

Parameter-efficient quantum anomaly detection method on a superconducting quantum processor

Maida Wang¹, Jinyang Jiang², and Peter V. Coveney^{1,3,4,*}

¹Centre for Computational Science, Department of Chemistry, *University College London*, London, United Kingdom

²Guanghua School of Management, *Peking University*, Beijing, China

³Advanced Research Computing Centre, *University College London*, London, United Kingdom

⁴Institute for Informatics, *University of Amsterdam*, Amsterdam, The Netherlands



(Received 31 March 2025; accepted 22 September 2025; published 23 October 2025)

Quantum machine learning has gained attention for its potential to address computational challenges. However, whether those algorithms can effectively solve practical problems and outperform their classical counterparts, especially on current quantum hardware, remains a critical question. In this work, we propose a quantum machine learning method, called Parameter-Efficient Quantum Anomaly Detection (PEQAD), for practical image anomaly detection, which aims to achieve both parameter efficiency and superior accuracy compared to classical models. Emulation results indicate that PEQAD demonstrates favorable recognition capabilities compared to classical baselines, achieving an average accuracy of over 90% on benchmarks with significantly fewer trainable parameters. Theoretical analysis suggests that PEQAD has a comparable expressivity to classical counterparts while requiring only a fraction of the parameters. Furthermore, we demonstrate the first implementation of a quantum anomaly detection method for general image datasets on a superconducting quantum processor. Specifically, we achieve an accuracy of over 80% with only 16 parameters on the device, providing initial evidence of PEQAD's practical viability and highlighting its significant reduction in parameter requirements.

DOI: [10.1103/cv9y-2cnj](https://doi.org/10.1103/cv9y-2cnj)

I. INTRODUCTION

In recent years, research on quantum computing has seen rapid development across multiple areas [1–3]. For instance, research teams like that at the University of Science and Technology of China have demonstrated quantum advantage in boson sampling problems [4]. These achievements mark the advent of the Noisy Intermediate-Scale Quantum (NISQ) era [5], where quantum computing systems can integrate dozens of noisy qubits. Despite the influence of noise, which limits these systems' capability for fault-tolerant universal quantum computation, they have demonstrated the potential to achieve performance beyond that of conventional computing in specific tasks. This has led current research to focus on leveraging quantum algorithms to achieve practical quantum advantage, with quantum machine learning being one of the most promising candidates and an active research hotspot.

Quantum machine learning (QML) aims to leverage quantum computing principles to enhance classical machine learning algorithms and address complex computational problems [6]. Recent advancements in quantum algorithms and hardware have enabled the exploration of quantum-enhanced

models for a wide range of applications, such as quantum autoencoders for data compression, quantum Boltzmann machines for generative modeling, and quantum kernel methods for pattern recognition [7,8].

However, applying QML to datasets with complex and diverse characteristics remains challenging [9], particularly in practical scenarios involving noisy and limited quantum devices. Therefore, there is an urgent need for more examples demonstrating the practical feasibility of QML on widely recognized classical datasets, especially given the limitations of quantum hardware, such as the small number of available qubits. While QML shows potential, more work is needed to explore its capabilities within realistic constraints, using well-known datasets and conducting practical implementations on contemporary quantum devices [10].

To date, many QML studies have focused on standard machine learning tasks, such as binary or multiclass classification. However, this paper aims to investigate the feasibility and efficacy of QML in anomaly detection. Fundamentally differing from binary classification that learns a boundary between two known classes, anomaly detection is typically framed as a one-class learning problem where the model is trained almost exclusively on “normal” data. The objective is to learn a compact description of the normal class, such that any data that deviate significantly from this learned description are identified as an anomaly. Anomaly detection is a crucial field in machine learning, largely because many real-world problems lack the balanced, two-class datasets required for standard supervised classification. In critical applications such as financial fraud prevention, system intrusion detection,

*Contact author: p.v.coveney@ucl.ac.uk

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

or industrial machinery monitoring, “normal” operational data are abundant, while confirmed “anomalous” events are, by nature, extremely rare. This scarcity often makes it impossible to form a viable training set for the abnormal class, yet the timely detection of these outliers is of paramount importance. This motivates a distinct one-class learning approach, focused on identifying any deviation from a well-characterized norm, rather than distinguishing between two well-known classes [11–14]. This importance has motivated the development of numerous effective anomaly detection algorithms [15–18]. While various models have been proposed for anomaly detection on conventional computing platforms, the application of quantum machine learning to this task remains relatively unexplored. Only a handful of recent models attempt to use QML for anomaly detection [19,20], and existing algorithms have often struggled to achieve the performance levels of conventional machine learning methods, particularly on practical datasets [10]. Indeed, it has been demonstrated that a number of prevalent quantum models, as documented in the literature, exhibit comparable or inferior performance to classical machine learning on both classical and quantum datasets, even when utilizing emulators [8]. This highlights the challenge of developing a QML algorithm that consistently outperforms classical methods for practical tasks.

Recent works, such as the kernel-based methods applied to high-energy physics data [21], have explored the potential of QML in this domain. Their method utilizes a quantum feature map to compute a kernel matrix, which is then used to train a classical Support Vector Machine for classification. Our approach with Parameter-Efficient Quantum Anomaly Detection (PEQAD) represents a different, end-to-end variational philosophy. The quantum circuit itself contains trainable parameters, θ , which are optimized directly for the one-class learning objective. This contrasts with the kernel-based methodology, where the quantum circuit provides a feature map for a separate classical optimization algorithm. Thus, while both are powerful hybrid approaches, PEQAD focuses on learning an adaptive, task-specific quantum embedding, whereas the kernel method focuses on leveraging a quantum-generated feature space for a robust classical classifier.

Specifically, we concentrate on three key aspects: First, we aim to develop a quantum machine learning method for anomaly detection, striving to achieve competitive or even superior performance compared to classical methods in accuracy. Second, we explore the expressivity of PEQAD and conclude that PEQAD may have a larger learning space even with significantly fewer parameters. Thus, we intentionally constrained the number of learnable parameters in PEQAD to a mere fraction, often just a few percent, of those used in typical classical deep learning models to investigate the potential parameter efficiency better. We also investigate the performance of PEQAD in scenarios with limited training data, as many real-world anomaly detection applications do not have large amounts of labeled data, requiring models that can learn effectively from small datasets.

Finally, we implement PEQAD on quantum hardware, making it, to the best of our knowledge, the first end-to-end quantum anomaly detection algorithm applied to general image datasets implemented on a quantum device. Unlike prior

studies that employed established methods and focused on domain-specific datasets, such as those from high-energy physics, our study demonstrates the versatility and scalability of PEQAD for general image datasets, with an emphasis on parameter efficiency. This experimental implementation serves multiple purposes: it demonstrates the compatibility of PEQAD on real hardware architecture and highlights its potential parameter efficiency in real quantum environments.

Our paper is organized as follows. In Sec. II, we provide an overview of related work, including recent developments in quantum machine learning, anomaly detection, and quantum anomaly detection approaches. Section III covers the preliminaries of quantum computing, introducing the basics of variational quantum circuits. In Sec. IV, we describe the structure of our proposed quantum machine learning algorithm, PEQAD, including its training process and data encoding strategy. Section V presents a theoretical analysis of PEQAD’s expressivity, comparing it to classical deep neural networks. In Sec. VI, we report its noiseless emulation and implementation on a superconducting quantum hardware platform. Finally, Sec. VII concludes the paper with a summary and discussion of future research directions.

II. RELATED WORK

Quantum machine learning. QML has undergone rapid development, driven by advancements in quantum algorithms and hardware, allowing for the exploration of quantum-enhanced models across various applications. For instance, quantum autoencoders have been used for data compression and noise reduction in quantum systems, demonstrating the utility of QML in efficiently encoding quantum data with minimal qubit usage [22–24]. Quantum Boltzmann machines have been proposed for quantum-state tomography and generative modeling, where they can learn complex quantum distributions [25,26]. Quantum generative adversarial learning has shown promise in learning and generating quantum states, with some successful applications in emulating quantum systems and verifying entanglement properties [7,27–29]. Additionally, quantum kernel methods have been explored for pattern recognition tasks, leveraging quantum computing to improve the representational power of kernels for supervised learning [8,30,31].

While QML has shown considerable potential in addressing challenges involving quantum data, such as improving the efficiency of learning quantum systems or emulating complex quantum states [32–34], its practical application to datasets with complex and diverse characteristics remains challenging [9]. For instance, a recent experiment conducted by a collaboration between Caltech and Google demonstrated that a 40-qubit quantum computer could learn from significantly fewer samples compared to a conventional computer [32]. Although this experiment was carefully controlled and not a practical case, it provides an encouraging glimpse into the potential of quantum machine learning. Therefore, it remains essential to present more compelling and practical examples that showcase the strong performance of quantum machine learning on widely recognized and classical tasks.

Anomaly detection addresses the unique challenges of one-class learning, where the goal is not to separate two known

data classes but rather to identify anomalous patterns or outliers; this problem has applications in monitoring machinery, detecting system intrusions, identifying financial fraud, and aiding medical diagnoses [11–14]. Traditional anomaly detection algorithms include Parzen’s density estimate [35], One-Class Support Vector Machine [36], and Support Vector Data Description (SVDD) [37]. With the progress of deep learning, deep AD algorithms such as AutoEncoders [15], Deep SVDD (DSVDD) [16], Deep Convolutional AutoEncoder (DCAE) [17], and one-class generative adversarial network (OC-GAN) [18] have emerged, leveraging advanced representation learning techniques.

Quantum anomaly detection. Applying QML to anomaly detection tasks remains relatively unexplored. Only a few recent models, such as variational quantum one-class classifier (VQOCC) [19] and a kernel-based quantum anomaly detection method applied to the latent space of proton collision events at the large hadron collider (LHC) [20], have been proposed to utilize QML for anomaly detection. The former does not explore hardware implementation, whereas the latter achieves implementation but uses a quantum kernel method tailored to high-energy physics datasets, with 300-dimensional input reduced to 4, 8, or 16 dimensions via autoencoders. In contrast, our work introduces PEQAD, a QML algorithm designed explicitly for anomaly detection on NISQ hardware, targeting general image datasets with original input sizes of 784 dimensions, reduced to 16 dimensions for hardware implementation. By emphasizing parameter efficiency and providing theoretical analysis, our study highlights the versatility of PEQAD in addressing general image anomaly detection tasks, making it more readily extendable to other tasks and larger scales. In numerical emulations, PEQAD consistently outperforms classical and existing quantum anomaly detection methods on benchmark datasets with fewer parameters. Unlike previous works that mainly focus on emulation, we extended our study to real quantum hardware, where PEQAD achieved comparable performance with significantly fewer parameters. This aligns with our theoretical analysis, which suggests that PEQAD possesses a larger expressive capacity, even with only a fraction of the parameters used in classical models. Our hardware experiments further demonstrate the feasibility and robustness of PEQAD, showing its effectiveness in noisy quantum environments while maintaining low parameter requirements. These features make PEQAD particularly suited for small-scale quantum hardware and for addressing complex, high-dimensional anomaly detection tasks.

Our motivation for applying a quantum approach to anomaly detection is rooted in the task’s central challenge: constructing a compact boundary around “normal” data in a feature space. We propose that parametrized quantum circuits are particularly suited for this purpose. By mapping data into a vast Hilbert space, these circuits can generate rich feature representations ideal for defining a minimal hypersphere. We hypothesize that through quantum superposition and entanglement, such circuits can identify complex non-linear data correlations with significantly fewer parameters than classical counterparts. This parameter efficiency could enable the formation of a hypersphere with a sharper decision

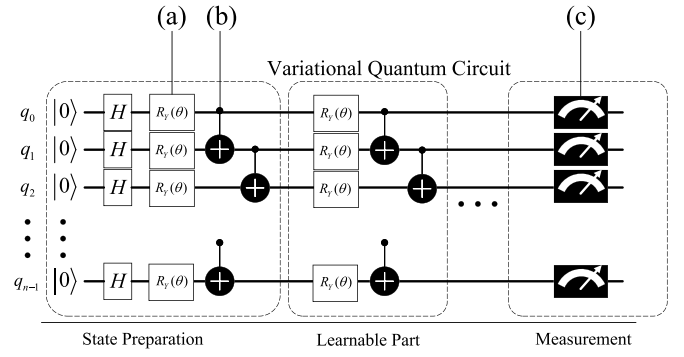


FIG. 1. Diagram of the variational quantum circuit. (a) The first set of parameter-containing gates is used to embed classical features into the quantum circuit, and the rest of them are trainable gates. (b) CNOT gates for entanglement. (c) The measurement part maps quantum states back to classical vectors.

boundary, enhancing the model’s sensitivity to subtle, unseen anomalies.

III. QUANTUM COMPUTING

Quantum computing operates on principles rooted in quantum mechanics, utilizing quantum bits, or qubits, as the fundamental units of information. These qubits exist in a superposition of two basis states, commonly referred to as $|0\rangle$ and $|1\rangle$. A qubit state ψ can be expressed as $\psi = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex amplitudes satisfying $|\alpha|^2 + |\beta|^2 = 1$. Measurement of a qubit causes the collapse of its superposition, with $|\alpha|^2$ and $|\beta|^2$ representing the probabilities of observing $|0\rangle$ and $|1\rangle$, respectively.

