trend of the models involved in the testing remains consistent with what is shown in figure 6, and only QEMOS has overcome the sensitivity to the number of qubits. This observation indicates that the data scale used for training our model is sufficient to support the demonstration of QEMOS's superiority.

In our model, there exists a strong correlation between the feature data and quantum noise. This suggests a certain association between our feature data and quantum noise. Decision tree node splitting in the random forest based on strongly correlated feature data effectively represents this type of noise without resorting to a functional fit [49, 50]. Furthermore, by averaging the results of all decision trees, the random forest model adeptly represents the slight perturbations of noise on the noisy circuit results and mitigates them. Consequently, the random forest model is capable of mitigating quantum noise

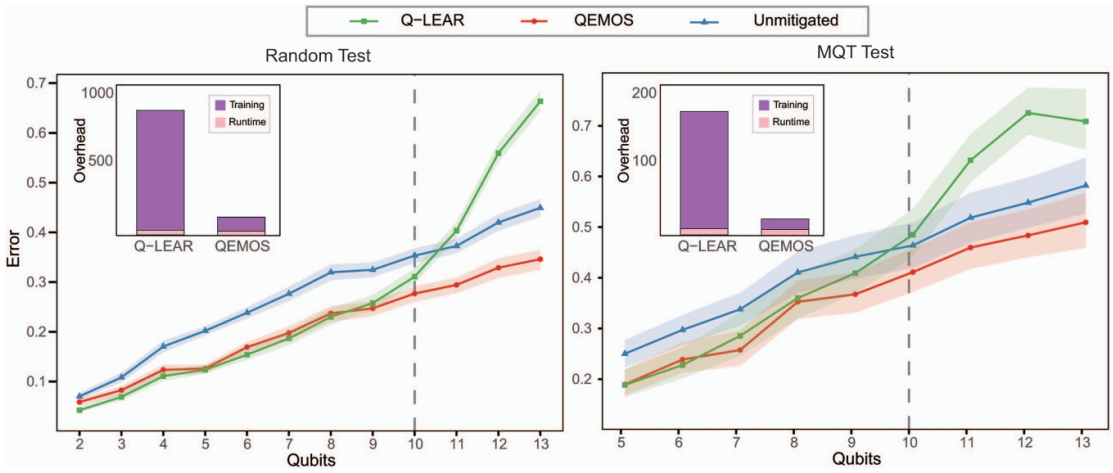

Figure 8. Comparison of QEMOS and Q-LEAR on the random circuit test set and MQT test set. The bar chart illustrates the classical resource consumption for training and testing of different models.

from the circuits whose number of qubits beyond the range of the training set. Figure 7 illustrates the technical principles of the random forest algorithm.

5.2.2. RQ2: Is the performance of QEMOS superior to Q-LEAR?

We conducted separate tests for QEMOS and Q-LEAR using both the random circuit test set and MQT test set. As shown in figure 8. During the analysis, we calculated the average circuit error for each number of qubits. Detailed data are shown in table 2.

For the random circuit test set, when the test circuits' number of qubits falls within the range of the training set's number of qubits, the Q-LEAR model exhibits noise mitigation. The QEMOS model, while slightly inferior, remains nearly on par with Q-LEAR. This is due to the random circuit's depth, which ranges from 1 to 10, resulting in a relatively diminutive level of quantum noise. However, as the test qubit count increases, the QEMOS model begins to outperform Q-LEAR when the test circuits' number of qubits reaches the boundary of the training set at 9 qubits. By 10 qubits, QEMOS notably excels, and beyond 11 qubits, Q-LEAR no longer demonstrates noise mitigation capability, instead introducing new noise to the circuit results. In contrast, the QEMOS model continues to effectively mitigate noise, demonstrating its insensitivity to number of qubits compared to the Q-LEAR model. Outside the number of qubits range of the training set, QEMOS outperforms the Q-LEAR model by an average of 31.74%.

As for the MQT test set, where circuits are more complex and noise levels are higher, demanding greater noise mitigation capabilities from the model, the test results are similar to those of the random forest. Outside the range of qubits in the training set, QEMOS outperforms Q-LEAR by an impressive 23.98% on average. Moreover, within the test set's number of qubits range, QEMOS slightly outperforms Q-LEAR, indicating that QEMOS excels on more complex quantum circuits. The feature data of QEMOS proves to be more suitable for practical quantum algorithm circuits of greater complexity.

