Quantum Information Processing (2024) 23:362
https://doi.org/10.1007/s11128-024-04566-9

A novel image classification framework based on variational quantum algorithms

Yixiong Chen[1]

Received: 5 March 2024 / Accepted: 2 October 2024 / Published online: 23 October 2024

Abstract

Image classification is a crucial task in machine learning with widespread practical applications. The existing classical framework for image classification typically utilizes a global pooling operation at the end of the network to reduce computational complexity and mitigate overfitting. However, this operation often results in a significant loss of information, which can affect the performance of classification models. To overcome this limitation, we introduce a novel image classification framework that leverages variational quantum algorithms (VQAs) hybrid approaches combining quantum and classical computing paradigms within quantum machine learning. The major advantage of our framework is the elimination of the need for the global pooling operation at the end of the network. In this way, our approach preserves more discriminative features and fine-grained details in the images, which enhances classification performance. Additionally, employing VQAs enables our framework to have fewer parameters than the classical framework, even in the absence of global pooling, which makes it more advantageous in preventing overfitting. We apply our method to different state-of-the-art image classification models and demonstrate the superiority of the proposed quantum architecture over its classical counterpart through a series of state vector simulation experiments on public datasets. Our experiments show that the proposed quantum framework achieves up to a 9.21% increase in accuracy and up to a 15.79% improvement in F1 score, compared to the classical framework. Additionally, we explore the impact of shot noise on our method through shot-based simulation and find that increasing the number of measurements does not always lead to better results. Selecting an appropriate number of measurements can yield optimal results, even surpassing those obtained from state vector simulation.

Keywords Image classification · Global pooling · Variational quantum algorithm · Quantum machine learning · Quantum computing

✉ Yixiong Chen
chenyixiong516@msn.com

[1] Beijing Science and Technology Manager Management Corporation, Beijing, China

1 Introduction

In light of recent breakthroughs in quantum technologies, particularly the availability of noisy intermediate-scale quantum (NISQ) processors [1], the field of quantum machine learning has attracted growing concerns and triggered an enormous amount of work [2–6]. Quantum machine learning is an interdisciplinary field that combines concepts from quantum physics and machine learning to develop innovative algorithms and models. By harnessing the power of quantum computing, quantum machine learning aims to improve the performance of machine learning algorithms. While still an emerging discipline, quantum machine learning has already demonstrated promising quantum extensions to classical machine learning techniques including support vector machines [7], clustering [8, 9], and principal component analysis [10].

Image classification is a fundamental task in computer vision that involves categorizing images into predefined classes or categories. Over the past few years, this field has witnessed rapid progress. Convolutional neural networks (CNNs) emerged as a breakthrough technique, significantly improving image classification accuracy. Various CNN architectures [11–17] have progressively advanced state-of-the-art (SOTA) results on benchmark datasets such as ImageNet [18]. Meanwhile, self-attention models like Transformers [19] in natural language processing have been introduced to computer vision. The corresponding transformer-based vision models, such as Vision Transformer (ViT) [20] and Swin Transformer [21], can achieve compelling results on image recognition and even outperform CNNs on some tasks. More recently, hybrid approaches combining CNN and transformer modules are also gaining attention. These hybrid CNN–transformer models, including CoAtNet [22] and MaxViT [23], have been demonstrated to improve the generalizability of pure transformer-based models and achieve excellent performance in image classification without being pre-trained on large-scale image datasets.

In most image classification architectures, the input images are processed by a backbone model, also referred to as a feature extractor, which extracts relevant features from the input. This backbone model generally consists of a series of convolutional layers, transformer-based layers, or a combination of both. The resulting feature maps are typically fed into a global pooling layer followed by fully connected layers and classifier. Global pooling is a commonly employed technique in the field of image recognition [12–17, 21–23]. It aggregates the spatial information of feature maps across the entire image into a single vector representation. The advantage of global pooling lies in its ability to capture the overall context and semantic information of an image, resulting in a compact and fixed-length representation that can be easily fed into subsequent layers for classification tasks. Global pooling also helps to mitigate the spatial variance of features, enabling the model to achieve translation invariance and robustness to object deformations. Moreover, it reduces the dimensionality of the feature maps, leading to computational efficiency and alleviating the risk of overfitting.

While global pooling provides some benefits for image classification, it has several limitations. Collapsing an entire feature map into a single vector results in the complete loss of fine-grained spatial information. The compact fixed-length representation has also restricted expressive capacity compared to a fully connected layer operating on spatial data, which may hinder capturing complex spatial relationships. This can

be disadvantageous for tasks that require precise object localization, discriminative information, or detailed spatial information. Furthermore, uninformative features corresponding to background or irrelevant regions can dominate the global summary statistic and render it unresponsive to salient features. Additionally, since global pooling treats the entire image equally, it may not effectively capture local variations or small-scale patterns that are important for accurate recognition.

Researchers have explored different strategies to enhance the performance of global pooling and mitigate its drawbacks. Firstly, several variants of global pooling are proposed to provide a trade-off between the global average pooling and global max pooling with a nonlinear log-average-exp (LAE) [24, 25] or log-sum-exp (LSE) [26] function, which helps extract features more effectively. Another approach is to leverage pyramid pooling [27–30] as an alternative to global pooling. Pyramid pooling methods divide the image into multiple regions of different scales and extract features from each region independently. This allows for the capture of multi-scale information and the preservation of spatial details. Moreover, attention mechanisms have also been integrated with global pooling to improve its discriminative power [31–33]. These mechanisms selectively emphasize salient regions or features while suppressing irrelevant ones, enhancing the model's ability to capture important information. In addition, spatial transformer networks (STNs) [34] have been proposed to enable adaptive spatial transformations prior to pooling. STNs apply learned transformations to align features, enhancing the localization accuracy of global pooling. Furthermore, the method of global second-order pooling [35–38] is also exploited for capturing more discriminative image representations. This approach calculates covariance matrices to obtain higher-order statistical information between features.

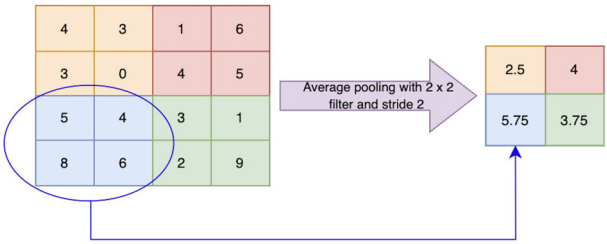
In contrast with the above methods, our work employs variational quantum algorithms (VQAs) [39–41] to address the issue of global pooling for the task of image classification. VQAs are a class of quantum algorithms that utilize classical computers to optimize a parameterized quantum circuit. Due to their hybrid quantum–classical approach that has potential noise resilience [39], VQAs have been considered as one of the leading candidates to achieve application-oriented quantum computational advantage on NISQ devices. Quantum machine learning models based on VQAs, particularly quantum neural networks, have been extensively studied and applied in the field of image classification [42–54]. A notable example is the hybrid quantum–classical convolutional neural network (QCCNN) proposed in [47], which outperformed classical CNNs on a Tetris dataset. Similarly, the introduction of quanvolutional layers in [45], which transform input data using nonparametric random quantum circuits, resulted in faster training and higher test set accuracy on the MNIST [55] dataset compared to purely classical CNNs. Another significant contribution is found in [46], where a quantum deep convolutional neural network (QDCNN) demonstrated exponential acceleration over classical models on the MNIST and GTSRB [56] datasets. Moreover, fully parameterized QCNNs were benchmarked in [51] for image classification, achieving excellent accuracy with fewer parameters and surpassing CNNs under similar training conditions on the MNIST and Fashion MNIST [57] datasets. Lastly, the study in [54] explored various pooling techniques in hybrid quantum–classical CNNs and showed comparable or superior performance to classical models for 2D medical image classification.

However, there is currently no work in the field of quantum machine learning that addresses the problem of information loss caused by global pooling in image classification models. Most of previous work focuses on constructing quantum versions of pooling operations using VQAs. Pooling and global pooling are two distinct operations. Typically, pooling layers, which operate on local regions of the image to reduce the spatial dimensions and retain important features, are used after convolutional layers, whereas global pooling, which operates on the entire image to aggregate features, is applied before classification. To the best of our knowledge, our work is the first in quantum machine learning to utilize VQAs to solve the global pooling problem in image classification models.

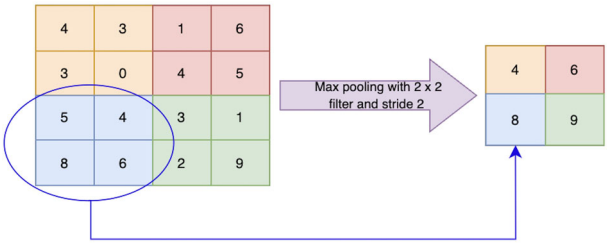
More specifically, in this paper, we propose a novel image classification framework based on VQAs. This framework differs from classical architectures primarily in that global pooling operations at the end of the network are no longer required. This is due to the introduction of the variational quantum circuits (VQCs) [41, 58, 59] which possess remarkable data storage capacity. Consequently, our framework can fully utilize the feature maps extracted via the backbone model for image classification, which is particularly advantageous for tasks requiring fine-grained features. Moreover, despite the removal of the global pooling layer, our framework does not suffer from the issue of excessive model parameters and overfitting, as commonly encountered in classical deep learning frameworks. In contrast, our framework actually has fewer parameters, thanks to the robust data storage and expressive capabilities of the variational quantum circuits. Furthermore, our framework exhibits high flexibility and adaptability, making it applicable to most of existing deep learning models for image classification.