Extending this concept to n qubits results in a composite state $|s\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$, where $|i\rangle$ denotes the i th computational basis states for n qubits. The complex amplitudes $\{\alpha_i\}$ satisfy the normalization condition $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$, representing the probability distribution of the state. Quantum circuits utilize quantum gates to manipulate qubits. As illustrated in Fig. 1, a variational quantum circuit (VQC) typically comprises several quantum gates, some of which are parametrized. Quantum gates operate as unitary transformations. Thus, a general parametric single-qubit quantum gate can be represented by a unitary matrix as follows:

$$U(\theta) = \begin{bmatrix} e^{i(\alpha+\beta)} \cos\left(\frac{\gamma}{2}\right) & -e^{i(\alpha-\beta)} \sin\left(\frac{\gamma}{2}\right) \\ e^{i(\alpha-\beta)} \sin\left(\frac{\gamma}{2}\right) & e^{i(\alpha+\beta)} \cos\left(\frac{\gamma}{2}\right) \end{bmatrix}, \quad (1)$$

where $\theta = (\alpha, \beta, \gamma)$ denotes parameters of the gate. These gates, represented as matrices, form the building blocks of quantum circuits. They manipulate qubits to perform operations and computations within the quantum computing framework. Those quantum gates transform one quantum state $|x\rangle$ into another output state $|y\rangle$ as denoted by $|y\rangle = U(\theta)|x\rangle$, where quantum gate parameters control the functional relationship. Increasing the number of trainable gates allows for more complicated models. Certain quantum gates, such as the CNOT gate, enable qubit entanglement, establishing dependencies between qubit states. When a CNOT gate is applied, the state of the “target” qubit $|t\rangle$ is flipped based on the state

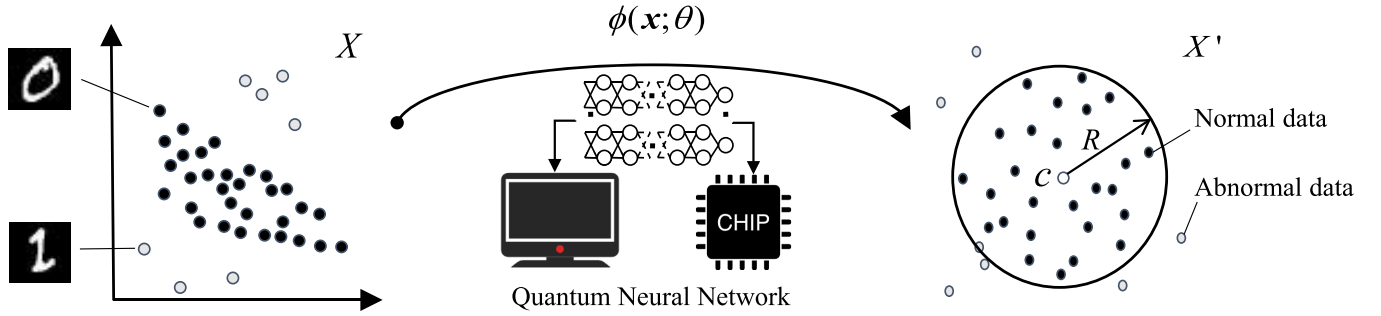


FIG. 2. Conceptual diagram of PEQAD, which is a quantum-based anomaly detection algorithm that adopts QNN to map “normal data” into the hypersphere. Data points residing within the interior of the hypersphere will be labeled as “normal data,” whereas those outside the hypersphere will be considered “abnormal data.”

of the “control” qubit $|c\rangle$, i.e.,

$$\text{CNOT} |c, t\rangle = |c, t \oplus c\rangle, \quad (2)$$

where $\oplus$ denotes the XOR operation. If the control qubit is in state $|1\rangle$, the target qubit is flipped; if it is $|0\rangle$, the target remains unchanged. After executing the circuit, qubits are measured, yielding output probabilities for specific states. In VQC, this measurement process converts quantum states back into classical data.

The combination of trainable quantum gates and fixed ones is widely adopted in VQC, as well as in variational quantum eigensolvers, due to the convenience it offers in optimizing such variational structures. This variational approach, involving a mix of trainable and fixed components, has become a standard choice in many quantum neural networks (QNNs), allowing them to extract complex data features and solve a wide range of problems [8,22–24]. Nevertheless, VQC is not without its limitations. Chief among these is the fact that the optimization scale and depth are constrained by what is known as the barren plateaus (BP) problem, which refers to the exponentially vanishing gradients in variational quantum circuits, making optimization intractable [38]. However, as will be demonstrated in the following sections, our method is not subject to the aforementioned issue in real physical experiments.

IV. METHODS

This section details the design, architecture, and training methodology of our PEQAD, a QML algorithm specified for anomaly detection tasks.

A. Modeling

The main idea of PEQAD is illustrated by the conceptual diagram depicted in Fig. 2. To be specific, PEQAD utilizes a QNN to transform raw data into a feature space, where the data points are supposed to be mapped into a hypersphere. The primary purpose of finding a hypersphere is to encompass a maximal amount of normal data while occupying a minimal volume. During the training stage of anomaly detection, normal data refer to the one class of data selected, while anomalies are considered unknown. Subsequently, the training process culminates in the acquisition of a learned hypersphere. During the testing process, data points whose QNN

output falls outside the learned hypersphere are classified as anomalies, while those inside are considered normal.

Let us denote the given classical training dataset as $X \subseteq \mathbb{R}^d$. $\phi(\cdot; \Theta)$ is a QNN that encodes the real-valued vector $x \in X$ into a corresponding quantum state $|x\rangle$ and then maps it back into the classical embedding set $X' \subseteq \mathbb{R}^p$ in the feature space. The encoding process will be discussed in the next subsection. The QNN consists of a series of quantum gates, and some of them contain trainable parameters $\theta = \{\theta^1, \theta^2, \dots, \theta^m\}$. Thus, $\phi(x; \theta) \in X'$ is the mapped representation of $x \in X$ given by the network structure ϕ with parameters θ . The hypersphere in the feature space is characterized by a center $c \in X'$ and radius post-training $R > 0$, which can be acquired in two ways during training [16]. In our case, the radius is calculated as the mean distance between the center of the hypersphere and the mapped points of the training dataset in the trained network’s feature space.

For the training dataset $X = \{x_1, x_2, \dots, x_n\}$, the training of the QNN embedding model as well as the optimal hypersphere radius searching can be described as the following optimization:

$$\min_{R, \theta} \left\{ R^2 + \frac{1}{vn} \sum_{i=1}^n \max\{0, \|\phi(x_i; \theta) - c\|^2 - R^2\} + \frac{\alpha}{2} \sum_{j=1}^m \|\theta^j\|_F^2 \right\}, \quad (3)$$

where $c = \frac{1}{n} \sum_{i=1}^n \phi(x_i; \theta)$, the second term is the penalty for points falling outside the hypersphere boundary, and the last term is the regularization term to prevent overfitting on the training data. The trade-off between the sphere volume and violations of the boundary constraint is controlled by $v \in (0, 1]$, while $\alpha > 0$ is the regularization coefficient.

To solve Eq. (3) more effectively, a simplified formulation of PEQAD training can be defined in a manner analogous to that employed in Ref. [16]:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n \|\phi(x_i; \theta) - c\|^2 + \frac{\alpha}{2} \sum_{j=1}^m \|\theta^j\|_F^2, \quad (4)$$

where PEQAD shrinks the hypersphere by minimizing the collective distance of all data representations to the center c . Equation (4) pushes the mapped data closer to the center point, compelling the QNN to uncover shared patterns among

elements in the dataset. Penalizing the overall distance across all data points, rather than excluding any outliers, supports the assumption that the bulk of the training data originates from one class. While testing, we define the anomaly score of the input data $\mathbf{x} \in X$ according to its calculated distance from the center of the hypersphere, i.e.,

$$S(X) = \mathbf{1}\{\|\phi(\mathbf{x}; \theta^*) - \mathbf{c}\|^2 - R^2 \geq a\}, \quad (5)$$

where $R = \frac{1}{n} \sum_{i=1}^n \|\phi(\mathbf{x}_i; \theta) - \mathbf{c}\|$, $\mathbf{1}\{\cdot\}$ denotes the indicator function and $a > 0$ is an adjustable threshold. When $S(X) = 0$, PEQAD recognizes the test data as normal; otherwise, it is considered abnormal data. Next, we continue to describe the execution steps of PEQAD.

B. Data encoding and *ansätze*

To handle classical data, a QML algorithm needs a procedure that encodes them into quantum states [39–41]. A straightforward way is to encode the data into the amplitudes of a state, i.e.,

$$\mathbf{x} = (x_1, x_2, \dots, x_N)^\top \Rightarrow |\mathbf{x}\rangle = \frac{1}{\|\mathbf{x}\|} \sum_{i=1}^N x_i |i\rangle, \quad (6)$$

where $N = 2^n$ and n is the least number of the qubits required. This can be implemented on various quantum computing platforms [42,43] and is predominantly performed on emulators. The reason is that implementing amplitude encoding on actual quantum hardware requires a number of quantum gates that scale quadratically with the number of qubits, making it significantly more challenging for systems with a large number of qubits. To date, as far as we are aware, no successful implementation of this encoding method on hardware has been reported. However, because our algorithm requires fewer qubits, it becomes feasible to implement this process on hardware without introducing excessive noise. With the amplitude encoding, we can greatly reduce the qubits used for data encoding to a logarithmic number $n = \lceil \log_2(D_c) \rceil$, where D_c represents the dimensionality of classical data.

The preparation of a quantum state via amplitude encoding is known to be generally intractable for an *arbitrary* classical vector, often requiring a gate complexity that scales exponentially with the number of qubits, n . However, this worst-case scenario does not typically apply to data with significant structural priors, such as the natural images considered in this work. We posit that such data occupy a low-entanglement corner of the Hilbert space, meaning they can be faithfully represented by a Matrix Product State (MPS) with a small bond dimension, χ [44]. Specifically, any n -qubit quantum state $|\psi\rangle$ can be expressed in an MPS form by decomposing its amplitude tensor into a product of smaller, site-specific tensors:

$$|\psi\rangle = \sum_{i_1, \dots, i_n=0}^1 \text{Tr}(A_{i_1}^{[1]} A_{i_2}^{[2]} \dots A_{i_n}^{[n]}) |i_1, i_2, \dots, i_n\rangle, \quad (7)$$

where each $A_{i_k}^{[k]}$ is a complex matrix of size at most $\chi \times \chi$ for the k th qubit, with physical index $i_k \in \{0, 1\}$. The bond dimension χ dictates the amount of entanglement the MPS can represent; a small χ corresponds to a low-entanglement

state that can be prepared and simulated efficiently. A key consequence is that these MPS-representable states can be prepared by quantum circuits with a gate count that scales polynomially—often linearly—in n . In our implementation, we leveraged this very principle. The state preparation circuits were compiled using an efficient synthesis algorithm that implicitly exploits this low-entanglement structure. We empirically observed that this approach yielded circuits of manageable depth for the datasets and qubit counts used, providing practical evidence for the low-entanglement hypothesis within our experimental regime. Therefore, the scalability of our methodology is not merely a future hope but is supported by our implementation choices. The primary limitation of the current study is the small qubit count of the available hardware, which restricts our ability to empirically verify this favorable polynomial scaling over a larger range of n . In addition to addressing the scalability of amplitude encoding, we also evaluated alternative encoding schemes to justify our selection. Specifically, we investigated the performance of angle encoding (referred to as Rotation Encoding in our experiments). However, as our results in Table II demonstrate, this approach led to a significant degradation in accuracy on both the emulator and the hardware. We hypothesize that this is due to the reduced ability of angle encoding to capture the high-dimensional correlations present in the image data, which reinforces our choice of the more expressive amplitude encoding for this proof-of-concept study.

Following the quantum data encoding, a VQC is employed for optimization. Specifically, we adopt the *ansätze* of the VQC shown in Fig. 3, which is derived from Ref. [45]. The *ansätze* comprise layers with parametrized single-qubit rotations $R_Y(\theta) = e^{-i\theta Y/2}$, followed by CNOT gates. These CNOT gates are arranged alternately between adjacent qubits. By employing these *ansätze*, we can reduce the required hardware scale, which has several positive impacts. First, it helps facilitate the optimization process. Second, it helps us use fewer learnable parameters, as the number of quantum gates required to train scales linearly with increased qubits when assuming a fixed number of quantum circuit layers. Following the execution of the variational circuit, we perform a computational basis measurement on each qubit (i.e., a projection onto the Z basis).

C. Training process

Pretraining. As depicted in Fig. 3(a), the pretraining stage employs an encoder-decoder structure. The model for pretraining consists of a quantum encoding layer and a classical decoding layer, where the quantum one is synchronized with the PEQAD's network. During pretraining, the encoder maps the input data $\mathbf{x}$ to an embedding $\mathbf{x}'$, while the decoder aims to recover the original $\mathbf{x}$ from $\mathbf{x}'$. The parameters of the encoder are adjusted to minimize reconstruction errors by parameter shift rules [46]. The center of the hypersphere $\mathbf{c}$ is also initialized by averaging all the embeddings in X' . Both model parameters and a preliminary hypersphere center are warmed up in this phase, accelerating the subsequent training stage.

Training. From this step onward, the PEQAD neural network is used for training and testing. Its structural details are illustrated in Fig. 3(b). The network comprises an amplitude

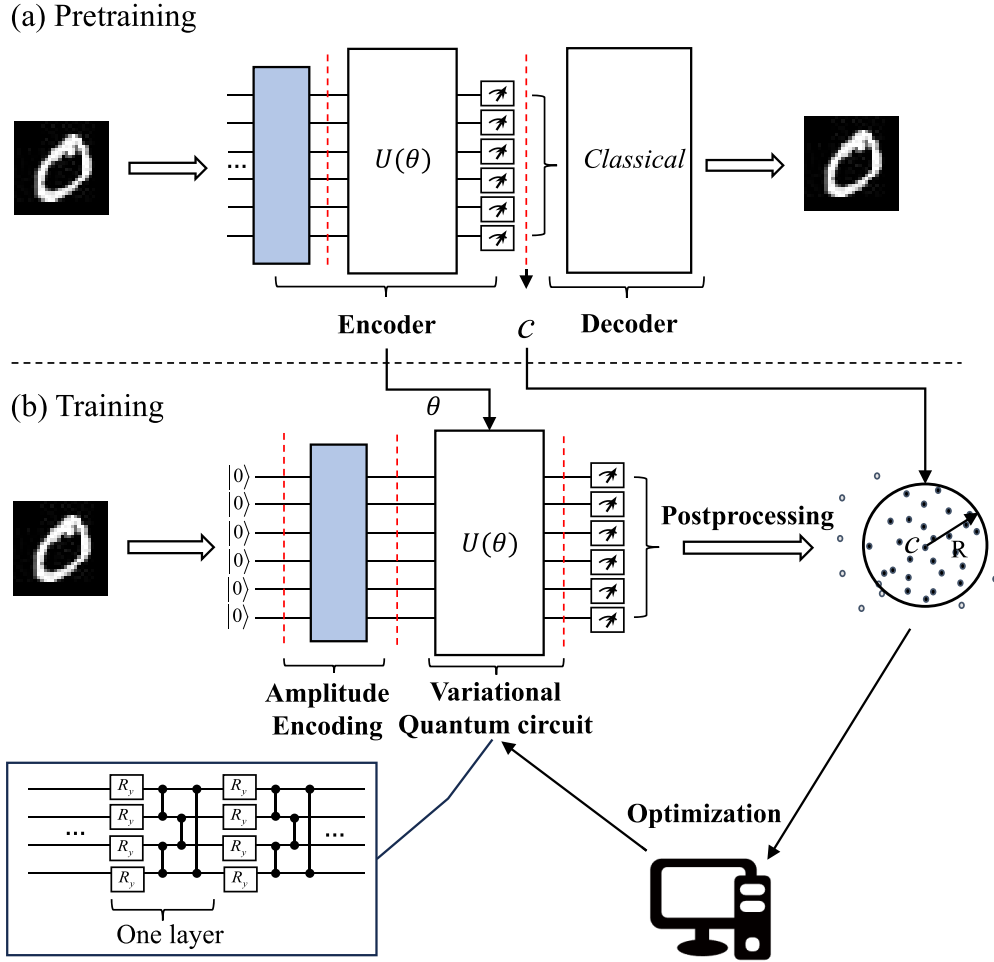


FIG. 3. Training process of the PEQAD algorithm. (a) The pretraining neural network of the method consists of two parts: a quantum encoding layer and a classical decoding layer, where the quantum encoding layer is consistent with the PEQAD network. After the quantum encoding layer, the center of the hypersphere mapped by PEQAD can be obtained. (b) The PEQAD neural network is composed of an amplitude encoding part (blue part), a VQC part, a measurement part, and a PEQAD postprocessing part.

encoding part, a VQC part, a measurement part (represented by the dashboard), and a PEQAD postprocessing part. The training stage mainly involves fine-tuning the PEQAD network. The optimization defined in Eq. (4) is utilized to minimize the volume of the hypersphere. Meanwhile, the hypersphere effectively encapsulates normal data patterns by reducing the average distance of all embeddings to the center. The ratio of the time consumption between pretraining and training is found to be approximately 1.025:1.