Additionally, during model training, the Q-LEAR model required 800 min for training on 480 000 rows of training data, whereas QEMOS, in contrast, only took 60 min. This indicates that the QEMOS model is more suitable for training larger-scale circuit data, ensuring superior scalability compared to Q-LEAR.

5.2.3. RQ3: What factors contribute to the efficacy of QEMOS?

As our work revolves around mitigating noise in circuit outputs, the probabilities corresponding to the noisy output states should play the most pivotal role in the feature data. Here, we exclusively focus on the significance of feature data, excluding Pro_{noisy} within the model. In a random forest model, the importance of feature data is evaluated based on the improvement in the splitting criterion for each node of the decision trees. We can employ the built-in function in the sklearn library to directly yield the importance of feature data, and figure 9(a) illustrates the significance of these features for the model.

It is evident from the assessment of the feature data that the importance of Pccir is the highest. This aligns with the findings of previous experiments and analyses, indicating the effectiveness of this feature in evaluating quantum noise and its critical role in learning and mitigating noise within the model. Additionally, the significance of Sa is relatively high. This is because within the same circuit, different

Table 2. Comparison of QEMOS and Q-LEAR on the random circuit test set and MQT Test Set. The column for each model represents the circuit error after processing by that model. The columns %S and %Q illustrate the reduction of quantum noise by the QEMOS and Q-LEAR models(given by $(m_1 - m_2)/m_1$). The promote column corresponds to the improvement of QEMOS over Q-LEAR in terms of effectiveness.

Data Set	Qubits	QEMOS	Q-LEAR	Unmitigated	%S	%Q	Promote
Random	2–9(Average)	0.155	0.147	0.214	0.276	0.313	−0.054
	10	0.277	0.311	0.353	0.216	0.12	0.109
	11	0.294	0.403	0.373	0.21	−0.083	0.27
	12	0.329	0.559	0.42	0.217	−0.331	0.412
	13	0.346	0.663	0.449	0.23	−0.477	0.478
MQT	5–9(Average)	0.281	0.297	0.347	0.19	0.144	0.054
	10	0.411	0.468	0.464	0.114	−0.01	0.122
	11	0.46	0.607	0.519	0.114	−0.171	0.243
	12	0.484	0.704	0.548	0.118	−0.284	0.313
	13	0.509	0.708	0.582	0.125	−0.217	0.281

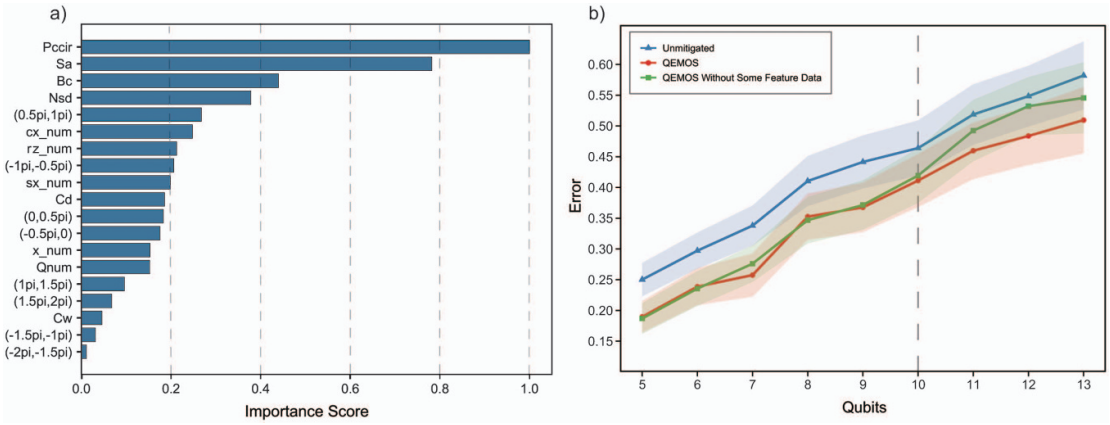

Figure 9. (a) Feature importance in the QEMOS model. (b) Ablation study for QEMOS.

output states correspond to different noise, and the model needs to distinguish and learn the differences between these states. The quantum backend encoding Bc is also a key feature, indicating that different quantum backends exhibit distinct noise features, necessitating the model to differentiate and learn the noise properties of different backends. Furthermore, the inclusion of the Nsd ranks fourth in importance within the feature data, aligning with the results of the prior correlation analysis, indicating its effectiveness in estimating state errors. Other feature data, such as Sa, exhibit relatively dispersed importance in the feature data, consistent with our expectations as these feature data are designed to assist the model in learning circuit features. Additionally, the importance of the number of *CX* gates within the feature data falls within the upper-midrange. This is due to the fact that the noise from two-qubit gates involves interactions between two qubits, potentially leading to amplified noise levels.