In summary, the contributions of our work are

- We propose a novel image classification framework based on VQAs, which does not require global pooling before the classification layer and enables models to capture more discriminative details from images using fewer parameters. As far as we know, our work is the first attempt to tackle the problem of global pooling in the classical image classification framework by employing quantum machine learning algorithms.
- We evaluate our proposed framework through state vector simulation on four challenging datasets, namely Croatian Fish [60], Aircraft [61], Breast Ultrasound image [62], and Apples or Tomatoes [63]. Experimental results demonstrate that our framework significantly outperform the classical framework.
- We implement shot-based simulation of our method on the AOT dataset to investigate its performance in the presence of shot noise. We observe that more measurements do not necessarily lead to optimal results. Instead, selecting an appropriate number of measurements can yield superior results compared to state vector simulation.



(a) Average pooling



(b) Max pooling

Fig. 1 Examples of two types of pooling operations. **a** Average pooling with a pooling area of size 2x2 and stride of 2. **b** Max pooling with a pooling area of size 2x2 and stride of 2

2 Method

2.1 Preliminaries

2.1.1 Global pooling

Pooling is a common operation in image classification models that plays a vital role in reducing spatial dimensions and capturing invariant features. It aims to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two commonly used pooling methods are average pooling and max pooling. Average pooling calculates the average value of the feature within each pooling region while max pooling selects the maximum value of the feature. Examples of these two pooling approaches are shown in Fig. 1.

Global pooling is a variant of pooling operation where the kernel size is equal to the size of the input feature map. Essentially, it computes a single value for each feature channel by summarizing the entire feature map. The two most common types of global pooling are global average pooling (GAP) and global max pooling (GMP), as illustrated in Fig. 2, which compute the average value and maximum value of each feature channel, respectively. Global pooling is often used as a bridge between the backbone model and fully connected layers in classical image classification architectures. They help to reduce overfitting by minimizing the total number of parameters in the model. However, they can also lead to substantial loss of spatial information and critical details

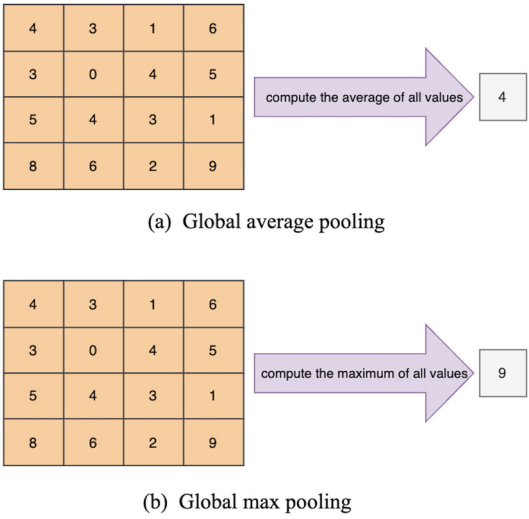


Fig. 2 Examples of two types of global pooling operations. **a** Global average pooling. **b** Global max pooling

by transforming an entire feature map to a single value. This prevents the model from capturing local details effectively and thus limits the model performance.

2.1.2 Quantum basics

Here, we provide a brief introduction to some fundamental concepts of quantum computing that are essential for understanding this paper.

- *Qubit and Quantum State* The qubit is the basic unit of information in quantum computing. Unlike a classical bit which has a value of either 0 or 1, a qubit can exist in a combination of the two states. This is called superposition. The quantum state of a qubit is described by a vector in a two-dimensional Hilbert space, commonly represented as

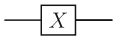
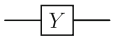
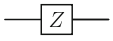
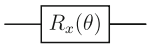
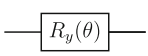
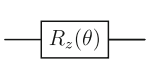
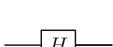
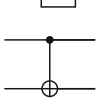
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

  where $|0\rangle$ and $|1\rangle$ are two computational basis states, and α and β are complex probability amplitude corresponding to each basis state, satisfying

$$|\alpha|^2 + |\beta|^2 = 1.$$

- *Quantum gate and Quantum circuit* Quantum gates are unitary operators or equivalently unitary matrices which are applied to states of qubits and perform unitary transformation. Common single-qubit quantum gates include the Hadamard gate, Pauli gates, and their corresponding rotation gates. In particular, single-qubit rotation gates are parametric gates that depend on one parameter and allow for a more flexible range of operations. These rotation gates, denoted by $R_i(\theta)$, perform a rotation around the i axis on the Bloch sphere [64] by an angle θ, where $i \in \{x, y, z\}$

Table 1 Summary of common single-qubit and two-qubit quantum gates

Quantum gate	Matrix expression	Symbol
Pauli-X gate	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	
Pauli-Y gate	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	
Pauli-Z gate	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	
R_x gate	$\begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$	
R_y gate	$\begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$	
R_z gate	$\begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$	
Hadamard gate	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	
CNOT gate	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	

and $\theta \in [0, 2\pi)$. Other examples of common gates include two-qubit gates such as the CNOT gate which can induce quantum entanglement between qubits. A summary of these quantum gates is found in Table 1.

A quantum circuit is a collection of interconnected quantum gates, and it is constructed by arranging the gates in a specific order to perform desired computations or quantum algorithms. Quantum gates and circuits play a crucial role in quantum computing, enabling the manipulation and processing of quantum information.

- *Observable and Measurement* Observables in quantum physics are self-adjoint operators acting on quantum states. The eigenvalues of observables are real numbers and represent physical quantities (e.g., position, energy and momentum) that can be measured. A measurement is the testing or manipulation of a physical system in order to yield a numerical result of the observable. A qubit can be in a superposition state (i.e., combination of all possible basis states) before the measurement. But after the measurement, it will collapse to one of those basis states with the probability obtained as the norm square of the corresponding probability amplitude (e.g., $|\alpha|^2$ for the state $|0\rangle$ in Eq. (1)). Therefore, measurement results in quantum physics are probabilistic in general.

2.2 Variational quantum algorithms

Variational quantum algorithms (VQAs) are a class of quantum algorithms that leverage the strengths of both classical and quantum computing to solve computational problems. Unlike traditional quantum algorithms that rely on precise and exact quantum operations, VQAs make use of variational quantum circuits, which provide a more flexible and adaptable approach to quantum computation, particularly in the context of noisy or imperfect quantum systems. A variational quantum circuit is comprised of three components.

- *Encoding module* The encoding module is responsible for mapping classical data into a quantum state representation. It encodes the classical input into a quantum state that can be effectively manipulated and processed by the subsequent modules. Examples of quantum encoding methods include amplitude encoding, angle encoding, and basis encoding. A comprehensive review of these methods can be found in [65]. In the current work, we focus on amplitude encoding which will be introduced in detail in subsection 2.3. Let us denote by $E(x)$ the encoding operator where x is the input vector. Then the encoded quantum state is obtained as

$$|x\rangle = E(x)|0\rangle. \tag{2}$$

- *Parameterized module* Following the encoding process, the parameterized module applies a series of single- and multi-qubit gates to the encoded quantum state. Single-qubit gates are primarily parametric rotation gates (e.g., R_x gate). Multi-qubit gates, usually CNOT gates or parametric controlled rotation gates, are used to generate correlated or entangled quantum states. This combination of single- and multi-qubit gates collectively forms a parameterized layer in the quantum circuit designed to extract task-specific features. This layer can be repeated multiple times to extend the feature space. In this sense, the parameterized module is implemented by a parameterized quantum circuit which is also referred to as an ansatz in the context of quantum computing. Let us denote all unitary operations within the parameterized module as $U(\theta)$, the resulting quantum state will be

$$|x, \theta\rangle = U(\theta)|x\rangle \tag{3}$$

  where θ represent all trainable parameters in the module.
- *Decoding module* Finally, the decoding module extracts useful information from the final state of the quantum system. To be more specific, this module transforms the final quantum state into a classical vector $f(x, \theta)$ by using a map:

$$\mathcal{M}: \quad |x, \theta\rangle \to f(x, \theta). \tag{4}$$

  This classical vector $f(x, \theta)$ is actually the expectation value of certain local observables $A^{\otimes m}$ (e.g., Pauli-Z operators $\sigma_z^{\otimes m}$) obtained from repeated measurements

$$f(x, \theta) = \langle x, \theta | A^{\otimes m} | x, \theta \rangle, \tag{5}$$

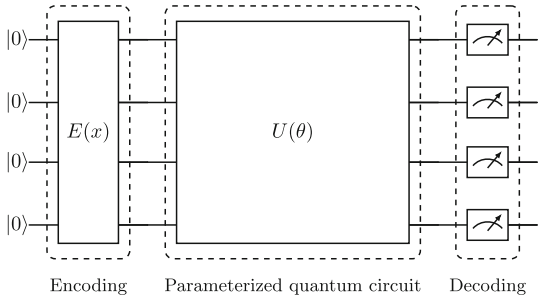


Fig. 3 An example of a variational quantum circuit. The classical input data x is encoded into a quantum state by the encoding module $E(x)$. This encoded state is then transformed by the following parameterized quantum circuit $U(\theta)$, namely the parameterized module, where θ are learnable parameters. Finally, the decoding module extracts classical information from the final quantum state by performing quantum measurements

where $\otimes$ denotes the tensor product operation of quantum operators and m indicates the number of qubits the operator A acts on. Here, m is equal or smaller than the total number of qubits n in the quantum system. The classical vector $f(x, \theta)$ can be used as the input features for the subsequent layer in the model.