V. CHARACTERIZATION OF PEQAD'S EXPRESSIVITY

To provide a theoretical basis and gain a better understanding of its performance, we analyze the expressivity of the PEQAD model. Expressivity is a crucial concept in quantum algorithms and has been extensively studied to understand the underlying theoretical foundations of effective machine learning models. Recent studies (e.g., [47,48]) also focus on the expressivity of variational quantum algorithms (VQAs). Understanding the theoretical expressivity of PEQAD is essential for gaining insights into its capabilities and potential advantages. We aim to reveal how PEQAD's expressivity

influences its performance in tasks that may not conform to independently and identically distributed data rather than deriving generalization error bounds under such conditions. In this section, we leverage the covering number [49], a powerful concept from statistical learning theory, to investigate the expressive power of PEQAD. This analysis examines how the complexity of the hypothesis space influences PEQAD's ability to capture intricate data relationships. We target comparing PEQAD with other classical or quantum anomaly detection algorithms from different perspectives. By integrating empirical results with theoretical insights, we can have a comprehensive assessment of PEQAD's efficacy and its potential applications in QML.

We begin by describing the process of the VQAs, which combine a q -qubit quantum circuit with a classical optimizer. During the training, the trainable parameters of the *ansätze* are optimized iteratively based on the quantum circuit's outputs to minimize the objective function $\mathcal{L}(\cdot, \cdot)$. The parameter update rule at iteration t is given by

$$\theta^{(t+1)} = \theta^{(t)} - \eta \left[\frac{\partial \mathcal{L}(h(\theta^{(t)}), O, \rho), c_1)}{\partial \theta} \right], \quad (8)$$

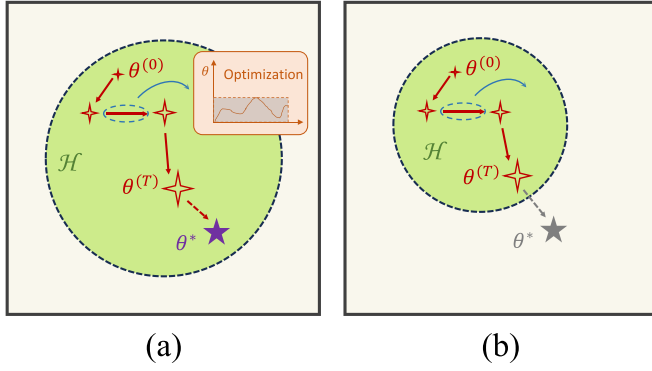


FIG. 4. Expressivity of PEQAD. If $\mathcal{H}$ is of moderate size and effectively encompasses the target concepts (purple solid star), the inferred hypothesis may closely approximate the target concept.

where η is the learning rate, $c_1 \in \mathbb{R}$ is the target label, and $h(\theta^{(t)}, O, \rho)$ is the quantum circuit's output. The gradient information $\partial \mathcal{L}(h(\theta^{(t)}, O, \rho), c_1) / \partial \theta$ can be obtained using the parameter shift rule or other methods.

In the context of anomaly detection, $\rho \in \mathbb{C}^{d^q \times d^q}$ represents the q -qubit input quantum state, $O \in \mathbb{C}^{d^q \times d^q}$ is the quantum observable, and $U(\theta) = \prod_{l=1}^P u_l(\theta) \in \mathbb{U}(d^q)$ denotes the *ansätze*, with $\theta \in \Theta$ being the trainable parameters. We denote the l th quantum gate acting on at most $k \leq q$ qubits as $u_l(\theta) \in \mathbb{U}(d^k)$, and $\mathbb{U}(d^q)$ is the unitary group of dimension d^q . Generally, the *ansätze* consist of P trainable gates, which correspond to the number of trainable parameters, i.e., $\Theta \subseteq [0, 2\pi]^P$. The ideal output of the quantum circuit is given by

$$h(\theta^{(t)}, O, \rho) = \text{Tr}(U(\theta^{(t)})^\dagger O U(\theta^{(t)}) \rho). \quad (9)$$

The definition of $h(\theta^{(t)}, O, \rho)$ can be generic. We now elucidate the connection between expressivity and model complexity in the context of PEQAD. The primary objective of PEQAD is to identify the best hypothesis $h^*(\theta, O, \rho) = \arg \min_{h(\theta, O, \rho) \in \mathcal{H}} \mathcal{L}(h(\theta, O, \rho), c_1)$ that can adeptly approximate the target concept, where $h(\theta, O, \rho)$ and c_1 correspond to $\phi(x; \theta)$ and c in PEQAD modeling. We denote the hypothesis space of PEQAD as $\mathcal{H}$, i.e.,

$$\mathcal{H} = \{(\text{Tr}(U(\theta)^\dagger O U(\theta) \rho) - c)^2 - R^2 \mid \theta \in \Theta\}, \quad (10)$$

where R and c retain the same definitions in previous sections.

The geometric representation of PEQAD's expressivity is illustrated in Fig. 4, where the expressivity of the utilized *ansätze* in PEQAD governs its hypothesis space denoted as $\mathcal{H}$ (solid green ellipse). If $\mathcal{H}$ is of a moderate size—meaning it is large enough to encompass the target concepts effectively but not excessively large—the inferred hypothesis may closely approximate the target concept. Conversely, if $\mathcal{H}$ is too limited in complexity, there will be a substantial discrepancy between the estimated hypothesis and the target concept (gray solid star). Furthermore, if $\mathcal{H}$ encompasses an excessively large space, optimization can become challenging. Both overly high and low expressivities of PEQAD can result in suboptimal performance. Therefore, understanding and controlling the expressivity of PEQAD is a crucial task and requires an effective metric for evaluating the complexity of $\mathcal{H}$. We utilize the covering number to establish an upper bound on the complex-

ity of $\mathcal{H}$, thereby quantifying the expressivity of PEQAD. This approach allows us to rigorously evaluate PEQAD's capacity to model complex distributions and suggest its theoretical capabilities.

Definition 1. The covering number $N(U, \varepsilon, \|\cdot\|)$ represents the minimum cardinality of any subset $V \subset U$ that effectively encompasses U with a scale of ε using the norm $\|\cdot\|$, i.e., $\sup_{A \in U} \min_{B \in V} \|A - B\| \leq \varepsilon$.

The covering number $N(\mathcal{H}, \varepsilon, \|\cdot\|)$ signifies the minimal count of spherical balls, each with a radius of ε , necessary to cover a designated space, allowing for potential overlaps comprehensively. $\varepsilon > 0$ is a predefined small constant, which remains independent of any specific factors [49]. This convention is commonly embraced in the machine learning domain to assess model capacity in diverse learning scenarios [50]. The subsequent theorem provides the upper bound of $N(\mathcal{H}, \varepsilon, \|\cdot\|)$ for PEQAD.

Theorem 1. For $\varepsilon \in (0, \frac{1}{10})$, the covering number of the hypothesis space of PEQAD can be represented as follows:

$$N(\mathcal{H}, \varepsilon, \|\cdot\|) \leq \left(\frac{7P(2\|O\| + 2\|c\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}, \quad (11)$$

where $\mathcal{H}$ denotes the function space represented by PEQAD, $N(\mathcal{H}, \varepsilon, \|\cdot\|)$ represents its covering number, and c is the center of the hypersphere.

Theorem 1 implies that the primary factor influencing the complexity of $\mathcal{H}$ is the quantum gates incorporated into $U(\theta)$. This is evidenced by the fact that the term $q^{2m}P$ contributes to an exponential increase in the complexity $N(\mathcal{H}, \varepsilon, \|\cdot\|)$. In contrast, the operator norm $\|O\|$ and the center $\|c\|$ affect the complexity in a polynomial manner.

Theorem 2. For $\varepsilon \in (0, \frac{1}{10})$, the supremum of the covering number of the hypothesis space of PEQAD is bounded as follows:

$$\left(\frac{7P\|O\|}{\varepsilon} \right)^{q^{2m}P} \leq \sup N(\mathcal{H}, \varepsilon, \|\cdot\|) \quad (12)$$

$$\leq \left(\frac{7P(2\|O\| + 2\|c\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}. \quad (13)$$

Theorem 2 establishes that the supreme expressive capacity of our method is strictly greater than that of normal VQC-based anomaly detection models without postprocessing like VQOCC. The lower bound shows that PEQAD can express at least the same function space as VQOCC, while the upper bound suggests its ability to capture a broader range of patterns due to its more flexible structure. This increased expressivity is crucial for more effective anomaly detection, particularly in complex real-world scenarios. PEQAD's ability to model richer data distributions may offer an advantage over simpler VQC methods.

Theorem 3. For a local depolarizing noise channel with depolarization rate p , the expressivity of PEQAD on noisy quantum devices is bounded by

$$N(\mathcal{H}, \varepsilon, \|\cdot\|) \leq (1-p)^{N_g} \left(\frac{7P(2\|O\| + 2\|c\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}, \quad (14)$$

where N_g is the number of gates affected by noise.

Theorem 3 illustrates how noise, specifically modeled by the factor $(1 - p)^{N_s}$, reduces the expressivity of PEQAD. Here, p represents the noise rate in a depolarizing noise channel and N is the number of quantum gates affected by the noise. As the noise rate p increases (i.e., as p approaches 1), the noise factor $(1 - p)^{N_s}$ approaches zero, which implies a significant reduction in the effective hypothesis space. This shrinking of the hypothesis space limits the PEQAD model's ability to learn and represent complex patterns, particularly in noisy quantum environments. Nevertheless, given that PEQAD encompasses a more expansive expressivity than other VQC anomaly detection algorithms with an equivalent number of parameters, it may retain a greater absolute expressive capacity under the influence of noise. The theorems above establish a theoretical upper bound on PEQAD's expressivity under different circumstances, demonstrating its capability to model functions in high-dimensional spaces. Detailed mathematical foundations supporting these findings are provided in the Appendixes.

Limitation. Since our work mainly considers anomaly detection tasks, the training set typically lacks outliers in general anomaly detection scenarios, while the test set includes outlier data. Consequently, statistical analysis of these sets is limited to analyzing the model's complexity, which is the focus of this subsection. That is, we are unable to exhibit the superiority of PEQAD further using an expression for the generalization error of the same distribution. However, integrating theoretical and experimental results allows us to demonstrate that the theoretical analysis aligns well with the emulation and experimental outcomes. This consistency provides deeper insights into PEQAD's superior expressive capacity, which enables it to effectively capture complex patterns, leading to better performance in practical anomaly detection tasks. The detailed discussion in the Appendixes elaborates on how the enhanced expressivity of PEQAD could further help harness the power of QML to address anomaly detection challenges.

To compare the expressivity of PEQAD with classical algorithms such as DSVDD, we also provide the theoretical covering number bound for classical models. This bound originally comes from Bartlett *et al.*'s work [51], and we further derived its form to focus on the impact of parameters on the model's expressivity. These derivations are presented in the Appendixes, and the resulting bound is as follows:

Theorem 4. For a fixed training set with data norm bounded by a constant t , and for $\varepsilon \in (0, \frac{1}{10})$, the covering number of the hypothesis space of DSVDD and other classical deep networks, denoted as $\mathcal{H}_c$, can be represented as follows:

$$\ln N(\mathcal{H}_c, \varepsilon, \|\cdot\|_2) \leq \frac{t^2(\ln(2) + 2\ln(\max(m_j \times n_j)))}{\varepsilon^2} \quad (15)$$

$$\times \left(\prod_{j=1}^L m_j n_j \|W_j\|_\infty^2 \right) \left(\sum_{i=1}^L \left(\frac{\sqrt{m_i}}{\sqrt{n_i}} \right)^{2/3} \right)^3, \quad (16)$$

where $N(\mathcal{H}_c, \varepsilon, \|\cdot\|_2)$ represents the covering number of hypothesis space $\mathcal{H}_c$ of a classical neural network, m_j and n_j are the dimensions of the weight matrices in each layer, W_j is the weight matrix of the j th layer, and $\|W_j\|_\infty$ represents the maximum absolute value of the elements in W_j . The constant

t is the norm of the data matrix, and L is the number of layers in the network.

An important consideration is that increasing the expressive power of a QML algorithm generally increases the likelihood of encountering the barren plateau problem [38]. To examine this potential issue, we conduct a theoretical analysis to assess whether our algorithm is particularly prone to the barren plateaus. As we demonstrate in our detailed analysis in Appendix F, if PEQAD were implemented with a generic, highly entangling ansatz that approximates a random unitary circuit (a t -design), the barren plateau phenomenon is indeed inevitable. Under this assumption of arbitrary unitaries, the expectation of the loss function's gradient is strictly zero [see Eq. (F40)], and its variance vanishes exponentially with the number of qubits q , scaling as $O(1/2^{3q})$ [see Eq. (F41)]. This makes optimization intractable for larger systems. However, our work employs a specific hardware-efficient, layered ansatz that is a fixed, structured circuit, not an arbitrary unitary. For this class of circuits, the gradient scaling behavior is fundamentally different. Our analysis for this specific ansatz confirms that the barren plateau is mitigated: the gradient expectation is not strictly zero but instead follows the scaling given by Eq. (F47), and the variance decays as described by Eq. (F48). This means that while a decaying trend exists, the gradients remain statistically significant for practical system sizes, ensuring the trainability of our model. This finding underscores the crucial role of codesigning the algorithm and ansatz to maintain trainability in practice, and a complete mathematical derivation of both the general and specific cases is provided in Appendix F.