To ascertain whether our feature data contributes to noise mitigation within the model, we excluded Pccir, Sa, Bc, Nsd, and Qs from the feature dataset. Subsequently, we retrained the model on circuits ranging from 5 to 9 qubits and tested it on the MQT test set. As illustrated in figure 9(b), the model without these features still demonstrates effective noise mitigation within the range of qubit number present in the training set. However, as the number of qubits of the test circuits exceeds the range of the training set, the model's performance begins to deteriorate. In contrast, the model trained with the complete feature dataset continues to perform well. This indicates that our feature data aids in mitigating the model's sensitivity to the number of qubits, enabling the model to effectively mitigate noise in circuits with a higher number of qubits.

5.2.4. RQ4: Can QEMOS operate effectively on real quantum computers?

The aforementioned experiments were conducted in six simulators. However, our ultimate objective is for the model to be applicable to real quantum computers. In order to validate the model's effectiveness on a real quantum computer, tianyan-176 was utilized to test this model. This choice was made

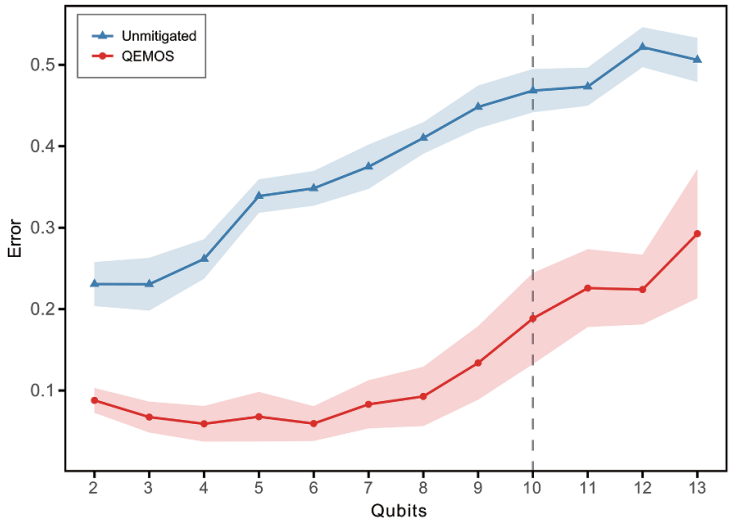

Figure 10. The test result of QEMOS model on real quantum computer.

Table 3. Test QEMOS on tianyan-176 with 25-qubit circuits.

Circuit	Unmitigated	Mited	Promote
Cir1	0.658	0.359	0.455
Cir2	0.605	0.115	0.81
Cir3	0.584	0.391	0.331
Cir4	0.678	0.532	0.215

because the tianyan-176 provides access to the backend properties and topology required for model training, and this information is continually updated in real time, meeting the training requirements of the model. Similarly, to validate the model's advantage of being insensitive to the number of qubits, we partitioned the results based on the number of qubits and calculated the average circuit error, as shown in the figure 10.

The test results on the real quantum processor demonstrate that our model performs exceptionally well, both within and beyond the range of number of qubits in the training set, with an average noise mitigation efficacy of 67.5%. This indicates that our model is not only applicable to noise simulators, but also to actual quantum processors, showcasing its practical value.

Furthermore, to further demonstrate the model's capability to overcome sensitivity to the number of qubits, and considering current classical and quantum resource constraints, we executed four 25-qubit circuits on a real quantum computer and tested these circuits using the model. The test results are presented in the table 3. Observing the test result of the four circuits, it is evident that the model exhibits mitigation effects. This indicates that our model can operate on a larger qubit scale, showing that it has overcome the sensitivity to the number of qubits and can be extended for practical application to quantum circuits whose number of qubits exceeds the number of qubits in the training set.

However, the error mitigation model still has limitations. Specifically, when conducting noise mitigation on a 25-qubit scale, we observed that if the probability distribution of the circuit output is relatively uniform, such that the probability values for all states are small and lack clear high-probability output, the model's mitigation efficacy is inadequate, or even absent. This is due to the excessively small probability values, exceeding the precision of the model's mitigation capabilities. In fact, for larger-scale qubit circuits, our model demonstrates effective error mitigation only when applied to circuits exhibiting distinctly high-probability solutions, while its mitigation efficacy is compromised for the opposite scenario. This presents a future research direction for machine learning-based QEM.