The structure of the variational quantum circuit is illustrated in Fig. 3.

VQAs employ a hybrid quantum–classical approach for optimization. The variational quantum circuit is executed on a quantum computer to generate measurement outcomes, while classical optimization techniques, such as stochastic gradient descent or Adam, are employed to update the parameters of the circuit. In this sense, variational quantum circuits can be seen as quantum analogs of classical neural networks and thus can easily be used for various machine learning tasks, such as classification and regression. It has been pointed out that variational quantum circuits are more expressive than classical neural networks [45, 58, 66].

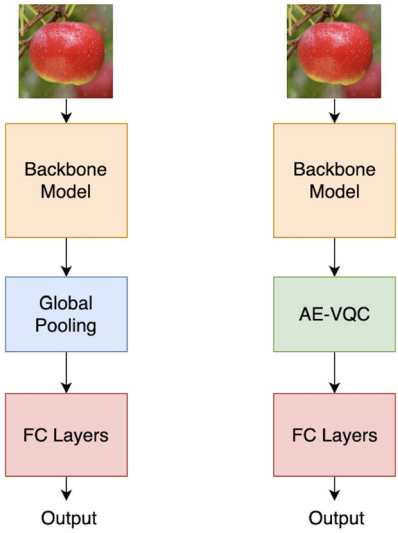
2.3 Amplitude encoding

Amplitude encoding is a technique used in quantum computing to encode classical data into the amplitudes of quantum states. To be more specific, it encodes a normalized classical N-dimensional vector x into amplitudes of an n-qubit quantum state with $n = \lceil \log_2 N \rceil$:

$$|\psi(x)\rangle = \sum_{i}^{N} x_i |i\rangle \tag{6}$$

where $|\psi(x)\rangle$ is the encoded state and $|i\rangle$ is the i-th computational basis state. The classical vector x must satisfy normalization condition: $|x|^2 = 1$, as it represents the amplitudes of a quantum state.

Amplitude encoding is a particularly efficient data encoding scheme as it enables a quantum computer to process a large amount of data simultaneously due to the



(a) Classical Framework (b) Proposed Framework

Fig. 4 Comparison of two image classification frameworks. **a** Classical framework. The input images are transformed by a backbone model into relevant features. The following global pooling module downsamples these feature maps and outputs a fix-length vector which is fed into fully connected (FC) layers for the final classification. **b** Proposed framework. The global pooling module is replaced by a variational quantum circuit with amplitude encoding which we denote by AE-VQC. The feature maps extracted by the backbone model are directly fed into this AE-VQC without dimensionality reduction. The outputs of the AE-VQC are transformed into classification results via FC layers

principles of superposition and interference. One common use of amplitude encoding is in quantum machine learning algorithms, where high-dimensional data vectors are encoded into the amplitudes of a quantum state. This allows quantum algorithms to operate on high-dimensional data in ways that could potentially be more efficient than classical algorithms.

2.4 Image classification framework based on variational quantum algorithms

The classical image classification framework, as shown in Fig. 4a, generally consists of a backbone model, a global pooling layer and several fully connected layers. Specifically, the backbone model is applied to input images to extract relevant features. The following global pooling module is then used to reduce the spatial dimensions of these feature maps. Finally, fully connected layers learn higher-level representations from these pooled features and perform the classification. As previously mentioned, the global pooling operation results in a significant loss of information, potentially degrading model performance.

To address this problem, we propose in this paper a novel image classification framework, as depicted in Fig. 4b. This framework is based on variational quantum algorithms and it is a departure from the classical framework. The primary idea of this

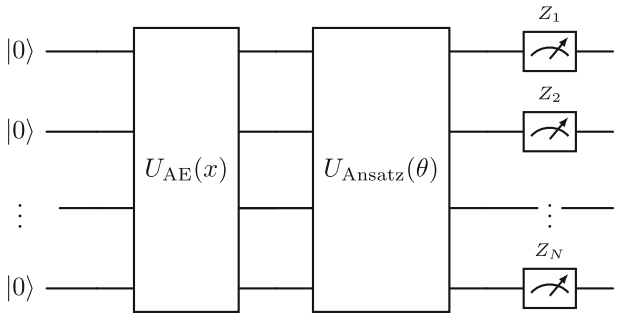


Fig. 5 Architecture of the AE-VQC with N qubits. $U_{\mathrm{AE}}(x)$ represents the amplitude encoding operator which encodes the input classical data x into a quantum state. $U_{\mathrm{Ansatz}}(\theta)$ denotes the ansatz with trainable parameters θ which unitarily transforms the encoded quantum state and explore complex feature space. $Z_1, \ldots, Z_N$ are the observables measured on the quantum state of N qubits

framework is to eliminate the global pooling operation commonly used at the end of the classical framework and replace it by a variational quantum circuit with amplitude encoding which we refer to as AE-VQC. The other parts of the proposed framework remain the same as the classical framework.

Next, we introduce the details of the AE-VQC in our framework, as shown in Fig. 5. Due to the absence of the global pooling layer following the backbone model, the input size of the AE-VQC becomes extremely large. For example, in the ResNet-18 model, the size of the feature maps after the last convolutional layer is $512 \times 7 \times 7 = 25088$. Therefore, amplitude encoding is selected in the AE-VQC to effectively store a large amount of information with fewer qubits. In the case of ResNet-18, only $\lceil \log_2 25088 \rceil = 15$ qubits are needed to encode those feature maps. If we were to use other data encoding methods, the variational quantum circuit would require a much larger number of qubits (e.g., 25088 qubits required by angle encoding for the case of ResNet-18), which is impractical in the current NISQ era. After the classical information is encoded into an initial quantum state, an ansatz (i.e., a parameterized quantum circuit) is used to transform this quantum state. The design of the ansatz is arbitrary. Here, we consider two ansatzes: Ansatz 1 and Ansatz 2, as shown in Fig. 6. Ansatz 1 simply applies successive parameterized R_x gates on each qubit. Ansatz 2 transforms the initial state into a uniform superposition state by using the H gate, followed by repeatedly applying a module composed of R_x and CNOT gates. With the introduction of the CNOT gate, Ansatz 2 may create quantum entanglement. Finally, in the decoding layer of the AE-VQC, we measure the Pauli-Z operators and use their expectation values as outputs, which are then passed into the subsequent classical layers.

Our proposed architecture has three advantages. First, with the absence of global pooling, the complete information extracted by the backbone model is preserved and can be sufficiently exploited for classification. This is crucial for image classification tasks requiring discriminative and fine-grained details. Second, our framework generally involves fewer parameters. In the AE-VQC, each parameterized quantum gate corresponds to one parameter, and each gate can only operate on one or two qubits. This locality helps reduce the number of parameters. In contrast, classical

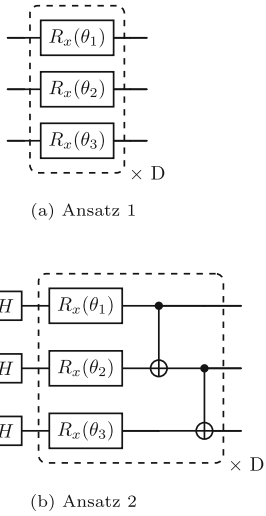


Fig. 6 Two types of ansatzes used in this work. The single-qubit gate $R_x(\theta)$ represents a rotation around the x axis of the Bloch sphere by an angle of θ, where $\theta \in [0, 2\pi)$ is a trainable parameter. Ansatz 2 contains two-qubit CNOT gates which might create quantum entanglement. The dashed box indicates a single circuit layer that can be repeated D times to enhance the expressive power of the ansatz

fully connected neural networks require each neuron to has connection weights with all neurons in the next layer, leading to an extensive amount of parameters. Also, the number of qubits in the AE-VQC is exponentially smaller than the size of the input features thanks to amplitude encoding, and the number of parameterized gates can be substantially reduced by designing an appropriate ansatz. For instance, in the case of 10-class classification using the ResNet-18 model, the total number of parameters after the last convolution layer is $15D+15\times 10 = 15(D+10)$ if Ansatz 1 is employed for the AE-VQC in the proposed framework. Here, $15D$ is the number of quantum parameters from the AE-VQC, where D indicates the number of repeated layers in the ansatz, and 15×10 is the number of classical parameters from the fully connected layer. Since choosing $D \leq 10$ is sufficient to ensure the expressive power of the AE-VQC, $15(D + 10)$ is generally much smaller than the number of parameters after the last convolution layer in the classical ResNet-18 model, which is $512 \times 10 = 5120$. Furthermore, in light of the strong expressibility of the AE-VQC, our framework requires less number of fully connected layers than the classical framework for some specific tasks, which also contributes to decreasing the number of parameters. Third, our architecture has high trainability and is well suited for implementation on future quantum devices. In the classical image classification framework, the backbone model typically generates feature maps with a spatial size of 7x7 and channel counts of 512, 1024, 2048, or 2560 [14–17, 21, 23, 67, 68]. This design strikes a balance between preserving important spatial information, enhancing feature representation capability, and improving computational efficiency, which has been proven to achieve optimal performance and resource utilization in practice. Therefore, the AE-VQC requires a number of qubits between $\lceil \log_2 512 \times 7 \times 7 \rceil = 15$ and $\lceil \log_2 2560 \times 7 \times 7 \rceil = 17$. With the continuous advancements in future quantum hardware, the realization of such small- to medium-scale quantum systems will become more feasible. Also, quantum circuits of this scale will help reduce the risk of encountering barren plateaus [69, 70].