Based on the results above, we derive the following specific conclusion regarding the expressivity of PEQAD compared to classical models.

Conclusion. PEQAD can achieve comparable or even superior performance using significantly fewer learning parameters, offering a potential advantage for scaling quantum machine learning in practical applications. We validate this observation both theoretically and experimentally. In the experimental section, we show that PEQAD, with only 16 parameters, achieves recognition performance equivalent to that of a classical neural network with approximately 400 parameters. In the theoretical analysis provided in the Appendixes, we provide evidence suggesting that the expressivity of PEQAD, even with 16 parameters on quantum hardware, can reach a similar level to that of a classical neural network with hundreds of parameters. We provide a detailed derivation of this result in Appendix D and E, where we compute the exact gradient statistics under our circuit design.

VI. IMPLEMENTATION AND RESULTS

A. Noiseless emulation results

Considering that the vast majority of QML algorithms, including quantum anomaly detection approaches, are typically evaluated in emulated, noise-free environments, we also provide results in this way to establish a fair comparison. Specifically, we first implement our PEQAD on three widely recognized image datasets: MNIST [52], FashionMNIST [53], and CIFAR-10 [54]. For each dataset, data from

TABLE I. Comparison via emulation results. QHDSVDD refers to the Quantum Hybrid Deep SVDD model developed from Ref. [45].

Dataset	Network	Train size	Q	P	AUC
MNIST	OC-GAN	all	—	>6500	90.57%
	DCAE	all	—	>6500	89.96%
	DSVDD	all	—	>6500	94.80%
	DSVDD	300	—	6500	86.59%
	QHDSVDD	300	16	6500	88.24%
	DSVDD	300	—	200	79.44%
	PEQAD (Ours)	300	10	200	92.26%
	PEQAD (Ours)	all	10	200	92.31%
Fashion MNIST	OC-GAN	all	—	>6500	86.93%
	DCAE	all	—	>6500	90.73%
	DSVDD	all	—	>6500	88.50%
	DSVDD	300	—	6500	88.20%
	QHDSVDD	300	16	6500	88.19%
	DSVDD	300	—	200	81.59%
	PEQAD (ours)	300	10	200	91.35%
	PEQAD (ours)	all	10	200	91.81%
CIFAR-10	OC-GAN	all	—	>6500	65.63%
	LSP-CAE	all	—	>6500	67.63%
	DSVDD	all	—	>6500	64.81%
	DSVDD	300	—	6500	63.95%
	DSVDD	300	—	200	57.25%
	PEQAD (Ours)	300	10	200	64.65%
	PEQAD (Ours)	all	10	200	63.31%

a single class are used as the normal training set, while the other classes are treated as anomalies. Performance is assessed based on ten independently replicated emulations using the Area Under the Curve (AUC) metric, which is standard in anomaly detection [55]. It represents the area under the Receiver Operating Characteristic curve. It serves as a standard metric to assess the model's ability to distinguish between normal and anomalous data points, ranging from 0.5 (random guess) to 1.0 (perfect classification). Given the computational overhead of quantum measurement, we constrain the size of each training set S_M by randomly sampling 300 images from the original class, which contains normal images from class $M \in \{0, \dots, 9\}$, respectively. Emulations are conducted using the PennyLane emulator integrated with a PyTorch-based Adam optimizer [56,57], enabling efficient parameter optimization through autodifferentiation. We use a batch size of 50, a learning rate of 1×10^{-3} , and 150 epochs for Adam to optimize the model. We vary the number of VQC parameters P from 120 to 280 and that of qubits Q from 10 to 16. If not specified, P and Q are set to 200 and 10 as default in the subsequent discussions. We benchmark PEQAD against established anomaly detection methods such as DCAE [17], OC-GAN [17], latent subspace projection convolutional autoencoder (LSP-CAE) [17], DSVDD [16], and QHDSVDD [45] to evaluate its efficacy rigorously. These comparisons aim to uncover PEQAD's advantages over both classical and hybrid quantum approaches, offering insights into its potential applications.

In Table I, we exhibit the overall AUC scores averaged among multiple independent emulations for each method mentioned above on three datasets, where different classes of

data are selected as the normal input each time. We denote the number of trainable parameters in our PEQAD and other classical deep anomaly detection neural networks as P , while Q stands for the number of qubits if applicable.

PEQAD consistently outperforms classical solutions across most evaluations, even when the latter employs significantly more parameters. For instance, on the MNIST dataset, PEQAD achieves an AUC of 92.26% with only 200 parameters and 300 training data, surpassing DSVDD by 5.67%. This relative improvement widens to 12.82% when both methods use an equal number of parameters. Similar trends are observed on FashionMNIST, where PEQAD achieves an AUC of 91.35% with 200 parameters and 300 data, outperforming DSVDD by 6.76%. Furthermore, PEQAD demonstrates superior performance to DSVDD on the FashionMNIST dataset when utilizing the complete training set. The MNIST and Cifar-10 datasets exhibit slightly inferior results compared to DSVDD, which has 6500 parameters. Nevertheless, it still exhibits considerably enhanced performance compared to DSVDD with 200 parameters. Compared to other deep neural networks like DCAE, which utilizes over 6500 parameters to achieve an AUC of 89.96% on MNIST, PEQAD demonstrates superior performance with significantly fewer computational resources. Even under more substantial training conditions, PEQAD maintains higher accuracy levels, showcasing its efficiency in anomaly detection tasks. PEQAD also exhibits competitive performance with much lower consumption on the CIFAR-10 dataset, demonstrating its ability to handle complex data structures. The robust and scalable performance of PEQAD in our emulations indicates a potential for quantum advantage in anomaly detection tasks. However, further investigation and experimental verification are required to demonstrate this advantage conclusively. These emulation results demonstrate PEQAD's viability for practical applications and indicate that it is a perfect candidate for future experiments on NISQ devices, which will be discussed in the next section. To further investigate the effect of different configurations, we performed an ablation study as shown in Fig. 5, which will be detailed in the following discussion.

Depth of VQC. We vary the depth D of the VQC within specified ranges, and Fig. 5(a) illustrates how the performance of PEQAD changes with different values of D . When D ranges from 12 to 24, PEQAD consistently achieves satisfactory AUC scores (above 80%) on both datasets. Optimal results on both MNIST and FashionMNIST are observed at $D = 20$. However, increasing D beyond 20 leads to a decline in AUC, likely due to overfitting caused by excessive trainable parameters. Therefore, we default to setting D to 20 for subsequent analyses.

Different VQC structures. We conduct emulations using various VQC architectures: Real Amplitudes Circuit [58], Bellman Circuit [58], Ladderlike Circuit [59], and Dressed Quantum Circuit (DC) [60]. The results are depicted in Fig. 5(b), demonstrating that varying VQC structures yield different performance outcomes. DC notably outperforms other structures, which is attributed to its alternating arrangement of entanglement gates across adjacent qubits, thereby enhancing the stability of circuit-wide entanglement and resulting in superior performance. Therefore, we use DC as the VQC structure in the emulations if not specifically stated.

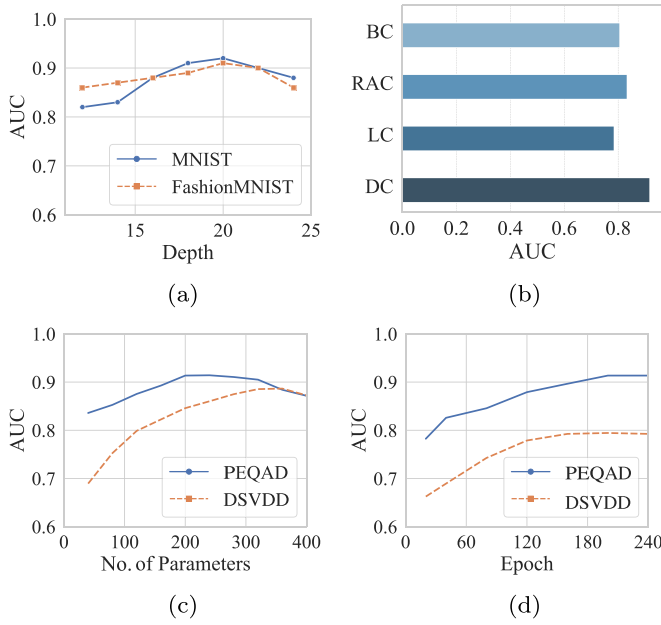


FIG. 5. Ablation study on the effects of VQC's depth, structure, number of parameters, and training epochs. (a) AUC as a function of circuit depth. (b) Performance of different VQC structures. (c) AUC versus the number of trainable parameters for PEQAD and DSVDD. (d) AUC as a function of training epochs. Although the effect of the PEQAD and its classical machine learning counterpart DSVDD varies with setting variation, PEQAD always outperforms DSVDD under the same conditions.

Parameters and epochs. The number of parameters and epochs can influence the performance of our PEQAD solution and other anomaly detection methods. Figure 5(c) demonstrates that PEQAD achieves robust performance on the FashionMNIST dataset when the number of parameters varies from 100 to 400, showing higher stability compared to its classical counterpart, DSVDD. Additionally, Fig. 5(d) illustrates that varying the number of training epochs from

40 to 240 consistently maintains PEQAD's high-performance level. However, beyond 200 epochs, both PEQAD and classical DSVDD experience a decline in AUC, which may be attributed to overfitting. Notably, PEQAD exhibits superior and more stable performance under most settings than its classical counterpart.

PEQAD and other QML anomaly detection methods. As illustrated in Fig. 6, the comparison between PEQAD and another QML anomaly detection method, Variational Quantum One-Class Classifier [19], shows the performance difference within the quantum-based algorithm family on specific anomaly detection tasks. PEQAD consistently achieves results comparable to the variational quantum one-class classifier across most categories and significantly outperforms it in challenging scenarios where the variational quantum one-class classifier fails (e.g., Classes 5 and 6), demonstrating its superior robustness and reliability. Another promising quantum anomaly detection algorithm has recently been published, which uses kernel-based methods [20]. However, as it has only been tested on specific high-energy datasets, a direct comparison with our work is not currently feasible. These results underscore the effectiveness of PEQAD and its suitability for a wider range of anomaly detection tasks.

B. Results on a superconducting quantum processor

For the first time, we implement a quantum anomaly detection method for general image datasets on quantum hardware. To thoroughly evaluate the performance of PEQAD, we conducted experiments on a 12-qubit superconducting processor oneD12 provided by the Institute of Quantum Information and Quantum Technology Innovation, Chinese Academy of Sciences (CAS) [61]. To ensure fairness, we kept all experimental emulation settings, including datasets and training conditions, consistent across the hardware platform and emulators. Specifically, we used only four qubits for the hardware implementation, facilitating the realization of amplitude encoding. The fidelity of the device used in our experiments can be found on the official websites of the Institute of

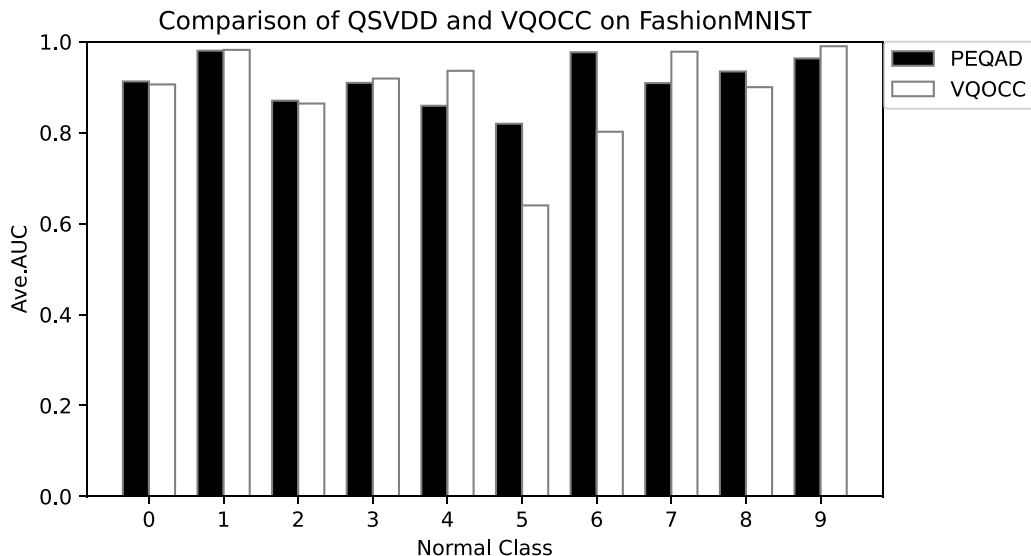


FIG. 6. Comparison of PEQAD and VQOCC.

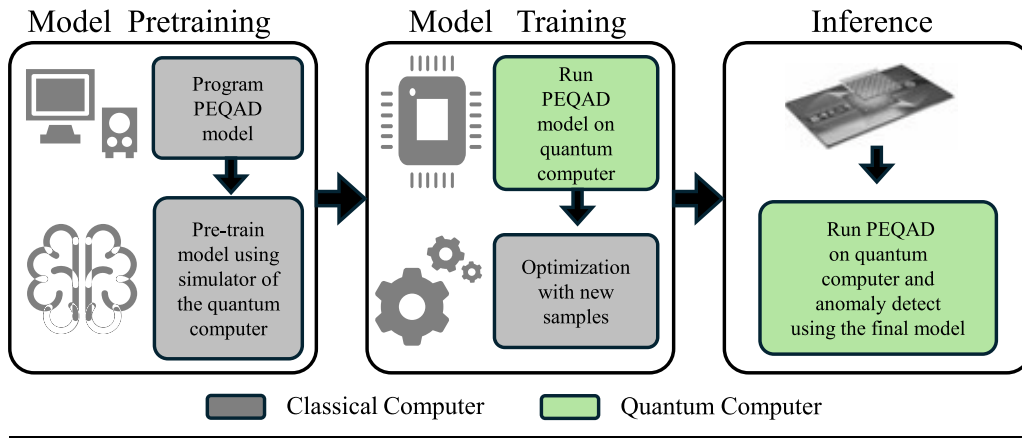


FIG. 7. Practical pipeline to address anomaly detection tasks on a quantum computer.

Quantum Information and Quantum Technology Innovation, CAS. For instance, the single-qubit gate fidelity for state 0 of the oneD12 chip we used averages at 0.9965 (ranging from 0.9946 to 0.9997), while the CZ gate fidelity averages at 0.9634 (with a range from 0.9354 to 0.9885). In order to provide a comprehensive demonstration of the performance and robustness of our algorithm in the presence of noise, and given the limited scale of the superconducting chip employed, we elected not to apply any noise reduction techniques or error mitigation methods to the implementation of our method on the quantum device.