Furthermore, in large-scale quantum circuits, if the circuit depth is particularly substantial and the number of quantum gates within the circuit is exceedingly high, resulting in substantial noise within the circuit and a significant deviation or lack of correlation between the noisy circuit output and the theoretical values, the noise mitigation model also becomes ineffective. This is because the noise in the circuit has surpassed the capability range of the noise mitigation model. In such cases, a potential approach

involves circuit cutting [51] and compiling optimizations to alleviate noise before conducting noise mitigation on the circuit, or utilizing more complex quantum error-correcting codes to eliminate noise, although this is currently challenging with existing hardware.

6. Discussion

The QEM method based on machine learning is actually a QEM method by collecting the feature data of the quantum computer, the feature data of the quantum circuit itself and the noisy results obtained from the output of the circuit, combined with machine learning, that is, using the classical computer for post-processing, the advantage of this method is that it can overcome the problem of resource consumption and technical complexity of the previous method, and at the same time, it has a good effect, therefore, the core problem of this method is how to evaluate the possible error of the quantum circuit before the quantum computer runs, and the resources consumed by this method should be acceptable, then, the evaluation of this error is used as one of the feature data of machine learning for QEM. We have a novel idea of using machine learning to evaluate quantum noise before a quantum circuit is run, and then using this noise data as feature data for subsequent ML, so far, the former has been relevant research [52], the latter is the scope of this study.

The most important among all the feature data adopted in this study is Pccir, which is collected from the measurement and control information of the backend of real quantum devices. During the experiment, we noticed that when the environment of the real device changes, the historical backend properties may not be applicable to the current status of the real device. Moreover, inaccurate backend properties will also affect the QEM performance of the model. In such cases, timely updating the measurement and control information and taking the average may be an effective solution. In view of this, our next research is attempting to develop a set of rapid noise test sets to quickly obtain the noise state of real quantum devices, so as to maintain the timeliness and accuracy of backend properties.

7. Conclusion and future work

This paper introduce a machine learning-based QEM method called QEMOS. This method evaluates quantum noise without disrupting the operation of quantum circuits or increasing quantum computing resources. It incorporates the evaluation results as one of the feature data for machine learning, alongside other data derived from quantum circuits and quantum computers. By combining these data, the method leverages a random forest model to realize QEM. Through assessments on six different simulated backends, the model demonstrates the capability to overcome the limitations associated with the sensitivity to the number of qubits, achieving an average improvement of 31.74% beyond previous methods for the circuits whose number of qubits exceeds the number of qubits in the training set. Furthermore, on a real quantum computer, training and testing were conducted using tianyan-176. Across all test sets, errors were reduced by an average of 67.5% compared to the unmitigated circuit output.

As previously described, this method has yet to explore the boundaries of QEM. For instance, when the quantum circuit involves a large number of qubits and exhibits considerable scale, the distribution of circuit output is so dispersed that the probabilities corresponding to all states approach zero, resulting in the model's poor ability to mitigate errors. Moreover, even when quantum circuits possess clearly high-probability output, if the circuits are overly complex and the errors are too substantial, exceeding the model's mitigation capacity, the model also faces limitations in error mitigation. Hence, the next focus should center on seeking a more advanced approach to enable the model to overcome these challenges.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://figshare.com/s/4c62893df7a76fc92f60.

Acknowledgment

This work is supported by the National Key RD Program of China (No. 2024YFB4504100). The experimental component of this research is supported by the Tianyan Quantum Computing Cloud Platform.