3 Experiments

In this section, we conduct two sets of experiments. First, we utilize state vector simulation to validate the advantages of our proposed method. Then, we explore the impact of shot noise on our method through shot-based simulation.

3.1 Datasets and evaluation metrics

We evaluate the proposed approach on the following four publicly available benchmarks.

- *Croatian Fish dataset* includes 794 images of 12 fish species in their unconstrained natural habitat, showcasing a broad spectrum of shapes, sizes, and poses. This dataset is particularly challenging due to the high visual similarity among fish species, background clutter, and varying lighting conditions. We randomly select 80% of the data as the training set and 20% as the test set.
- *Aircraft dataset* is a collection of 10,000 images of airplanes, spanning across 100 distinct aircraft models. This dataset is designed specifically for fine-grained visual classification and is a part of the ImageNet 2013 FGVC challenge. It provides a complex classification task due to subtle visual differences between aircraft models, and significant variability in aircraft design and branding. Due to resource limitations, we select a subset of the dataset, which includes 8 variants of the Boeing 737 aircraft model ranging from 737–200 to 737–900. The training and test sets contain 533 and 267 images, respectively.
- *Breast Ultrasound image (BUSI) dataset* consists of 780 breast ultrasound images among women in ages between 25 and 75 years old. This dataset is categorized into three classes: normal, benign, and malignant images. The challenge of this dataset lies in differentiating between benign and malignant tissue, a task of significant importance in the early detection of breast cancer. We find an identical image in both of the benign and malignant categories, making it impossible to determine which category these two images belong to. To avoid impacting the experiment results, we remove both images from the dataset. Finally, the training and test sets include 620 and 158 images, respectively.
- *Apples or Tomatoes (AOT) dataset* is designed for a binary classification task which focuses on classifying images as either apples or tomatoes. Despite the apparent simplicity, the task is non-trivial due to similarities in color and shape between the two classes, especially when the fruits are partially obscured or in unusual orientations. The dataset consists of 294 images in the training set and 97 images in the test set.

We choose accuracy and macro-averaged F1 score over the test set as evaluation metrics to evaluate and compare our proposed framework against the classical framework. The macro-averaged F1 score is a metric commonly used to evaluate the performance of multi-class classification models, especially when dealing with imbalanced datasets.

Table 2 Detailed architectural specifications of the ansatzes used in the experiments. D indicates the number of repeated layers in the ansatz

Dataset	Model	Ansatz	D
Croatian fish	ResNet-18_q	Ansatz 2	1
	MaxViT-T_q	Ansatz 1	1
Aircraft	ResNet-18_q	Ansatz 1	1
	MaxViT-T_q	Ansatz 1	1
BUSI	ResNet-18_q	Ansatz 1	1
	MaxViT-T_q	Ansatz 1	1
AOT	ResNet-18_q	Ansatz 1	1
	MaxViT-T_q	Ansatz 1	3

It is calculated by first computing the F1 score for each class, defined as

$$\mathrm{F1}_i = 2 \times \frac{\mathrm{precision}_i \times \mathrm{recall}_i}{\mathrm{precision}_i + \mathrm{recall}_i}, \tag{7}$$

and then averaging these scores across all classes:

$$\text{macro-averaged F1} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{F1}_i, \tag{8}$$

where N is the number of classes. This approach treats each class equally and prevents classes with a larger number of samples from dominating the entire evaluation result, thus providing a fairer assessment of the overall model performance across all classes.

3.2 Training setup

3.2.1 State vector simulation

In the first experiment, we implement a state vector simulation to validate the performance of our proposed framework. We utilize out method to modify two types of classic image classification models: ResNet and MaxViT, both of which have achieved SOTA results for the task of image classification. ResNet is a family of image classification models based purely on convolutional neural networks. It is well known for the introduction of skip connections that bypass or shortcut across convolutional layers. The ResNet architecture serves as a fundamental building block in many mainstream computer vision models and represents a significant milestone in the development of deep learning. MaxViT is a family of hybrid (CNN + ViT) image classification models that achieves better performances than both SOTA ConvNets and Transformers across a wide range of tasks such as image classification, object detection, and segmentation. This model employs a multi-axis attention mechanism to better adapt to arbitrary input resolutions with only linear computational complexity.

Due to the resource constraints, we select lightweight ResNet-18 and MaxViT-T models from these two types of architectures as baseline models to test our proposed

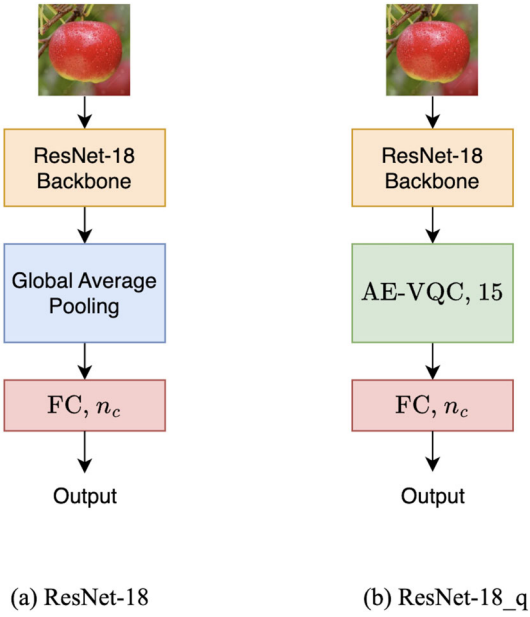
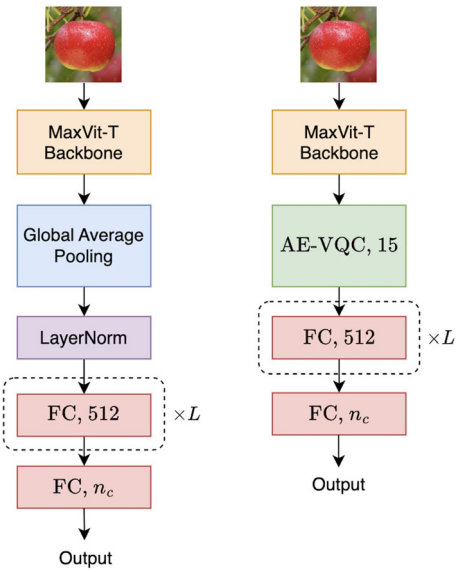


Fig. 7 Comparison of ResNet-18 and ResNet-18_q. **a** ResNet-18. The model processes input images of size $224 \times 224 \times 3$ through a ResNet-18 backbone network. Following the backbone, a global average pooling (GAP) layer reduces spatial dimensions, outputting features of size 512. These pooled features are then passed through a single fully connected (FC) layer, which directly outputs n_c classification scores, corresponding to the n_c classes in the classification task. **b** ResNet-18_q. The model follows the same architecture as ResNet-18, except that the GAP layer is replaced with a AE-VQC with 15 qubits. Since the output dimension of the AE-VQC is 15, the input size to the FC layer is reduced from 512 to 15

Table 3 Optimal number L of FC layers before the final FC layer in both MaxViT-T and MaxViT-T_q in the experiments

Dataset	Model	L
Croatian fish	MaxViT-T	1
	MaxViT-T_q	1
Aircraft	MaxViT-T	1
	MaxViT-T_q	1
BUSI	MaxViT-T	1
	MaxViT-T_q	0
AOT	MaxViT-T	1
	MaxViT-T_q	0

framework. The architectures of these two models are illustrated in Figs. 7a and 8a, respectively. ResNet-18 employs a straightforward approach, appending a global average pooling (GAP) layer to the backbone, followed by a single fully connected (FC) layer that directly outputs n_c classification results, where n_c represents the number of classes. On the other hand, MaxViT-T adopts a more elaborate structure. After the GAP layer, the model applies a layer normalization operation, followed by a stack of L fully connected layers, each outputting 512 features. In our experiments, this L can be either 0 or 1. The output of this stack is then passed through a final FC layer for classification. We refer to the quantum counterparts of these two classic models as ResNet-18_q and MaxViT-T_q, respectively. Figure 7b illustrates the architecture of ResNet-18_q, and Fig. 8b shows the structure of MaxViT-T_q. In particular, MaxViT-T_q a does not add layer normalization after the AE-VQC, as the output range of the AE-VQC is between -1 and 1. Both ResNet-18 and MaxViT-T yield a feature map