During training on the quantum processor, postprocessing was applied to the measurement data to map learned states onto the hypersphere, following Eq. (4). Once training was completed, the model was evaluated on the test set using Eq. (5) to determine whether the data points were anomalies. The overall experimental pipeline for implementing PEQAD on quantum hardware in the NISQ era is illustrated in Fig. 7.

The implementation of PEQAD on the quantum hardware platform is shown in Fig. 8(a). To perform the VQC, each qubit is first initialized in the ground state, and encoding is applied in the pink region of the quantum processor to encode the input data. In the yellow region, the circuit is parametrized with single-qubit rotations and entangling gates between neighboring qubits to create a trainable variational quantum circuit. After executing the VQC, we apply the preprocessing step of PEQAD by calculating the anomaly detection score based on the output quantum states. The results are then compared to a predetermined threshold to classify the states, completing the implementation of PEQAD on the quantum processor.

We compare the performance of PEQAD across different quantum platforms, including emulators and the real superconducting quantum processor, namely, the VQC emulator, the superconducting emulator, and the actual superconducting quantum processor oneD12. The superconducting emulator is implemented to replicate its respective chip structure accurately. In contrast, the VQC emulator is designed to resemble the general configuration of the hardware setups closely. For the VQC emulation, we employed two different structures: one using a global four-qubit unitary with amplitude embedding (denoted as VQC) and one using rotation embedding

[denoted as VQC (RE)]. VQC and VQC (RE) are designed to mirror the configurations of the superconducting processor oneD12 closely but using different encoding methods. This approach allows us to evaluate the algorithm's adaptability to different quantum hardware structures and ansätze.

All experiments were conducted under consistent conditions to ensure fairness, each using 100 randomly selected normal samples for training and another 100 for testing. All emulators were allocated computational resources equivalent to four qubits. The MNIST and FashionMNIST datasets, originally consisting of 196 dimensions (after normalization and first initialization), were downsampled to 16 using a pooling technique. This reduction was done to facilitate efficient data encoding on the quantum hardware, which is constrained by the number of available qubits and their connectivity. The training was performed over 200–1000 epochs with a learning rate of 1×10^{-2} . For the optimization of PEQAD on the quantum processor, we utilized Adam and simultaneous perturbation stochastic approximation [62]. We continue to use AUC as the evaluation metric in this part.

The experimental and emulation results of PEQAD are summarized in Table II. We observed that the performance of PEQAD on the superconducting processor is generally comparable to that obtained using emulators, including the superconducting emulator and VQC emulator. On the MNIST dataset, PEQAD achieved an accuracy of 81.2% on the superconducting processor, closely matching the 84.8% accuracy obtained with the VQC emulator and surpassing the 75.6% achieved with VQC (RE). On the FashionMNIST dataset, the accuracy reached 83.4% on the superconducting processor, aligning well with the 87.5% from VQC and 82.1% from VQC (RE). Moreover, rotation encoding has been observed to markedly diminish the accuracy of recognition, both on the emulator and on the hardware. This outcome is likely attributable to the markedly insufficient exploitation of Hilbert space by this encoding method. Overall, despite a slight decline in accuracy when implemented on actual quantum hardware compared to emulators, the results from the superconducting processor demonstrated favorable performance, highlighting PEQAD's robustness on the real quantum platform. The observed discrepancies between hardware and emulation results can likely be attributed to the noisy con-

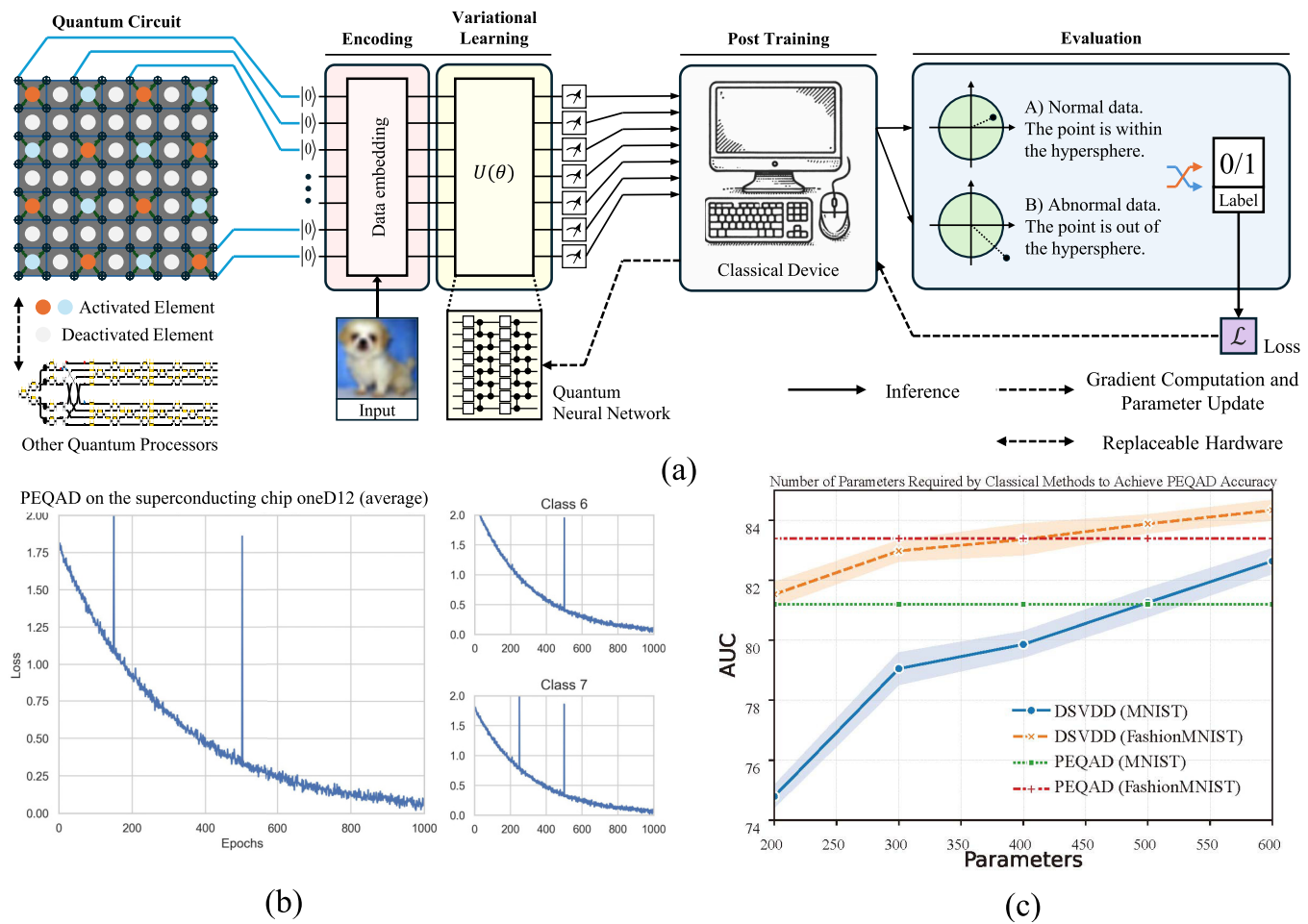


FIG. 8. (a) Conceptual diagram showing the deployment of PEQAD on a quantum processor. (b) The optimization process of PEQAD on the superconducting processor oneD12 shows some fluctuations due to hardware instability. Despite these, the final loss is minimized effectively. (c) The parameter efficiency comparison between PEQAD on the superconducting quantum processor and DSVDD on a conventional computer. Dashed lines represent PEQAD hardware results using 16 parameters on MNIST or FashionMNIST, while solid lines show DSVDD performance as parameters increase. DSVDD requires significantly more parameters to match PEQAD performance.

ditions typical of NISQ devices and the limited number of qubits, which constrains the quantum model’s expressivity.

TABLE II. Results on the superconducting processor and the corresponding emulators.

Dataset	Device	Q	P	AUC
MNIST	VQC	4	16	84.8%
	oneD12 emulator	4	16	83.2%
	oneD12 processor	4	16	81.2%
	VQC (RE)	4	16	75.6%
	oneD12 processor (RE)	4	16	73.4%
	DSVDD	–	500+	81.2%
	DSVDD	–	600+	83.6%
FashionMNIST	VQC	4	16	87.5%
	oneD12 emulator	4	16	85.2%
	oneD12 processor	4	16	83.4%
	VQC (RE)	4	16	82.1%
	oneD12 processor (RE)	4	16	77.4%
	DSVDD	–	300+	83.4%
	DSVDD	–	800+	85.3%

Importantly, PEQAD performed consistently well without any error mitigation techniques, demonstrating its resilience in realistic quantum computing environments.

It is important to note that our experiments indicate that PEQAD can achieve competitive performance with a much smaller number of parameters compared to its classical counterpart, DSVDD. To provide a fair comparison between PEQAD and its conventional counterpart, DSVDD, experiments and emulations were conducted under identical conditions, differing only in the number of model parameters. As illustrated in Fig. 8(c), we progressively augmented the number of parameters in DSVDD, commencing from a modest value until its performance reached a level commensurate with PEQAD. DSVDD necessitates a parameter count that is more than a few tens of times greater than required to achieve the results of our 16-parameter experiment in hardware. The specific parameter numbers are presented in the accompanying table. As shown in Table II, the findings demonstrated that DSVDD necessitated a minimum of 300 parameters to attain comparable recognition accuracy, whereas PEQAD only required four qubits and 16 learnable parameters.

This notable reduction in the number of required parameters exemplifies the parameter efficiency of PEQAD, even within the constraints of current quantum hardware, such as limited gate depth and qubit count. A reduction in the number of parameters also results in a reduction in the complexity and difficulty of the training process, which is a crucial factor for machine learning algorithms. Moreover, the capacity to attain robust performance with a reduced number of parameters suggests that our approach can readily scale as quantum hardware advances, ultimately facilitating the utilization of more sophisticated and profound quantum models. These insights highlight the potential of PEQAD in real-world quantum machine learning tasks and provide guidance for the design of future experiments aimed at further exploring the capabilities of quantum computing.

Furthermore, during the experiments on the superconducting processor oneD12, as shown in Fig. 8(b), we observed some fluctuations in the loss function during optimization, which may be due to hardware instability inherent in the superconducting quantum device. Nevertheless, these fluctuations did not affect the final optimization result, largely due to the algorithm's incorporation of postprocessing techniques such as regularization, which conferred robustness against such perturbations.

All the findings above underscore the practical potential of PEQAD and suggest that QML methods have the potential to be effectively implemented on quantum hardware. The decision to forgo error mitigation is deliberate, as we aimed to highlight the inherent noise tolerance of QML algorithms. This demonstrates the progress made in quantum machine learning and indicates that QML methods, when carefully designed, may be well suited for the NISQ era, where noise and limited quantum circuit depth remain significant challenges.

VII. CONCLUSION

In this work, we present several contributions that have advanced the field of quantum machine learning for anomaly detection. First, we develop PEQAD, a quantum machine learning algorithm designed to be compatible with the NISQ device while effectively handling general image datasets. To establish a theoretical foundation, we provide a comprehensive examination of its expressivity, which offers a basis for its implementation on quantum hardware. We further validate the effectiveness and robustness of PEQAD by evaluating it on practical datasets, including different benchmarks, where it shows superior performance compared to various classical methods. Furthermore, we successfully implemented PEQAD on the superconducting quantum device, demonstrating the feasibility of quantum-based anomaly detection and highlighting the potential benefits of reduced parameter requirements in quantum computing for this domain. By bridging quantum machine learning and anomaly detection, we demonstrate the use of quantum machine learning on widely relevant datasets, showcasing its diverse applications and reinforcing its role as a viable tool for addressing more real-world challenges. In summary, our method advances quantum anomaly detection by integrating parameter efficiency, scalability, and adaptability to general image datasets. Distinct from prior quantum anomaly methods, our approach demonstrates versatility and

practicality across general image datasets, supported by theoretical analysis and validated through hardware experiments, emphasizing its parameter efficiency.

In the next phase of our work, we will focus on developing robust statistical techniques for anomaly detection and deriving generalization bounds to enhance algorithm robustness and applicability. We plan to incorporate error mitigation techniques and scale our experiments to larger and different quantum chips. We hope these improvements yield even more promising results, further showcasing the capabilities of PEQAD. QML offers potential advantages in terms of accuracy, parameter efficiency, and scalability concerning training set size, all of which are particularly relevant advantages when applied to real-world classical datasets.

ACKNOWLEDGMENTS

The authors acknowledge funding support from the UK Engineering and Physical Sciences Research Council (EP-SRC) under the project “Software Environment for Actionable & VVUQ-evaluated Exascale Applications (SEAVEA)” (Grant No. EP/W007711/1). The authors acknowledge Tim Weaving for his insightful comments and suggestions. The authors also thank the members of the Qchemistry (Quantum Chemistry) Group in the Centre for Computational Science of University College London for their helpful input during this research. The authors acknowledge hardware support from the Institute of Quantum Information and Quantum Technology Innovation, Chinese Academy of Sciences in Shanghai, China. The authors also acknowledge use of the PennyLane AI and Qiskit software for this work.

DATA AVAILABILITY

The data supporting this study's findings are available within the article. All datasets used in this paper are publicly accessible: MNIST can be downloaded from [63], Fashion-MNIST from [64], and CIFAR-10 from [65]. The code for PEQAD is available at [66].

APPENDIX A: EXPRESSIVITY OF PEQAD

The intuition of the proof is as follows. The proof of Theorem 1 employs the definition of the operator norm.

Definition A 1 (Operator norm). Assume that A is an $n \times n$ matrix. The operator norm of A is defined as

$$\|A\| = \sup_{\|x\|_2=1, x \in \mathbb{C}^n} \|Ax\|. \quad (\text{A1})$$

Alternatively, $\|A\| = \sqrt{\lambda_1(AA^\dagger)}$, where $\lambda_i(AA^\dagger)$ is the i th largest eigenvalue of the matrix $AA^\dagger$.

Besides the above definition, the proof of Theorem 1 leverages the following two lemmas. In particular, the first lemma allows us to employ the covering number of one metric space $(\mathcal{H}_1, d_1)$ to bound the covering number of another metric space $(\mathcal{H}_2, d_2)$.