ORCID iDs

Zheng Tu 0009-0005-2817-5229
Jinchen Xu 0000-0002-6275-2617
Xin Zhou 0000-0003-0231-2895
Yu Zhu 0009-0005-1660-310X
Yi Liu 0009-0006-7118-5753
Qiming Du 0000-0002-6823-2321
Hang Lian 0009-0000-0008-6287
Bei Zhou 0000-0003-1515-0602

References

[1] Bogdanov Y I, Chernyavskiy A Y, Holevo A, Lukichev V F and Orlikovsky A A 2013 Modeling of quantum noise and the quality of hardware components of quantum computers *Int. Conf. Micro-and Nano-Electronics 2012* vol 8700 (SPIE) pp 404–15
[2] Resch S and Karpuzcu U R 2021 Benchmarking quantum computers and the impact of quantum noise *ACM Comput. Surv.* **54** 142
[3] Preskill J 2018 Quantum Computing in the NISQ era and beyond *Quantum* **2** 79
[4] Ponnath A 2006 Difficulties in the implementation of quantum computers (arXiv:cs/0602096)
[5] Donkor E and Kumavor P D 2002 Prospects and challenges for the realization of quantum computers *Proc. SPIE* **4732** 153–9
[6] Saini R, Sewada R, Arora A and Kumar B G S 2022 Current challenge and limitations in quantum computation *J. Nonlinear Anal. Optimization* **13** 26–35
[7] Du Y, Hsieh M-H, Liu T, Tao D and Liu N 2021 Quantum noise protects quantum classifiers against adversaries *Phys. Rev. Res.* **3** 023153
[8] Li Y and Benjamin S C 2017 Efficient variational quantum simulator incorporating active error minimization *Phys. Rev. X* **7** 021050
[9] Temme K, Bravyi S and Gambetta J M 2017 Error mitigation for short-depth quantum circuits *Phys. Rev. Lett.* **119** 180509
[10] Barron G S and Wood C J 2020 Measurement error mitigation for variational quantum algorithms (arXiv:2010.08520)
[11] Cai Z, Babbush R, Benjamin S C, Endo S, Huggins W J, Li Y, McClean J R and O'Brien T E 2023 Quantum error mitigation *Rev. Mod. Phys.* **95** 045005
[12] Muqeet A, Ali S, Yue T and Arcaini P 2024 A machine learning-based error mitigation approach for reliable software development on ibm's quantum computers *Companion Proc. 32nd ACM Int. Conf. on the Foundations of Software Engineering (FSE 2024)* (Association for Computing Machinery) pp 80–91
[13] Rines R, Obenland K, and Chuang I 2019 Empirical determination of the simulation capacity of a near-term quantum computer (arXiv:1905.10724)
[14] Pan J-wei and Zeilinger A 1998 Greenberger-Horne-Zeilinger-state analyzer *Phys. Rev. A* **57** 2208–11
[15] Maronese M, Moro L, Rocutto L and Prati E 2022 *Quantum Compiling* (Springer) pp 39–74
[16] Zhang C-Y, Zheng Z-J, Fei S-M and Feng M 2023 Dynamics of quantum networks in noisy environments *Entropy* **25**
[17] Tian L, Lloyd S and Orlando T P 2001 Environmental noise on a qubit through entanglement with a quantum circuit *Optical Fiber Communication Conf. and Int. Conf. on Quantum Information* (Optica Publishing Group) p B23
[18] Dobrovitski V V, de Lange G, Ristè D and Hanson R 2010 Bootstrap tomography of the pulses for quantum control *Phys. Rev. Lett.* **105** 077601
[19] Shor P W 1996 Fault-tolerant quantum computation *Proc. 37th Conf. on Foundations of Computer Science* pp 56–65
[20] Bravyi S, Cross A W, Gambetta J M, Maslov D, Rall P and Yoder T J 2024 High-threshold and low-overhead fault-tolerant quantum memory *Nature* **627** 778–82
[21] Wu J and Karunamuni R J 2009 On minimum Hellinger distance estimation *Canadian J. Stat.* **37** 514–33
[22] Quek Y, França D S, Khatri S, Meyer J J and Eisert J 2024 Exponentially tighter bounds on limitations of quantum error mitigation *Nat. Phys.* **20** 1648–58
[23] Patil H P, Li P, Liu J and Zhou H 2023 Folding-free zne: a comprehensive quantum zero-noise extrapolation approach for mitigating depolarizing and decoherence noise *2023 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)* vol 01 pp 898–909
[24] Cai Z 2021 Multi-exponential error extrapolation and combining error mitigation techniques for NISQ applications *npj Quantum Inf.* **7** 80
[25] Giurgica-Tiron T, Hindy Y, LaRose R, Mari A and Zeng W J 2020 Digital zero noise extrapolation for quantum error mitigation *2020 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)* pp 306–16
[26] Halder S, Shrikhande C and Maitra R 2023 Development of zero-noise extrapolated projective quantum algorithm for accurate evaluation of molecular energetics in noisy quantum devices *J. Chem. Phys.* **159** 114115
[27] Schultz K, LaRose R, Mari A, Quiroz G, Shammah N, David Clader B and Zeng W J 2022 Impact of time-correlated noise on zero-noise extrapolation *Phys. Rev. A* **106** 052406
[28] Wallman J J and Emerson J 2016 Noise tailoring for scalable quantum computation via randomized compiling *Phys. Rev. A* **94** 052325
[29] Berg E V D, Minev Z K, Kandala A and Temme K 2023 Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors *Nat. Phys.* **19** 1116–21
[30] McDonough B, Mari A, Shammah N, Stemen N T, Wahl M, Zeng W J and Orth P P 2022 Automated quantum error mitigation based on probabilistic error reduction *2022 IEEE/ACM 3rd Int. Workshop on Quantum Computing Software (QCS)* pp 83–93
[31] Strikis A, Qin D, Chen Y, Benjamin S C and Li Y 2021 Learning-based quantum error mitigation *PRX Quantum* **2** 040330
[32] Liao H, Wang D S, Sitdikov I, Salcedo C, Seif A and Minev Z K 2024 Machine learning for practical quantum error mitigation *Nat. Mach. Intell.* **6** 1478–86
[33] Nagata K, Resconi G, Nakamura T, Batle J, Abdalla S and Farouk A 2017 A generalization of the Bernstein-Vazirani algorithm *MOJ Ecol. Environ. Sci.* **2** 00010