(a) MaxVit-T (b) MaxVit-T_q

Fig. 8 Comparison of MaxViT-T and MaxViT-T_q. **a** MaxViT-T. The model takes images of size 224 × 224 × 3 as input, which are initially processed by a MaxViT-T backbone network for feature extraction. The resulting feature maps are then compressed via a global average pooling (GAP) layer, yielding a 512-dimensional feature vector. These pooled features are then passed through a layer normalization operation, followed by L fully connected (FC) layers, each producing 512 features, with $L = 0, 1$. The output of these FC layers is fed into a final FC layer that produces n_c classification scores. **b** MaxViT-T_q. The model adopts the architecture of MaxViT-T but replaces the GAP layer with a AE-VQC using 15 qubits. The output dimension of the AE-VQC is 15, consequently reducing the input size to the first FC layer from 512 to 15. Additionally, since the output range of the AE-VQC is between -1 and 1, it is not followed by layer normalization. Similarly, the hyperparameter L here is either 0 or 1

size of $512 \times 7 \times 7 = 25088$ prior to the GAP layer. Hence, in the corresponding quantum models, the number of qubits required for the AE-VQC is $\lceil \log_2 25088 \rceil = 15$. The ansatz used in the AE-VQC varies according to the model and dataset. Detailed architectural specifications of these ansatzes are presented in Table 2. We train each model from scratch for 200 epochs using a mini-batch size of 32 and the Adam optimizer with a learning rate of 0.001. In addition, as both MaxViT-T and MaxViT-T_q include a hyperparameter L, we list the optimal values of it for these two models across different datasets in Table 3.

3.2.2 Shot-based simulation

In the previous experiment, we employ a state vector simulation. This means that the expectation values of observables and the gradients of quantum parameters are exactly calculated. However, on actual quantum hardware, the expectation values of observables can only be estimated by sampling a finite number of shots (i.e., the number

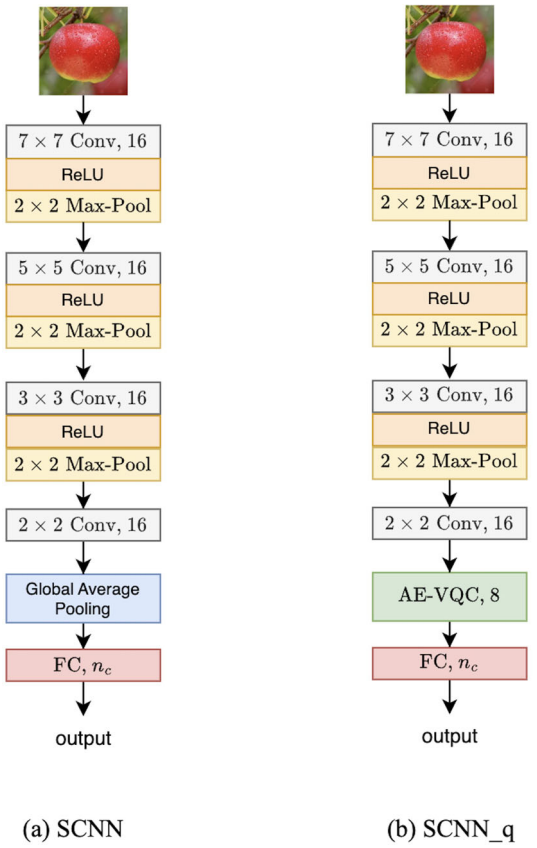


Fig. 9 Comparison of SCNN and SCNN_q. **a** SCNN. A simplified CNN model for image classification. The network processes 224 × 224 × 3 input images through four convolutional layers. The first two layers are followed by a ReLU activation function and a 2 × 2 max pooling layer with a stride of 2, gradually reducing spatial dimensions. The initial layer transforms the 3-channel input into 16 channels using a 7x7 kernel with a stride of 2. The second layer employs a 5 × 5 kernel with a stride of 2. The third and fourth convolutional layers utilize 3 × 3 and 2 × 2 kernels, respectively, both with a stride of 1. A global average pooling (GAP) layer then condenses the feature maps into a 1 × 1 × 16 representation. Finally, a fully connected (FC) layer with 16 input features outputs n_c classification scores. **b** SCNN_q. The quantum counterpart of SCNN. It modifies the SCNN by replacing the GAP layer with a AE-VQC, which outputs a feature vector of size 8. Consequently, the FC layer's input size is reduced from 16 to 8, while retaining the rest of the SCNN architecture

of repeated executions of a quantum circuit). These probabilistic results introduce noise into the gradient calculations. To study how our quantum algorithms perform on quantum hardware, we need to consider the impact of shot noise. Therefore, we conduct the second experiment using a shot-based simulation.

In the case of shot-based simulation, using gradient-based optimizers (e.g., Adam used in this work) to train hybrid quantum–classical algorithms is very time-consuming. This is primarily because the calculation of quantum gradients requires the parameter shift rule method [58, 71], which necessitates a large number of cir-

cuit executions. Given resource constraints, training a ResNet-18_q or MaxViT-T_q with a 15-qubit circuit is unfeasible for our study as it would take an extremely long time. Therefore, we design a simplified image classification convolutional neural network model, as shown in Fig. 9a. It consists of four convolutional layers. The first three convolutional layers are followed by a ReLU activation and then a 2×2 max pooling layer with a stride of 2. The first convolutional layer takes an input with 3 channels and outputs 16 channels, using a 7x7 kernel with a stride of 2. The second and third convolutional layers employ 5×5 and 3×3 kernels with strides of 2 and 1, respectively. The final convolutional layer utilizes a 2×2 kernel with a stride of 1. After the convolutional layers, a GAP layer reduces the spatial dimensions to 1x1. Finally, a FC layer with 16 input features produces the classification results. We refer to this simplified CNN model as SCNN. Similarly, we apply our proposed method to transform the SCNN model into a quantum model, referred to as SCNN_q, by substituting the GAP layer with a AE-VQC. The architecture of SCNN_q is illustrated in Fig. 9b. Since the feature map preceding the GAP layer in SCNN has dimensions of $4 \times 4 \times 16 = 256$, the AE-VQC requires $\lceil \log_2 256 \rceil = 8$ qubits. We chose Ansatz1 with a depth of 1 as the ansatz circuit for the AE-VQC. To investigate the effects of shot noise on model performance, we train and test this quantum model on both a state vector simulator and a shot-based simulator. These two trained quantum models are referred to as SCNN_q_sv and SCNN_q_sb, respectively. Again, due to computational resource constraints, we limit our study to the AOT dataset for this experiment. We train all models on this dataset for 100 epochs using a mini-batch size of 2 and the Adam optimizer with a learning rate of 0.001.

3.3 Experimental environment

We conduct our experiments on a local computer with a M1 Pro 10-core CPU by using PennyLane [72] and PyTorch [73]. PennyLane is a quantum machine learning open-source library which allows for quantum differentiable programming. With a comprehensive set of features, simulators, and hardware, PennyLane enables users to easily build, optimize, and deploy quantum–classical applications. To build the tested models, we implement the classical modules with PyTorch and quantum modules using PennyLane. For the first experiment, we choose the PennyLane's standard state vector simulator *default.qubit* as the backend for executing quantum circuits in our hybrid models. This simulator supports both back-propagation [74, 75] and adjoint [76] differentiation methods for calculating quantum gradients using the PyTorch interface. For the second experiment, we select the Qulacs simulator [77], a high-performance shot-based simulator accessible through the PennyLane-Qulacs plugin [72]. Quantum gradient computation on this simulator is performed using the parameter shift rule method.