Lemma A 1 (Lemma 5 [67]). Let $(\mathcal{H}_1, d_1)$ and $(\mathcal{H}_2, d_2)$ be metric spaces, and $f : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ be K -Lipschitz continuous such that

$$d_2(f(x), f(y)) \leq K d_1(x, y), \quad \forall x, y \in \mathcal{H}_1. \quad (\text{A2})$$

Then, their covering numbers satisfy

$$N(\mathcal{H}_2, \varepsilon, d_2) \leq N(\mathcal{H}_1, \varepsilon/K, d_1). \quad (\text{A3})$$

The second lemma presents the covering number of the operator group

$$\mathcal{H}_{\text{circ}} := \{U(\theta)^\dagger O U(\theta) \mid \theta \in \Theta\}, \quad (\text{A4})$$

where $U(\theta) = \prod_{i=1}^k u_i(\theta_i)$ and only P parameters in $U(\theta)$ are trainable. The detail can be found in Ref. [47].

Lemma A2 (Lemma 2 [47]). Following notations in Theorem 1, suppose that the q -qubit *ansätze* used contain a total of q trainable parameters, with each gate acting on at most m qubits. The ε -covering number for the operator group $\mathcal{H}_{\text{circ}}$ with respect to the operator-norm distance satisfies

$$N(\mathcal{H}_{\text{circ}}, \varepsilon, \|\cdot\|) \leq \left(\frac{7P\|O\|}{\varepsilon} \right)^{q^{2m}P}, \quad (\text{A5})$$

where P is the number of trainable parameters in the quantum circuit, $\|O\|$ is the operator norm of the observable O , and q is the number of qubits.

The covering number, $N(\mathcal{H}, \varepsilon, \|\cdot\|)$, provides a measure of the complexity of the hypothesis space $\mathcal{H}$. In the context of QML, this is particularly useful as it helps quantify the expressivity of quantum models by determining how well the hypothesis space can approximate different data distributions. The above formulas are derived following Ref. [47] and apply to all postprocessing-free VQAs, which means they can be used for theoretical calculations of expressivities for postprocessing-free variational quantum one-class algorithms such as VQOCC.

Next, we proceed with the derivation to prove Theorem 1. Recall the definition of the hypothesis space $\mathcal{H}$ in Eq. (2) and Lemma A1. When $\mathcal{H}_1$ refers to the hypothesis space $\mathcal{H}$ and $\mathcal{H}_2$ refers to the unitary group $U(d^N)$, the upper bound of the covering number of $\mathcal{H}$, i.e., $N(\mathcal{H}, \varepsilon, d_2)$, can be derived by first quantifying the Lipschitz constant K and then relating this to the covering number of the operator group $\mathcal{H}_{\text{circ}}$ from Lemma A2. Under these observations, we now analyze the upper bound of the covering number $N(\mathcal{H}, \varepsilon, \|\cdot\|)$. Consider an arbitrary element U from the function space $\mathcal{H}_{\text{circ}}$. Since the function space $\mathcal{H}_{\text{circ}}$ is bounded and can be covered by a finite number of elements concerning the operator norm $\|\cdot\|$, for any $\varepsilon > 0$, there exists an element U_ε in the covering set such that the distance between U and U_ε satisfies

$$\|U^\dagger O U - U_\varepsilon^\dagger O U_\varepsilon\| \leq \varepsilon. \quad (\text{A6})$$

This follows from the definition of the covering number, which ensures that for every element U in $\mathcal{H}_{\text{circ}}$, there exists an adjacent element U_ε in the covering set whose distance from U is bounded by ε . Therefore, we can derive the relation between $d_2(S_{\text{PEQAD}}(U), S_{\text{PEQAD}}(U_\varepsilon))$ and $d_1(U(\theta)^\dagger O U(\theta), (U_\varepsilon(\theta))^\dagger O (U_\varepsilon(\theta)))$ as follows:

$$\begin{aligned} d_2(S_{\text{PEQAD}}(U), S_{\text{PEQAD}}(U_\varepsilon)) &= |(\text{Tr}(U(\theta)^\dagger O U(\theta)\rho) - \mathbf{c})^2 - R^2 \\ &\quad - ((\text{Tr}(U_\varepsilon(\theta)^\dagger O U_\varepsilon(\theta)\rho) - \mathbf{c})^2 - R^2)|, \end{aligned} \quad (\text{A7})$$

where $\text{Tr}(U(\theta)^\dagger O U(\theta)\rho)$ represents the expectation value of the observable O concerning the quantum state ρ transformed

by the unitary $U(\theta)$. This expectation value is a critical component in determining the anomaly score S_{PEQAD} , as it encodes the feature transformation performed by the VQC. S_{PEQAD} represents the anomaly score in the article. This is the final function output by PEQAD for determining whether the data are anomalous, which is the function we have learned. We abbreviate $\text{Tr}(U(\theta)^\dagger O U(\theta)\rho)$ as Tr and $\text{Tr}((U_\varepsilon(\theta))^\dagger O (U_\varepsilon(\theta))\rho)$ as Tr_ε . Under this notation, the original equations can be written as follows:

$$\begin{aligned} d_2(S_{\text{PEQAD}}(U), S_{\text{PEQAD}}(U_\varepsilon)) &= |(\text{Tr} - \mathbf{c})^2 - (\text{Tr}_\varepsilon - \mathbf{c})^2| \\ &= (\text{Tr} + \text{Tr}_\varepsilon - 2\mathbf{c})(\text{Tr} - \text{Tr}_\varepsilon) \\ &\leq \|U(\theta)^\dagger O U(\theta) + U_\varepsilon(\theta)^\dagger O U_\varepsilon(\theta) - 2\mathbf{c}\| \\ &\quad \cdot \|U(\theta)^\dagger O U(\theta) - U_\varepsilon(\theta)^\dagger O U_\varepsilon(\theta)\| \cdot \text{Tr}^2(\rho) \\ &\leq |2\|O\| - 2\|\mathbf{c}\|| \cdot d_1(U(\theta)^\dagger O U(\theta), \\ &\quad U_\varepsilon(\theta)^\dagger O U_\varepsilon(\theta)), \end{aligned} \quad (\text{A8})$$

where the first inequality uses the triangle inequality once and the Cauchy-Schwartz inequality twice, and the second inequality employs the triangle inequality, $\text{Tr}^2(\rho) = 1$, $\|U\| = 1$ and also

$$\|U^\dagger O U - U_\varepsilon^\dagger O U_\varepsilon\| = d_1(U^\dagger O U, U_\varepsilon^\dagger O U_\varepsilon). \quad (\text{A9})$$

By applying the triangle inequality and the Cauchy-Schwarz inequality, we bound the difference between the anomaly scores $S_{\text{PEQAD}}(U)$ and $S_{\text{PEQAD}}(U_\varepsilon)$. The triangle inequality allows us to separate the difference into terms involving the operator norms of $U(\theta)^\dagger O U(\theta)$ and $U_\varepsilon(\theta)^\dagger O U_\varepsilon(\theta)$, which are further bounded using the properties of operator norms. From here, we obtain the relationship between $(\mathcal{H}, d_2)$ and $(\mathcal{H}_{\text{circ}}, d_1)$.

Next, we use Lemmas A1 and A2 and consider that

$$d_2(S_{\text{PEQAD}}(\mathbf{x}), S_{\text{PEQAD}}(\mathbf{y})) \leq K d_1(\mathbf{x}, \mathbf{y}), \quad (\text{A10})$$

where $K = |2\|O\| - 2\|\mathbf{c}\||$. In our case, K is determined by the operator norm of O and the $\mathbf{c}$. Specifically, we can derive that $K = |2\|O\| - 2\|\mathbf{c}\|| \leq 2\|O\| + 2\|\mathbf{c}\|$ by removing the absolute value. Now we have $K \leq 2\|O\| + 2\|\mathbf{c}\|$ and can further obtain

$$N(\mathcal{H}, \varepsilon, \|\cdot\|) \leq N(\mathcal{H}_{\text{circ}}, \varepsilon/K, \|\cdot\|). \quad (\text{A11})$$

Substituting the value of $N(\mathcal{H}_{\text{circ}}, \varepsilon, \|\cdot\|)$ from Lemma A2, we get

$$N(\mathcal{H}, \varepsilon, \|\cdot\|) \leq \left(\frac{7P(2\|O\| + 2\|\mathbf{c}\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}. \quad (\text{A12})$$

Therefore, we have obtained Theorem 1.

For further consideration, VQC is a fully flexible component of our model. PEQAD can express the entire space that VQC is capable of. According to Lemma A2, we can assume that the maximum theoretical covering number for a standalone VQC is given by:

$$\sup N(\mathcal{H}_{\text{circ}}, \varepsilon, \|\cdot\|) = \left(\frac{7P'\|O\|}{\varepsilon} \right)^{q^{2m}P'}, \quad (\text{A13})$$

where P' represents the number of trainable parameters in the VQC model and q is the number of qubits.

Since PEQAD contains VQC as a subset, the supremum of the covering number of PEQAD must be greater than or equal to that of VQC. Therefore, we can have $P' = P$ and establish the following inequality for the covering number of PEQAD:

$$\sup N(\mathcal{H}, \varepsilon, \|\cdot\|) \geq \sup N(\mathcal{H}_{\text{circ}}, \varepsilon, \|\cdot\|) = \left(\frac{7P\|O\|}{\varepsilon} \right)^{q^{2m}P}. \quad (\text{A14})$$

At the same time, the upper bound of the expressivity of PEQAD can be constrained by the following expression derived from the theoretical analysis of its function space:

$$\sup N(\mathcal{H}, \varepsilon, \|\cdot\|) \leq \left(\frac{7P(2\|O\| + 2\|\mathbf{c}\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}. \quad (\text{A15})$$

Therefore, the expressivity of PEQAD is bounded as follows:

$$\begin{aligned} \left(\frac{7P\|O\|}{\varepsilon} \right)^{q^{2m}P} &\leq \sup N(\mathcal{H}, \varepsilon, \|\cdot\|) \\ &\leq \left(\frac{7P(2\|O\| + 2\|\mathbf{c}\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}. \end{aligned} \quad (\text{A16})$$

This is Theorem 2. In particular, considering that $\|O\| = 1$ in our emulations and experiments, the upper bound of PEQAD's expressivity is larger than that of its counterparts. Compared to postprocessing-free variational quantum algorithms such as VQOCC, the expressivity of PEQAD benefits from the additional structure provided by the hypersphere constraint in the anomaly detection setting. Rather than reducing the hypothesis space, this constraint expands the expressive capacity of PEQAD, allowing it to capture a broader range of patterns. As a result, the expressivity of PEQAD is greater than other VQA-based anomaly detection methods, providing a more flexible framework for detecting complex anomalies. Intuitively, the expressivity of a quantum model, much like in classical machine learning, dictates its ability to model complex data distributions. While excessive expressivity can lead to overfitting in some cases, PEQAD's design balances this by utilizing a QNN architecture that enhances flexibility without succumbing to overfitting.

APPENDIX B: EXPRESSIVITY OF PEQAD ON NOISY DEVICES

In this section, we extend the expressivity analysis of PEQAD to quantum devices operating under noisy conditions. Similar to the proof of expressivity in the ideal case, we derive the covering number of PEQAD when the quantum operations are subject to noise.

We begin by recalling the covering number for PEQAD in the ideal (noise-free) case:

$$N(\mathcal{H}, \varepsilon, \|\cdot\|) \leq \left(\frac{7P(2\|O\| + 2\|\mathbf{c}\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}, \quad (\text{B1})$$

where $\mathcal{H}$ denotes the hypothesis space spanned by the PEQAD model, P is the number of trainable parameters, and q is the number of qubits.

Next, we consider the noisy case. Under noisy conditions, the relation between the distance in the output states and the distance in the unitary operators becomes more complex; i.e., the depolarization channel $\mathcal{E}_p(\cdot)$ is applied to each quantum gate in $U(\cdot)$. Following the explicit form of the depolarization channel, the distance d_2 between the noisy output states and the distance d_1 between the unitary operators satisfy the following relation:

$$\begin{aligned} d_2(\text{Tr}(O\mathcal{E}_p(U\rho U^\dagger)), \text{Tr}(O\mathcal{E}_p(U_\varepsilon\rho U_\varepsilon^\dagger))) \\ &= \text{Tr}(O\mathcal{E}_p(U\rho U^\dagger) - O\mathcal{E}_p(U_\varepsilon\rho U_\varepsilon^\dagger)) \\ &= (1-p)^{N_g} \text{Tr}(O(U\rho U^\dagger - U_\varepsilon\rho U_\varepsilon^\dagger)) \\ &\leq (1-p)^{N_g} \|U^\dagger O U - U_\varepsilon^\dagger O U_\varepsilon\| \text{Tr}(\rho) \\ &= (1-p)^{N_g} \|U^\dagger O U - U_\varepsilon^\dagger O U_\varepsilon\| \\ &= (1-p)^{N_g} d_1(U, U_\varepsilon), \end{aligned} \quad (\text{B2})$$

where N_g is the number of gates that are influenced by noise.

The second equality uses the depolarizing noise property. The inequality follows from applying the contractive property of the trace norm, and $\text{Tr}(\rho) = 1$ is used to simplify the expression.

This result indicates that the term K is

$$K = (1-p)^{N_g}. \quad (\text{B3})$$

Using Lemma A2, the covering number of the VQA under depolarizing noise is then upper bounded by

$$N(\mathcal{H}_{\text{circ}}, \varepsilon, \|\cdot\|) \leq (1-p)^{N_g} \left(\frac{7P\|O\|}{\varepsilon} \right)^{q^{2m}P}. \quad (\text{B4})$$

The bound reflects how the presence of depolarizing noise reduces the expressivity of the quantum variational circuit.

Following the same reasoning as in Theorem 1, we obtain that the expressivity of PEQAD in the presence of noise is bounded by

$$N(\mathcal{H}, \varepsilon, \|\cdot\|) \leq (1-p)^{N_g} \left(\frac{7P(2\|O\| + 2\|\mathbf{c}\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}, \quad (\text{B5})$$

where p represents the noise rate. This equation quantifies the effect of noise on the expressive power of PEQAD and offers insight into its limitations in practical, noisy quantum devices.

In this way, we derive Theorem 3. This bound demonstrates that the expressivity of PEQAD is reduced in noisy environments due to the noise factor $(1-p)^{N_g}$. As the noise level increases (i.e., as p approaches 1), the effective hypothesis space is reduced, limiting the model's ability to capture complex patterns.