[34] Gulde S, Riebe M, Lancaster G P T, Becher C, Eschner J, Häffner H, Schmidt-Kaler F, Chuang I L and Blatt R 2003 Implementation of the Deutsch–Jozsa algorithm on an ion-trap quantum computer *Nature* **421** 48–50
[35] Jozsa R 1999 Searching in Grover's algorithm (arXiv:quant-ph/9901021)
[36] Cao Y, Romero J and Aspuru-Guzik A 2018 Potential of quantum computing for drug discovery *IBM J. Res. Dev.* **62** 6–1
[37] Raković D *et al* 2010 On some quantum approaches to biomolecular recognition *Contemp. Mater.* **1** 80–86
[38] Ekert A and Jozsa R 1996 Quantum computation and Shor's factoring algorithm *Rev. Mod. Phys.* **68** 733
[39] Shor P W 1999 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM Rev.* **41** 303–32
[40] Quetschlich N, Burgholzer L and Wille R 2023 MQT Bench: benchmarking software and design automation tools for quantum computing *Quantum* **7** 1062
[41] Brassard G, Hoyer P, Mosca M and Tapp A 2000 Quantum amplitude amplification and estimation (arXiv:quant-ph/0005055)
[42] Cerezo M *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 625–44
[43] Zhou L, Wang S-T, Choi S, Pichler H and Lukin M D 2020 Quantum approximate optimization algorithm: performance, mechanism and implementation on near-term devices *Phys. Rev. X* **10** 021067
[44] Li J 2024 Iterative method to improve the precision of the quantum-phase-estimation algorithm *Phys. Rev. A* **109** 032606
[45] Tilly J *et al* 2022 The variational quantum eigensolver: a review of methods and best practices *Phys. Rep.* **986** 1–128
[46] Swain M, Devrari V, Rai A, Behera B K and Panigrahi P K 2020 Generation of perfect w-state and demonstration of its application to quantum information splitting (arXiv:2006.01742)
[47] Theng D and Bhoyar K K 2024 Feature selection techniques for machine learning: a survey of more than two decades of research *Knowledge Inf. Syst.* **66** 1575–637
[48] Gopika N and Kowshalaya A M M E 2018 Correlation based feature selection algorithm for machine learning *2018 3rd Int. Conf. on Communication and Electronics Systems (ICCES)* pp 692–5
[49] Breiman L 2001 Random forests *Mach. Learn.* **45** 5–32
[50] Biau G 2012 Analysis of a random forests model *J. Mach. Learn. Res.* **13** 1063–95
[51] Tang W, Tomesh T, Suchara M, Larson J and Martonosi M 2021 CutQC: using small Quantum computers for large Quantum circuit evaluations *26th ACM Int. Conf. on Architectural Support for Programming Languages and Operating Systems* pp 473–86
[52] Pan Z, Feng Y, Li Z, Liu Y and Li Y 2023 Understanding the impact of quantum noise on quantum programs *2023 IEEE Int. Conf. on Software Analysis, Evolution and Reengineering (SANER)* pp 426–37