Table 4 Performance comparisons of ResNet-18 and ResNet-18_q on four public datasets

Dataset	Model	Params*	Acc (%)	F1 (%)	Epoch Time (s)
Croatian fish	ResNet-18	6,156	91.52	91.77	76.34
	ResNet-18_q	222	**94.55**	**94.56**	106.82
Aircraft	ResNet-18	4,104	31.46	31.74	69.46
	ResNet-18_q	143	**36.33**	**36.93**	81.40
BUSI	ResNet-18	1,539	80.38	79.10	77.21
	ResNet-18_q	63	**86.08**	**84.63**	90.40
AOT	ResNet-18	1,026	82.47	82.35	36.01
	ResNet-18_q	41	**87.63**	**87.52**	42.95

Acc and F1 are accuracy and macro-averaged F1 score, respectively, reported on test sets. Epoch time refers to the time taken for the model to complete one epoch on the entire training set. Params* indicates the number of parameters following the backbone model. ResNet-18 and ResNet-18_q have exactly the same backbone model. The best accuracy and F1 score for each dataset are shown in bold

3.4 Results

3.4.1 State vector results

First, we summarized the state vector results in Tables 4 and 5. It can be observed that both quantum models outperform their classical counterparts by significant margins across all datasets, which is in consistence with our expectation. For instance, ResNet-18_q exhibits improvements in test accuracy of up to 5.70% and in test F1 score of up to 5.53%, over the classical ResNet-18 model. More remarkably, MaxViT-T_q provides up to 9.21% higher accuracy and up to 15.79% higher F1 score, when compared with MaxViT-T. These performance improvements mainly arise from the capability of our framework to preserve and utilize the full feature maps obtained by feature extractors of image classification models. It is worth noting that in most of our experiments, an ansatz depth of 1 is sufficient for our quantum model to demonstrate superior performance. This shallow ansatz can enhance the efficiency of model training and is also more suitable for implementation on near-term quantum computers. Additionally, it is very interesting to note that Ansatz 1 generally performs better than Ansatz 2 in our experiments, even though Ansatz 1 does not utilize entanglement. In fact, we also attempted to introduce entanglement in Ansatz 1 by adding entangling gates (e.g., CNOT gates) after the single-qubit gates. However, the experimental results indicate that this degrades the performance of our quantum model. So we finally did not include entanglement in Ansatz 1. While this result does not demonstrate the importance of entanglement for quantum model performance, as suggested by previous literature [47, 51, 52, 78], it is consistent with the findings of a recent work [79]. In [79], the authors conducted a series of experiments and discovered that removing entanglement from quantum models often leads to equally good or even better model performance. Therefore, whether entanglement is a decisive factor in determining the performance of quantum models remains an open question.

Table 5 Performance comparisons of MaxViT-T and MaxViT-T_q on four public datasets

Dataset	Model	Params*	Acc (%)	F1 (%)	Epoch Time (s)
Croatian fish	MaxViT-T	269,836	71.52	68.54	607.40
	MaxViT-T_q	14,363	**76.97**	**75.15**	624.91
Aircraft	MaxViT-T	267,784	13.11	6.56	552.93
	MaxViT-T_q	12,311	**22.32**	**22.35**	586.32
BUSI	MaxViT-T	265,219	78.48	77.08	588.87
	MaxViT-T_q	63	**84.18**	**81.97**	629.15
AOT	MaxViT-T	264,706	84.54	83.98	296.13
	MaxViT-T_q	107	**87.63**	**87.56**	316.57

The best accuracy and F1 score for each dataset are shown in bold

Furthermore, as shown in Tables 4 and 5, quantum models have fewer parameters than classical models, even without the use of the global pooling layer after the backbone. In particular, MaxViT-T_q achieves better classification performances than MaxViT-T on BUSI and AOT Datasets, using only 0.2% and 0.4% of the number of parameters, respectively. This is primarily due to the powerful data storage and expressive capabilities of the variational quantum circuit with amplitude encoding. It should be noted that we only present the parameter count of the layers after the backbone model, since both the proposed and classical frameworks use exactly the same backbone model.

In addition, we report the time required to complete one epoch on the training set for each dataset in Tables 4 and 5. Overall, quantum models show relatively lower training efficiency compared to classical models, with an increase of 2.88–39.93% in epoch time. Notably, ResNet-18_q exhibits the lowest efficiency on the Aircraft dataset, primarily due to using the more complex Ansatz 2. This discrepancy in performance stems from the fact that quantum models are trained on quantum simulators, which simulate quantum circuits on classical computers. This simulation process demands substantial computational resources and memory, especially as the number of qubits and circuit depth increase. However, it is worth noting that the computational performance of different simulators can vary by orders of magnitude depending on the simulation task, regardless of problem size [80, 81]. Choosing a simulator suitable for our quantum models and tasks might potentially reduce the training time gap between quantum and classical models. Another factor hindering the efficiency of our quantum models is the use of amplitude encoding to load classical data onto the amplitudes of quantum states. To ensure precise initial quantum states, amplitude encoding often requires very complex circuits, and the circuit depth increases exponentially with the number of qubits. This leads to one of the main bottlenecks for quantum machine learning algorithms. Therefore, exploring more efficient amplitude encoding implementations to enhance the training speed of our model is an important area for future work, which we discuss in Sect. 4.

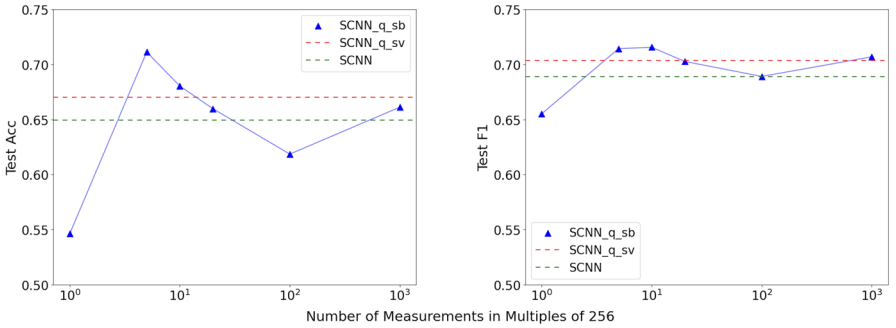


Fig. 10 Impact of number of measurements on the performance of model SCNN_q_sb. The x-axis shows the number of measurements in multiples of 256, plotted on a base-10 logarithmic scale, where 256 is the number of all possible quantum states that the 8-qubit quantum circuit in SCNN_q_sb can represent. The y-axes in **a** and **b** represent the test accuracy and test F1 score on the AOT dataset, respectively. The results of models SCNN_q_sv and SCNN are also displayed as benchmarks

3.4.2 Shot-based results

To investigate the impact of shot noise on the performance of our proposed framework, we compare the results of the model SCNN_q_sb with shot-based simulation and the model SCNN_q_sv with state vector simulation on the AOT dataset in Fig. 10. We train six SCNN_q_sb models, using 256, 1280, 2560, 5120, 25600, and 256000 as shot numbers, respectively, to estimate expectation values. In particular, 256 is the number of quantum states that can be represented by 8 qubits, which is the number of qubits used in the model SCNN_q_sb.

We find that when the shot number is large, such as 25600 and 256000, the test accuracy and test F1 of the SCNN_q_sb model continuously approach those of SCNN_q_sv. This is because a large number of measurements leads to more precise estimates of expectation values. Theoretically, shot-based simulation with an infinite number of measurements is equivalent to state vector simulation. Conversely, when the shot number is small, such as 256, the performance of the SCNN_q_sb model significantly differs from that of the SCNN_q_sv model, with a 10.31% and 4.85% reduction in test accuracy and test F1 score, respectively. This performance gap stems from the severe lack of a sufficient number of circuit executions to accurately estimate the expectation value, introducing considerable noise into parameter optimization and consequently leading to a large optimization error.

Furthermore, we observe an interesting result. When the shot number is 1280 and 2560, the SCNN_q_sb model achieves its best performance, even surpassing the SCNN_q_sv model. We hypothesize that this is because this level of measurements introduces moderate shot noise into the gradients of quantum parameters, which enables the Adam optimizer to more easily escape local minima or saddle points, and converge toward the global minimum. Similar arguments can be found in both machine learning literature [82–85] and quantum machine learning literature [86]. In [84], the authors found that adding noise to gradients consistently improved the performance

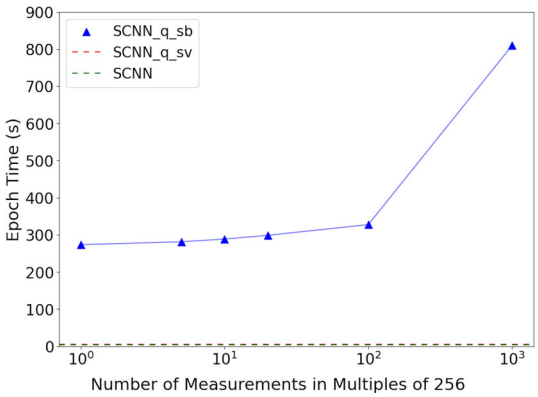


Fig. 11 Impact of number of measurements on the time to perform one epoch of training for model SCNN_q_sb. The x-axis uses a base-10 logarithmic scale

of the Neural Programmer model when trained with Adam for the task of question answering from tables, regardless of whether the tasks were simple or challenging. In [86], when Adam was used to optimize ansatz parameters of variational quantum linear solver algorithms for three problem instances, shot-based results were continuously better than state vector results.

The shot-based result on the AOT dataset implies that increasing measurements for expectation value estimation in the SCNN q sb model does not guarantee optimal performance. Instead, executing $2^n \times 5$ or $2^n \times 10$ circuits can not only save computational resources but also lead to better results than state vector simulation, where n represents the number of qubits used in the model. In fact, due to the perturbation from both mini-batch noise and shot noise, we refer to the optimization method used in this experiment as a “doubly perturbed" Adam optimizer. Whether this optimizer can consistently benefit our proposed model, and what level of shot noise needs to be injected into the gradient, still requires further investigation using more models and datasets.

Additionally, to demonstrate the advantages of quantum models, we still present the results of the classical model SCNN in Fig. 10. We can see that the performance of SCNN_q_sv still surpasses that of SCNN. When the shot number is 1280, 2560, 5120 and 256000, SCNN_q_sb can also outperform SCNN. However, using 1280 shots can significantly improves the model training efficiency, saving 66.23% of training time compared to using 256000 shots, as shown in Fig. 11.