APPENDIX C: EXPRESSIVITY OF THE CLASSICAL MACHINE LEARNING COUNTERPARTS

To compare the expressivity of PEQAD with classical counterparts, we provide the detailed derivation of the covering number bound for classical neural networks based on Bartlett *et al.*'s [51] Theorem 3.3. The covering number bound

for a classical neural network is given by

$$\ln N(\mathcal{H}_c, \varepsilon, \|\cdot\|_2) \leq \frac{t^2 \ln(2W^2)}{\varepsilon^2} \left(\prod_{j=1}^L s_j^2 \rho_j^2 \right) \left(\sum_{i=1}^L \left(\frac{b_i}{s_i} \right)^{2/3} \right)^3, \quad (\text{C1})$$

where t is the norm of the data matrix and p represents the noise rate. Since the training set is fixed in our comparison emulations and experiments, we treat the norm of the data matrix as a constant t to simplify our analysis. s_i represents the spectral norm bound of the weight matrix at layer i . b_i represents the $(2, 1)$ matrix norm bound of the weight matrix at layer i . ρ_i represents the Lipschitz constant of the activation function at layer i . W is the maximum dimension of the matrices in the network layers. We can express s_i and b_i regarding network structure to further analyze their impact.

Spectral norm bound s_i can be estimated as

$$s_i \leq \sqrt{m_i n_i} \|W_i\|_\infty, \quad (\text{C2})$$

where m_i and n_i are the output and input dimensions of the weight matrix W_i at layer i , respectively, and $\|W_i\|_\infty$ represents the maximum absolute value of the elements in the weight matrix W_i .

For the $(2, 1)$ norm bound b_i of the weight matrix at layer i ,

$$b_i \leq n_i \sqrt{m_i} \|W_i\|_\infty. \quad (\text{C3})$$

Substituting the upper bounds of s_i and b_i into the original covering number bound, we have

$$\ln N(\mathcal{H}_c, \varepsilon, \|\cdot\|_2) \leq \frac{t^2 \ln(2W^2)}{\varepsilon^2} \left(\prod_{j=1}^L (\sqrt{m_j n_j} \|W_j\|_\infty)^2 \rho_j^2 \right) \times \left(\sum_{i=1}^L \left(\frac{n_i \sqrt{m_i} \|W_i\|_\infty}{\sqrt{m_i n_i} \|W_i\|_\infty} \right)^{2/3} \right)^3. \quad (\text{C4})$$

Further simplifying,

$$\ln N(\mathcal{H}_c, \varepsilon, \|\cdot\|_2) \leq \frac{t^2 \ln(2W^2)}{\varepsilon^2} \left(\prod_{j=1}^L m_j n_j \|W_j\|_\infty^2 \rho_j^2 \right) \times \left(\sum_{i=1}^L \left(\frac{\sqrt{m_i}}{\sqrt{n_i}} \right)^{2/3} \right)^3. \quad (\text{C5})$$

Finally, we can represent W as the maximum of $m_j \times n_j$:

$$W = \max(m_j \times n_j). \quad (\text{C6})$$

We also note that the Lipschitz constant of the activation functions used in our networks is always less than or equal to 1. For example, the ReLU activation function has a Lipschitz constant of 1, commonly used in deep networks. Thus, the

covering number bound becomes

$$\ln N(\mathcal{H}_c, \varepsilon, \|\cdot\|_2) \leq \frac{t^2 (\ln(2) + 2 \ln(\max(m_j \times n_j)))}{\varepsilon^2} \times \left(\prod_{j=1}^L m_j n_j \|W_j\|_\infty^2 \right) \times \left(\sum_{i=1}^L \left(\frac{\sqrt{m_i}}{\sqrt{n_i}} \right)^{2/3} \right)^3. \quad (\text{C7})$$

Thus, we obtained Theorem 4.

APPENDIX D: ANALYSIS AND COMPARISON OF PEQAD AND CLASSICAL METHODS

In this section, we provide a detailed comparison of the expressivity bounds of PEQAD and classical deep machine learning anomaly detection models, focusing on their respective covering number bounds. Recall that the expressivity of PEQAD can be characterized by the following covering number bound:

$$N(\mathcal{H}_q, \varepsilon, |\cdot|) \leq \left(\frac{7P(2\|O\| + 2\|c\|)\|O\|}{\varepsilon} \right)^{q^{2m}P}. \quad (\text{D1})$$

Taking the natural logarithm of both sides, we obtain

$$\ln N(\mathcal{H}_q, \varepsilon, |\cdot|) \leq q^{2m}P \cdot \ln \left(\frac{7P(2\|O\| + 2\|c\|)\|O\|}{\varepsilon} \right). \quad (\text{D2})$$

This logarithmic form facilitates a direct comparison with the expressivity bound of classical models. For a classical neural network consisting of two layers, where the first layer has an input dimension of 40 and an output dimension of 10, and the second layer has an input dimension of 10 and an output dimension of 2, the covering number bound can be expressed as

$$\ln N(\mathcal{H}_c, \varepsilon, \|\cdot\|_2) \leq \frac{t^2 (\ln(2) + 2 \ln(400))}{\varepsilon^2} \times 8000 \times \|W_1\|_\infty^2 \|W_2\|_\infty^2 \times (2^{-2/3} + 5^{-1/3})^3. \quad (\text{D3})$$

This bound indicates that the expressivity of classical neural networks depends on a polynomial relationship between various network parameters, including input-output dimensions, the number of layers, and the number of trainable parameters.

For PEQAD with four qubits ($q = 4$), 16 trainable parameters ($P = 16$), and a maximum gate interaction involving two qubits ($k = 2$), the covering number bound can be simplified to

$$\ln N(\mathcal{H}_q, \varepsilon, |\cdot|) \leq 4096 \cdot \ln \left(\frac{7 \times 16 (2\|O\| + 2\|c\|)\|O\|}{\varepsilon} \right). \quad (\text{D4})$$

Although it is difficult to compare the two directly, this bound shows that the expressivity of PEQAD scales in a polynomial fashion, similar to the classical model. However, PEQAD achieves this level of expressivity with significantly fewer

parameters compared to the classical model. In both models, expressivity is influenced by the number of parameters. Still, in different forms, the classical model's expressivity depends on input-output dimensions, layer size, and the magnitude of the weights, forming a complex polynomial relationship, while PEQAD's expressivity also has a polynomial dependence on the number of trainable parameters P and the number of qubits q . Under the given conditions—4 qubits and 16 trainable parameters for PEQAD versus a classical two-layer network—the expressivity bounds can be comparable. This suggests that PEQAD can achieve an expressivity similar to classical networks, even with fewer parameters. Moreover, the true strength of PEQAD lies in its potential to leverage quantum properties such as entanglement and superposition, which might provide practical advantages in certain scenarios, particularly for high-dimensional problems or complex data distributions. Thus, we illustrate the comparison of the expressivity of our algorithm and a classical deep machine learning anomaly detection through this specific example, which is part of the conclusion. In the experimental section, we further validate the capability of our algorithm through empirical results.

APPENDIX E: QUANTITATIVE COMPARISON WITH TYPICAL PARAMETERS

To provide a more tangible and straightforward comparison, we now quantify the expressivity bounds derived in our theoretical analysis. We substitute typical parameter values—drawn from our hardware experiments and common deep learning architectures—to illustrate the claim of parameter efficiency. Our goal is to estimate the number of classical parameters required for a network to achieve a theoretical expressivity comparable to that of our small-scale quantum model. We begin with the expressivity bound for PEQAD, as given in Theorem 1. We consider the model implemented on our quantum hardware, which is characterized by the following parameters: Number of qubits, $q = 4$. Number of trainable parameters, $P = 16$. Maximum gate locality, $m = 2$ (for our two-qubit CNOT gates). Furthermore, we make the standard assumptions that the observable is a Pauli operator, thus its operator norm $\|O\| = 1$, and that the hypersphere's center norm $\|c\|$ is on the order of 1, which is reasonable for normalized data. Substituting these values into the logarithmic covering number bound,

$$\begin{aligned} \ln N(\mathcal{H}_q, \epsilon, |\cdot|) &\leq q^{2m} P \cdot \ln \left(\frac{7P(2\|O\| + 2\|c\|)\|O\|}{\epsilon} \right) \\ &\leq 4^{2 \times 2} \times 16 \cdot \ln \left(\frac{7 \times 16(2 \cdot 1 + 2 \cdot 1) \cdot 1}{\epsilon} \right) \\ &\leq 256 \times 16 \cdot \ln \left(\frac{448}{\epsilon} \right) \\ &\leq 4096 \cdot \ln \left(\frac{448}{\epsilon} \right). \end{aligned} \quad (E1)$$

This result indicates that the theoretical expressivity of our 16-parameter quantum model scales with a prefactor of approximately 4096.

Next, we analyze the bound for a classical deep network (DSVDD) from Theorem 4. We consider a typical five-layer

network ($L = 5$). We define an architecture where the input dimensionality matches our quantum encoding ($n_1 = 16$) and the network width is inspired by the suggestion of $W = 256$:

$$\begin{aligned} 16 &\xrightarrow{n_1=16, m_1=256} 256 \xrightarrow{n_2=256, m_2=128} 128 \xrightarrow{n_3=128, m_3=64} \\ 64 &\xrightarrow{n_4=64, m_4=32} 32 \xrightarrow{n_5=32, m_5=10} 10. \end{aligned}$$

The total number of parameters for this network exceeds 45 000. We again assume normalized data and weights, such that the data norm $t \approx 1$ and the weight matrix norm $\|W_j\|_\infty \approx 1$. The classical expressivity bound is highly sensitive to the term $\prod_{j=1}^L m_j n_j$:

$$\begin{aligned} \prod_{j=1}^L m_j n_j &= (256 \times 16) \times (128 \times 256) \times (64 \times 128) \\ &\quad \times (32 \times 64) \times (10 \times 32) \\ &\approx (4.1 \times 10^3) \times (3.3 \times 10^4) \times (8.2 \times 10^3) \\ &\quad \times (2.0 \times 10^3) \times (3.2 \times 10^2) \\ &\approx 7.1 \times 10^{17}. \end{aligned} \quad (E2)$$

Even without calculating the other prefactors, it is evident that this product term, which is directly related to the network's parameter count and dimensions, results in an astronomically large value for the expressivity bound.

This illustrative calculation reveals a stark contrast in how the models' parameters contribute to their theoretical expressivity. To achieve an expressivity bound ($\ln N_c$) on the same order of magnitude as that of the quantum model [$\ln N_q \sim O(10^3)$], the classical network relies on a massive parameter count, which is reflected in the extremely large product-of-dimensions term. This provides a quantitative, albeit approximate, justification for our claim of parameter efficiency. It demonstrates how a quantum model, by leveraging the high dimensionality of Hilbert space (captured by the q^{2mP} term), may achieve a rich expressive capacity with substantially fewer parameters than a classical deep network. This theoretical insight aligns well with our empirical findings, where the 16-parameter PEQAD performed comparably to a classical DSVDD requiring over 300 parameters.

APPENDIX F: BARREN PLATEAU PHENOMENON FOR PEQAD

In this section, we will show that even if our algorithm increases the expressive power using postprocessing, it does not accelerate the BP phenomenon [38]. We provide the detailed derivation showing that the PEQAD loss function $(E(\theta) - c)^2 - R^2$ exhibits the BP phenomenon under a one-design unitary distribution. While the expected gradient vanishes, the presence of the factor $(E(\theta) - c)$ can modulate the rate of decay, which provides a nuanced understanding of PEQAD's expressivity without exacerbating the BP problem. This Appendix further provides the theoretical derivation supporting the claim in Sec. V that the PEQAD ansatz does not exhibit BP on our physical circuit.

The PEQAD loss function is defined as

$$L(\theta) = (E(\theta) - c)^2 - R^2, \quad (F1)$$

where R^2 is a constant. Since R^2 does not depend on θ , it does not contribute to the gradient. Thus, for gradient analysis, we simplify the loss function to

$$L(\theta) = (E(\theta) - c)^2. \quad (\text{F2})$$

Here, $E(\theta)$ is the expectation value:

$$E(\theta) = \langle 0|U^\dagger(\theta)OU(\theta)|0\rangle. \quad (\text{F3})$$

The gradient of $L(\theta)$, with respect to the k th parametrized quantum gate, is given by

$$\partial_k L(\theta) = 2(E(\theta) - c) \cdot \partial_k E(\theta). \quad (\text{F4})$$

Here, the gradient of $E(\theta)$, with respect to the k th parametrized quantum gate, is expressed as

$$\partial_k E(\theta) = \frac{\partial}{\partial \theta_k} \langle 0|U^\dagger(\theta)OU(\theta)|0\rangle. \quad (\text{F5})$$

Using the parametrized unitary decomposition $U(\theta) = U_+U_-$, where U_+ represents the product of gates from the $(k+1)$ th to the last parametrized gate, and U_- represents the product of gates from the first to the k th parametrized gate, we have

$$V_k = \frac{\partial U_+}{\partial \theta_k} U_+^\dagger. \quad (\text{F6})$$

The derivative of $E(\theta)$ is given by

$$\partial_k E(\theta) = \frac{\partial}{\partial \theta_k} \langle 0|U^\dagger(\theta)OU(\theta)|0\rangle. \quad (\text{F7})$$

Using the decomposition $U(\theta) = U_+U_-$, we write

$$\frac{\partial U(\theta)}{\partial \theta_k} = U_+ \frac{\partial U_-}{\partial \theta_k} + \frac{\partial U_+}{\partial \theta_k} U_-. \quad (\text{F8})$$

Substituting this into $\partial_k E(\theta)$, we have

$$\partial_k E(\theta) = \langle 0|\frac{\partial U^\dagger(\theta)}{\partial \theta_k}OU(\theta)|0\rangle + \langle 0|U^\dagger(\theta)O\frac{\partial U(\theta)}{\partial \theta_k}|0\rangle. \quad (\text{F9})$$

For the first term,

$$\frac{\partial U^\dagger(\theta)}{\partial \theta_k} = \frac{\partial U_-^\dagger}{\partial \theta_k} U_+^\dagger + U_-^\dagger \frac{\partial U_+^\dagger}{\partial \theta_k}. \quad (\text{F10})$$

Thus,

$$\begin{aligned} \langle 0|\frac{\partial U^\dagger(\theta)}{\partial \theta_k}OU(\theta)|0\rangle &= \langle 0|\frac{\partial U_-^\dagger}{\partial \theta_k}U_+^\dagger OU_+U_-|0\rangle \\ &+ \langle 0|U_-^\dagger \frac{\partial U_+^\dagger}{\partial \theta_k}OU_+U_-|0\rangle. \end{aligned} \quad (\text{F11})$$

For the second term,

$$\begin{aligned} \langle 0|U^\dagger(\theta)O\frac{\partial U(\theta)}{\partial \theta_k}|0\rangle &= \langle 0|U_-^\dagger U_+^\dagger O\frac{\partial U_-}{\partial \theta_k}|0\rangle \\ &+ \langle 0|U_-^\dagger U_+^\dagger O\frac{\partial U_+}{\partial \theta_k}U_-|0\rangle. \end{aligned} \quad (\text{F12})$$

Combining these terms and simplifying using the commutator, we obtain

$$\partial_k E(\theta) = i\langle 0|U_-^\dagger[V_k, U_+^\dagger OU_+]U_-|0\rangle, \quad (\text{F13})$$

where $V_k = \frac{\partial U_+}{\partial \theta_k} U_+^\dagger$.