4 Conclusion and discussion

In this work, we propose a novel image classification framework based on variational quantum algorithms. This framework differs from both existing classical and quantum image classification frameworks. The most notable feature of our framework is the elimination of the global pooling module commonly used at the end of the classical image classification network, an operation which reduces model complexity but results in a significant loss of important information. This feature enables our framework to learn discriminative details within images. Moreover, in spite of the

absence of the global pooling, our framework still has fewer parameters compared to the classical framework, preserving the capability to prevent overfitting. This stems from the adoption of the variational quantum circuit with amplitude encoding. We apply our approach to two different types of mainstream image classification models, namely the convolution-based ResNet and hybrid CNN–transformer-based MaxVit. These two benchmark models represent state-of-the-art performances in the domain of image classification. Through extensive state vector simulation experiments on four challenging datasets, our proposed quantum framework exhibits superior performances over the classical framework in terms of classification metrics (e.g., accuracy and F1 score). In addition, the results of the shot-based simulation experiments on the AOT dataset show that doubly perturbed Adam optimizer can further enhance the performance of our proposed method without necessitating a maximal number of measurements. It is worth noting that our framework is highly flexible and can be applied to a wide range of classical image classification models. Furthermore, our framework enjoys high trainability and is well suited for implementation on future quantum devices, as it requires only a fixed small number of qubits and shallow ansatzes.

Despite a series of achievements obtained in this work, there are still some limitations that need to be addressed in the future research. Firstly, amplitude encoding is a crucial component of our method, but its implementation can be challenging, requiring exponentially deep quantum circuits. While efficient implementation of amplitude encoding is still a research area in progress, several approaches based on approximate encoding [87–89] have been proposed. Works [87, 88] train variational quantum circuits with maximum mean discrepancy and fidelity cost functions, respectively, to achieve encoding with $\mathcal{O}(\text{poly}(n))$ gates compared to the exponential $\mathcal{O}(2^n)$ gates required for exact encoding, where n represents the number of qubits. In addition to variational circuits, the study [89] also employs genetic and matrix product states (MPS) algorithms to approximate the desired quantum states, resulting in circuits two orders of magnitude shallower than exact encoding methods. Genetic algorithms reduce gate count and enhance noise resistance, while MPS efficiently represent low-entanglement quantum states, lowering resource requirements for quantum algorithms. Despite the introduction of approximation errors, these methods have shown good classification results. In particular, the authors in [89] demonstrate that their approximate techniques can not only preserve the classification accuracy but also increase adversarial robustness of quantum machine leaning models for image classification. Given that our proposed model is a gradient-based machine learning model which does not require exact computations, a promising direction for future work is to implement these approximate encoding techniques in our model to enhance data processing efficiency while maintaining the classification performance.

With potentially improved computational efficiency achieved through approximate encoding methods, more complex models (e.g., ResNet-18_q and MaxViT-T_q) can be employed on a wider range of datasets to further investigate the impact of shot noise on model performance and validate the potential advantages of doubly perturbed Adam optimizer. For instance, we aim to determine what level of shot noise can be considered as moderate noise, which could help doubly perturbed Adam optimizer improve the classification performance of our quantum models. Furthermore, all experiments in this work are performed using quantum simulators. Future work could involve examining

how our models perform on real quantum hardware, where realistic noise should be taken into account.

Moreover, we can explore more ansatzes for our framework in addition to the two ansatzes used in this paper. Finally, our method is currently limited to the area of image classification. It would be worthwhile to extend this approach to other machine learning tasks such as natural language processing.

Author Contributions Y.C. is responsible for all aspects of the work, including the conception and design of the study, data collection and analysis, performing experiments, interpretation of the results, and writing the manuscript.

Data Availability The data that support the findings of this study are publicly available.

Code Availability The code that supports the findings of this study is available upon request.

Declarations

Conflict of interest The authors have no conflict of interest to disclose.

References

1. Cheng, B., Deng, X.-H., Gu, X., He, Y., Hu, G., Huang, P., Li, J., Lin, B.-C., Lu, D., Lu, Y., et al.: Noisy intermediate-scale quantum computers. Front. Phys. **18**(2), 21308 (2023)
2. Schuld, M., Sinayskiy, I., Petruccione, F.: An introduction to quantum machine learning. Contemp. Phys. **56**(2), 172–185 (2015)
3. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
4. Dunjko, V., Briegel, H.J.: Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Rep. Prog. Phys. **81**(7), 074001 (2018)
5. Zhang, Y., Ni, Q.: Recent advances in quantum machine learning. Quant. Eng. **2**(1), 34 (2020)
6. Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L., Coles, P.J.: Challenges and opportunities in quantum machine learning. Nature Comput. Sci. **2**(9), 567–576 (2022)
7. Varatharajan, R., Manogaran, G., Priyan, M.: A big data classification approach using lda with an enhanced svm method for ecg signals in cloud computing. Multimed. Tools Appl. **77**(8), 10195–10215 (2018)
8. Kerenidis, I., Landman, J., Luongo, A., Prakash, A.: q-means: A quantum algorithm for unsupervised machine learning (2018). arXiv preprint arXiv:1812.03584
9. Otterbach, J., Manenti, R., Alidoust, N., Bestwick, A., Block, M., Bloom, B., Caldwell, S., Didier, N., Fried, E.S., Hong, S., et al.: Unsupervised machine learning on a hybrid quantum computer (2017). arXiv preprint arXiv:1712.05771
10. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum principal component analysis. Nature Phy. (2013). https://doi.org/10.1038/nphys3029

11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural. Inf. Process. Syst. **25**, 1097–1105 (2012)
12. Lin, M., Chen, Q., Yan, S.: Network in network (2013). arXiv preprint arXiv:1312.4400
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
14. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
16. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
17. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114. PMLR (2019)
18. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. Ieee (2009)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł, Polosukhin, I.: Attention is all you need. Adv Neural Inf. Process. Syst. (2017). https://doi.org/10.48550/arXiv.1706.03762
20. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale (2010). arxiv 2020. arXiv preprint arXiv:2010.11929
21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
22. Dai, Z., Liu, H., Le, Q.V., Tan, M.: Coatnet: marrying convolution and attention for all data sizes. Adv. Neural. Inf. Process. Syst. **34**, 3965–3977 (2021)
23. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxvit: Multi-axis vision transformer. In: European Conference on Computer Vision, pp. 459–479. Springer (2022)
24. Zhang, B., Zhao, Q., Feng, W., Lyu, S.: Alphamex: a smarter global pooling method for convolutional neural networks. Neurocomputing **321**, 36–48 (2018)
25. Lowe, S.C., Trappenberg, T., Oore, S.: Logavgexp provides a principled and performant global pooling operator (2021). arXiv preprint arXiv:2111.01742
26. Sun, C., Paluri, M., Collobert, R., Nevatia, R., Bourdev, L.: Pronet: Learning to propose object-specific boxes for cascaded neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3485–3493 (2016)
27. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
28. Jose, A., Lopez, R.D., Heisterklaus, I., Wien, M.: Pyramid pooling of convolutional feature maps for image retrieval. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 480–484. IEEE (2018)
29. Qi, K., Guan, Q., Yang, C., Peng, F., Shen, S., Wu, H.: Concentric circle pooling in deep convolutional networks for remote sensing scene classification. Remote Sens. **10**(6), 934 (2018)
30. Lin, J., Ma, L., Yao, Y.: A fourier domain training framework for convolutional neural networks based on the fourier domain pyramid pooling method and fourier domain exponential linear unit. IEEE Access **7**, 116612–116631 (2019)
31. Qiu, S.: Global weighted average pooling bridges pixel-level localization and image-level classification (2018). arXiv preprint arXiv:1809.08264
32. Zhang, X., Zhang, X.: Global learnable pooling with enhancing distinctive feature for image classification. IEEE Access **8**, 98539–98547 (2020)
33. Behera, A., Wharton, Z., Hewage, P.R., Bera, A.: Context-aware attentional pooling (cap) for fine-grained visual classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 929–937 (2021)
34. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Adv. Neural Inf. Process. Syst. (2015). https://doi.org/10.48550/arXiv.1506.02025

35. Li, P., Xie, J., Wang, Q., Gao, Z.: Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 947–955 (2018)
36. Gao, Z., Xie, J., Wang, Q., Li, P.: Global second-order pooling convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3024–3033 (2019)
37. Wang, Q., Xie, J., Zuo, W., Zhang, L., Li, P.: Deep cnns meet global covariance pooling: Better representation and generalization. IEEE Trans. Pattern Anal. Mach. Intell. **43**(8), 2582–2597 (2020)
38. Song, Y., Sebe, N., Wang, W.: On the eigenvalues of global covariance pooling for fine-grained visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **45**(3), 3554–3566 (2022)
39. McClean, J.R., Romero, J., Babbush, R., Aspuru-Guzik, A.: The theory of variational hybrid quantum-classical algorithms. New J. Phys. **18**(2), 023023 (2016)
40. Bharti, K., Cervera-Lierta, A., Kyaw, T.H., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J.S., Menke, T., et al.: Noisy intermediate-scale quantum algorithms. Rev. Mod. Phys. **94**(1), 015004 (2022)
41. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., et al.: Variational quantum algorithms. Nature Rev. Phys. **3**(9), 625–644 (2021)
42. Dang, Y., Jiang, N., Hu, H., Ji, Z., Zhang, W.: Image classification based on quantum k-nearest-neighbor algorithm. Quant. Inf. Process. **17**, 1–18 (2018)
43. Kerenidis, I., Landman, J., Prakash, A.: Quantum algorithms for deep convolutional neural networks (2019). arXiv preprint arXiv:1911.01117
44. Mari, A., Bromley, T.R., Izaac, J., Schuld, M., Killoran, N.: Transfer learning in hybrid classical-quantum neural networks. Quantum **4**, 340 (2020)
45. Henderson, M., Shakya, S., Pradhan, S., Cook, T.: Quanvolutional neural networks: powering image recognition with quantum circuits. Quant. Mach. Int. **2**(1), 2 (2020)
46. Li, Y., Zhou, R.-G., Xu, R., Luo, J., Hu, W.: A quantum deep convolutional neural network for image recognition. Quant. Sci. Technol. **5**(4), 044003 (2020)
47. Liu, J., Lim, K.H., Wood, K.L., Huang, W., Guo, C., Huang, H.-L.: Hybrid quantum-classical convolutional neural networks. Sci. China Phys., Mech. Astron. **64**(9), 290311 (2021)
48. Henderson, M., Gallina, J., Brett, M.: Methods for accelerating geospatial data processing using quantum computers. Quant. Mach. Intell. **3**(1), 4 (2021)
49. Houssein, E.H., Abohashima, Z., Elhoseny, M., Mohamed, W.M.: Hybrid quantum convolutional neural networks model for covid-19 prediction using chest x-ray images (2021). arXiv preprint arXiv:2102.06535
50. Sagingalieva, A., Kurkin, A., Melnikov, A., Kuhmistrov, D., Perelshtein, M., Melnikov, A., Skolik, A., Von Dollen, D.: Hyperparameter optimization of hybrid quantum neural networks for car classification (2022). arXiv preprint arXiv:2205.04878
51. Hur, T., Kim, L., Park, D.K.: Quantum convolutional neural network for classical data classification. Quant. Mach. Intell. **4**(1), 3 (2022)
52. Senokosov, A., Sedykh, A., Sagingalieva, A., Melnikov, A.: Quantum machine learning for image classification (2023). arXiv preprint arXiv:2304.09224
53. Melnikov, A., Kordzanganeh, M., Alodjants, A., Lee, R.-K.: Quantum machine learning: from physics to software engineering. Adv. Phys.: X **8**(1), 2165452 (2023)
54. Monnet, M., Gebran, H., Matic-Flierl, A., Kiwit, F., Schachtner, B., Bentellis, A., Lorenz, J.M.: Pooling techniques in hybrid quantum-classical convolutional neural networks. In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), vol. 1, pp. 601–610. IEEE (2023)
55. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process. Mag. **29**(6), 141–142 (2012). https://doi.org/10.1109/MSP.2012.2211477
56. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man versus computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw. **32**, 323–332 (2012). https://doi.org/10.1016/j.neunet.2012.02.016. (**Selected Papers from IJCNN 2011**)
57. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017). arXiv preprint arXiv:1708.07747
58. Mitarai, K., Negoro, M., Kitagawa, M., Fujii, K.: Quantum circuit learning. Phys. Rev. A **98**(3), 032309 (2018)
59. Farhi, E., Neven, H.: Classification with quantum neural networks on near term processors (2018). arXiv preprint arXiv:1802.06002

60. Jäger, J., Simon, M., Denzler, J., Wolff, V., Fricke-Neuderth, K., Kruschel, C.: Croatian fish dataset: Fine-grained classification of fish species in their natural habitat. Swansea Bmvc **2**, 6.1-6.7 (2015)
61. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft (2013). arXiv preprint arXiv:1306.5151
62. Al-Dhabyani, W., Gomaa, M., Khaled, H., Fahmy, A.: Dataset of breast ultrasound images. Data Brief **28**, 104863 (2020)
63. Cortinhas, S.: Apples or tomatoes. Online available at: https://www.kaggle.com/datasets/samuelcortinhas/apples-or-tomatoes-image-classification
64. Bloch, F.: Nuclear induction. Phys. Rev. **70**(7–8), 460 (1946)
65. Weigold, M., Barzen, J., Leymann, F., Salm, M.: Data encoding patterns for quantum computing. In: HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. 22 (2019)
66. Havlíček, V., Córcoles, A.D., Temme, K., Harrow, A.W., Kandala, A., Chow, J.M., Gambetta, J.M.: Supervised learning with quantum-enhanced feature spaces. Nature **567**(7747), 209–212 (2019)
67. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications (2017). arXiv preprint arXiv:1704.04861
68. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)
69. McClean, J.R., Boixo, S., Smelyanskiy, V.N., Babbush, R., Neven, H.: Barren plateaus in quantum neural network training landscapes. Nat. Commun. **9**(1), 4812 (2018)
70. Cerezo, M., Sone, A., Volkoff, T., Cincio, L., Coles, P.J.: Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat. Commun. **12**(1), 1791 (2021)
71. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., Killoran, N.: Evaluating analytic gradients on quantum hardware. Phys. Rev. A **99**(3), 032331 (2019)
72. Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Alam, M.S., Ahmed, S., Arrazola, J.M., Blank, C., Delgado, A., Jahangiri, S., et al.: Pennylane: Automatic differentiation of hybrid quantum-classical computations (2018). arXiv preprint arXiv:1811.04968
73. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Adv. Neural. Inf. Process. Syst. **32**, 8026–8037 (2019)
74. Linnainmaa, S.: Taylor expansion of the accumulated rounding error. BIT Numer. Math. **16**(2), 146–160 (1976)
75. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Representations by Back-propagating Errors. Nature **323**(6088), 533–536 (1986). https://doi.org/10.1038/323533a0
76. Jones, T., Gacon, J.: Efficient calculation of gradients in classical simulations of variational quantum algorithms (2020). arXiv preprint arXiv:2009.02823
77. Suzuki, Y., Kawase, Y., Masumura, Y., Hiraga, Y., Nakadai, M., Chen, J., Nakanishi, K.M., Mitarai, K., Imai, R., Tamiya, S., et al.: Qulacs: a fast and versatile quantum circuit simulator for research purpose. Quantum **5**, 559 (2021)
78. Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N.: Circuit-centric quantum classifiers. Phys. Rev. A **101**(3), 032308 (2020)
79. Bowles, J., Ahmed, S., Schuld, M.: Better than classical? the subtle art of benchmarking quantum machine learning models (2024). arXiv preprint arXiv:2403.07059
80. Kordzanganeh, M., Buchberger, M., Kyriacou, B., Povolotskii, M., Fischer, W., Kurkin, A., Somogyi, W., Sagingalieva, A., Pflitsch, M., Melnikov, A.: Benchmarking simulated and physical quantum processing units using quantum and hybrid algorithms. Adv. Quant. Technol. **6**(8), 2300043 (2023)
81. Jamadagni, A., Läuchli, A.M., Hempel, C.: Benchmarking quantum computer simulation software packages (2024). arXiv preprint arXiv:2401.09076
82. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization 3rd international conference on learning representations. In: ICLR 2015-Conference Track Proceedings, vol. 1 (2015)
83. Bengio, Y., Goodfellow, I., Courville, A.: Deep Learning, vol. 1. MIT press Cambridge, MA, USA (2017)
84. Neelakantan, A., Vilnis, L., Le, Q.V., Sutskever, I., Kaiser, L., Kurach, K., Martens, J.: Adding gradient noise improves learning for very deep networks (2020). arxiv 2015. arXiv preprint arXiv:1511.06807

85. Zhou, M., Liu, T., Li, Y., Lin, D., Zhou, E., Zhao, T.: Toward understanding the importance of noise in training neural networks. In: International Conference on Machine Learning, pp. 7594–7602. PMLR (2019)
86. Pellow-Jarman, A., Sinayskiy, I., Pillay, A., Petruccione, F.: A comparison of various classical optimizers for a variational quantum linear solver. Quantum Inf. Process. **20**(6), 202 (2021)
87. Nakaji, K., Uno, S., Suzuki, Y., Raymond, R., Onodera, T., Tanaka, T., Tezuka, H., Mitsuda, N., Yamamoto, N.: Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators. Phys. Rev. Res. **4**(2), 023136 (2022)
88. Mitsuda, N., Ichimura, T., Nakaji, K., Suzuki, Y., Tanaka, T., Raymond, R., Tezuka, H., Onodera, T., Yamamoto, N.: Approximate complex amplitude encoding algorithm and its application to data classification problems. Phys. Rev. A **109**(5), 052423 (2024)
89. Usman, M., West, M.T., Nakhl, A.C., Heredge, J., Creevey, F.M., Hollenberg, L.C., Sevior, M.: Drastic circuit depth reductions with preserved adversarial robustness by approximate encoding for quantum machine learning. Intelli. Comput. **3**, 100 (2024)