Recall the loss function $L(\theta) = (E(\theta) - c)^2$, where

$$\partial_k L(\theta) = 2(E(\theta) - c)\partial_k E(\theta). \quad (\text{F14})$$

Consider the parametrized unitary $U(\theta) = U_+U_-$, with the distribution defined by $p(U_+)$ and $p(U_-)$. For the overall unitary distribution $p(U)$, we write

$$p(U) = \int dU_+ p(U_+) \int dU_- p(U_-) \delta(U_+U_- - U), \quad (\text{F15})$$

where δ is the delta function over the unitary group, ensuring $U = U_+U_-$.

The expectation of $\partial_k L(\theta)$ over the unitary group distribution is

$$\langle \partial_k L(\theta) \rangle = \int dU p(U) \partial_k L(\theta). \quad (\text{F16})$$

Substituting the decomposition of $p(U)$,

$$\begin{aligned} \langle \partial_k L(\theta) \rangle &= \int dU_+ p(U_+) \int dU_- p(U_-) \\ &\times \int dU \delta(U_+U_- - U) \cdot 2(E(\theta) - c) \partial_k E(\theta). \end{aligned} \quad (\text{F17})$$

Since $\delta(U_+U_- - U)$ ensures $U = U_+U_-$, this simplifies to

$$\langle \partial_k L(\theta) \rangle = \int dU_+ p(U_+) \int dU_- p(U_-) 2(E(\theta) - c) \partial_k E(\theta). \quad (\text{F18})$$

Substituting the expression for $\partial_k E(\theta)$,

$$\begin{aligned} \langle \partial_k L(\theta) \rangle &= \int dU_+ p(U_+) \int dU_- p(U_-) \cdot 2(E(\theta) - c) \\ &\cdot i \cdot \langle 0|U_-^\dagger[V_k, U_+^\dagger OU_+]U_-|0\rangle. \end{aligned} \quad (\text{F19})$$

With $\rho_- = U_-|0\rangle\langle 0|U_-^\dagger$, we rewrite

$$\begin{aligned} \langle \partial_k L(\theta) \rangle &= 2i \int dU_+ p(U_+) \int dU_- p(U_-) \cdot (E(\theta) - c) \\ &\cdot \text{Tr}([V_k, U_+^\dagger OU_+] \rho_-). \end{aligned} \quad (\text{F20})$$

Recall

$$E(\theta) = \text{Tr}(OU_+ \rho_- U_+^\dagger). \quad (\text{F21})$$

Case 1: If U_- is a one-design unitary matrix: Averaging over U_- , we use the one-design property:

$$\int dU_- p(U_-) \rho_- = \frac{I}{D}, \quad (\text{F22})$$

where $D = 2^q$ is the Hilbert space dimension. The averaged sense, $\rho_- \rightarrow I/D$. Substituting, we have

$$\begin{aligned} \langle \partial_k L(\theta) \rangle &= 2i \int dU_+ p(U_+) \cdot \left(\frac{\text{Tr}(O)}{D} - c \right) \\ &\cdot \text{Tr} \left([V_k, U_+^\dagger OU_+] \cdot \frac{I}{D} \right). \end{aligned} \quad (\text{F23})$$

Since the commutator's trace vanishes,

$$\text{Tr}([V_k, X]) = 0, \quad \forall X. \quad (\text{F24})$$

Thus,

$$\langle \partial_k L(\theta) \rangle = 0. \quad (\text{F25})$$

Case 2: U_+ is a one-design unitary matrix. Using the one-design property for U_+ ,

$$\int dU_+ p(U_+) U_+^\dagger O U_+ = \frac{\text{Tr}(O)}{D} I. \quad (\text{F26})$$

Substituting, we have

$$\begin{aligned} \langle \partial_k L(\theta) \rangle &= 2i \int dU_- p(U_-) \cdot \left(\frac{\text{Tr}(O)}{D} - c \right) \\ &\cdot \text{Tr} \left([V_k, U_+^\dagger O U_+] \cdot \frac{\text{Tr}(O)}{D} I \right). \end{aligned} \quad (\text{F27})$$

The commutator $[V_k, U_+^\dagger O U_+]$ vanishes, as $[V_k, I] = 0$. Thus,

$$\langle \partial_k L(\theta) \rangle = 0. \quad (\text{F28})$$

Combining these results, we have that as long as any portion of the circuit approaches a one-design, the gradient expectation will vanish, resulting in $\langle \partial_k L(\theta) \rangle = 0$. Although the expected gradient vanishes, the presence of $(E(\theta) - c)$ modulates the gradient's decay rate. One of the key observations is that when $|E(\theta) - c| > 1$, the decay of the gradient is slower, which helps maintain a balance between expressivity and optimization. In our emulations, this condition is satisfied in most cases, indicating that PEQAD generally operates in a regime where the gradient decay is mitigated, allowing for enhanced expressivity without significantly exacerbating the BP phenomenon. Even in scenarios where $|E(\theta) - c| \leq 1$, our results indicate that the algorithm does not exhibit a substantial acceleration of the BP issue.

Below, we provide a mathematical derivation to extend the scaling results of the variance of $\partial_k E(\theta)$ under two-design Haar distributions to the case of $\partial_k L(\theta)$, where

$$L(\theta) = (E(\theta) - c)^2. \quad (\text{F29})$$

The variance of $\partial_k L(\theta)$ is

$$\text{Var}[\partial_k L] = \langle (\partial_k L)^2 \rangle - \langle \partial_k L \rangle^2. \quad (\text{F30})$$

For the first term,

$$(\partial_k L)^2 = 4(E(\theta) - c)^2 \cdot (\partial_k E(\theta))^2. \quad (\text{F31})$$

Taking the expectation over the two-design distributions for either U_- or U_+ , where the other one is not two-design [38], we write

$$\langle (\partial_k L)^2 \rangle = 4 \langle (E(\theta) - c)^2 \cdot (\partial_k E(\theta))^2 \rangle. \quad (\text{F32})$$

From standard results for two-design distributions, the variance of $\partial_k E(\theta)$ scales as [38]

$$\text{Var}[\partial_k E(\theta)] = O\left(\frac{1}{D^2}\right). \quad (\text{F33})$$

For $(E(\theta) - c)^2$, the second moment remains constant so that it can be bounded as

$$\langle (E(\theta) - c)^2 \rangle = O(1). \quad (\text{F34})$$

Substituting,

$$\langle (E(\theta) - c)^2 \rangle = O(1), \quad \langle (\partial_k E(\theta))^2 \rangle = O\left(\frac{1}{D^2}\right). \quad (\text{F35})$$

Thus, we find the worst case is that

$$\langle (\partial_k L)^2 \rangle = O\left(1 \cdot \frac{1}{D^2}\right) = O\left(\frac{1}{D^2}\right). \quad (\text{F36})$$

For the second term,

$$\langle \partial_k L \rangle = \langle 2(E(\theta) - c) \cdot \partial_k E(\theta) \rangle. \quad (\text{F37})$$

From the properties of two-design unitaries

$$\langle \partial_k E(\theta) \rangle = 0. \quad (\text{F38})$$

Since $(E(\theta) - c)$ is independent of the specific gradient distribution $\partial_k E(\theta)$,

$$\langle (E(\theta) - c) \cdot \partial_k E(\theta) \rangle = \langle (E(\theta) - c) \rangle \cdot \langle \partial_k E(\theta) \rangle = 0. \quad (\text{F39})$$

Thus, for either U_+ or U_- ,

$$\langle \partial_k L \rangle = 0, \quad \langle \partial_k L \rangle^2 = 0. \quad (\text{F40})$$

Thus, the variance of the loss function's gradient, $\text{Var}[\partial_k L]$, scales with the Hilbert space dimension $D = 2^q$ as

$$\text{Var}[\partial_k L] = O\left(\frac{1}{2^{3q}}\right). \quad (\text{F41})$$

This result follows the derivation presented in Eqs. (16)–(18) of Ref. [38], which relies on a unitary decomposition $U = U_+ U_-$.

Using Chebyshev's inequality, we can bound the probability of observing significant deviations of $\partial_k C$ from its mean:

$$P(|\partial_k C - \langle \partial_k C \rangle| \geq c') \leq \frac{\text{Var}[\partial_k C]}{c'^2}. \quad (\text{F42})$$

Given that

$$\text{Var}[\partial_k C] = O\left(\frac{1}{2^{3q}}\right), \quad (\text{F43})$$

we have

$$P(|\partial_k C| \geq c') \leq \frac{O\left(\frac{1}{2^{3q}}\right)}{c'^2}. \quad (\text{F44})$$

This result demonstrates that the gradient variance decays exponentially with the number of qubits q , consistent with the barren plateau phenomenon in normal variational quantum circuits.

In conclusion, the PEQAD loss function retains the BP phenomenon inherent in its underlying variational structure if under t -design ($t \geq 2$) unitary distributions. However, our earlier analysis in the Appendixes demonstrated that the postprocessing mechanism in PEQAD may increase the expressive capacity of the model, providing a larger hypothesis space for learning. Crucially, as shown in this section, the enhanced expressive capacity of PEQAD does not significantly exacerbate the BP phenomenon. The loss function structure ensures manageable gradient decay, balancing expressivity and optimization stability.

The above analysis assumes Haar-random circuits drawn from a t -design ($t \geq 2$) distribution and uses the Weingarten formula [68]. This assumption is made to support generality, since different implementations of PEQAD may involve distinct variational ansätze. However, the specific, structured ansatz used in our implementation of PEQAD does not approximate a unitary t -design and is not suitable for the

Weingarten formula [69]. Therefore, in this section, we provide a refined derivation of the gradient behavior specific to our physical circuit structure (i.e., RY + CNOT brick-wall ansatz), leveraging direct moment computation to show that barren plateaus are mitigated under our design. Our ansatz consists of alternating layers of fixed CNOT gates arranged in a brick-wall pattern and parametrized single-qubit RY(θ) rotations, with one RY gate per qubit per layer. Let $q = 4$ be the number of qubits and $P = 16$ the total number of trainable parameters, where each layer contributes q parameters. The measurement observable is a local operator $O = Z_j$ acting on qubit j , and the loss function is again defined as $L(\theta) = (E(\theta) - c)^2$, with $E(\theta) = \langle 0|U^\dagger(\theta)OU(\theta)|0\rangle$. The gradient of the loss with respect to any parameter θ_k is given by

$$\partial_k L = 2(E(\theta) - c) \cdot \partial_k E(\theta). \quad (\text{F45})$$

To compute $\partial_k E(\theta)$, we consider the Heisenberg-evolved observable $U^\dagger(\theta)OU(\theta)$ and how it transforms under conjugation by the single-qubit gate $U_k(\theta_k) = \text{RY}(\theta_k)$. For any Hermitian operator A acting on q qubits, if U_k acts only on qubit k , then the expected conjugation over the rotation axis (assuming uniformly sampled RX, RY, and RZ) yields

$$\mathbb{E}[U_k^\dagger A U_k] = \frac{1}{3}(A + \mathbb{I}_k \otimes \text{Tr}_k A), \quad (\text{F46})$$

where $\text{Tr}_k A$ denotes the partial trace over qubit k and $\mathbb{I}_k$ is the identity operator acting on that qubit. Applying this recursively across all q qubits, and noting the locality of the observable O , the expected gradient becomes [69]

$$|\langle \partial_k L \rangle| \propto \frac{1}{3^q}. \quad (\text{F47})$$

This demonstrates that the expected gradient decays exponentially with the number of qubits q , but does not vanish entirely, in contrast to results derived under the Weingarten calculus assumption of full Haar randomness. Moreover, the variance of the gradient is shown to scale as

$$\text{Var}[\partial_k L] \propto \frac{p}{8^q \cdot P}, \quad (\text{F48})$$

where p denotes the number of effective parameters that influence the measurement observable O , which remains small due to the locality of light-cone propagation through the layered CNOT structure. In our setting with $q = 4$ and $P = 16$, the expected gradient satisfies $|\langle \partial_k L \rangle| \sim 1.2 \times 10^{-2}$, while the variance is approximately $\sim 3.1 \times 10^{-6}$, since $p = O(1)$. These results confirm that the barren plateau phenomenon has little influence on our ansatz, as the gradients remain nonzero and statistically significant even in realistic quantum circuits, enabling effective optimization of the PEQAD objective. In summary, our analysis provides a nuanced perspective on the trainability of the PEQAD algorithm. We have first shown that, consistent with established theory, our loss function is susceptible to BP when paired with a generic, highly entangling ansatz that approximates a unitary t -design. This confirms the presence of the BP phenomenon under general conditions. Crucially, however, we have also demonstrated that our specific choice of a hardware-efficient, local ansatz mitigates this issue. For this structured circuit, our direct computation confirms that the gradients remain

nonzero and statistically significant within a practical regime, ensuring the algorithm's effective optimization. This result underscores a viable path forward for developing effective variational quantum algorithms under practical constraints: The careful codesign of the algorithm's objective function and the physical ansatz is essential to guarantee trainability. The successful performance of PEQAD in both our simulations and hardware experiments showcases the robustness of this codesign approach in both theoretical and experimental contexts.

APPENDIX G: ANALYSIS OF THE THEORETICAL EXPRESSIVITY OF PEQAD AND EMULATION AGREEMENT

The analysis of the theoretical expressivity of PEQAD and its agreement with emulation results reveals several important insights. Based on the data presented in Fig. 9 and the theoretical analysis, we can observe that when the expressivity of PEQAD is relatively weak, i.e., when the target concept is excluded, training remains stable but yields unsatisfactory results, i.e., underfitting. In such cases, increasing the number of trainable parameters increases the expressivity. This leads to improved performance, as shown in the figure when we incrementally increased the number of trainable parameters from 10 to 70, resulting in an increase in AUC. Conversely, too much expressivity leads to a worse AUC, i.e., overfitting. This can be seen in the figure, where the PEQAD model with 60 trainable parameters outperforms the model with 240 parameters. In summary, finding the right balance of expressivity in PEQAD is essential. The results of our experiments underscore the importance of carefully tuning expressivity to achieve optimal performance in anomaly detection tasks.

Furthermore, the results presented in Figs. 9(a) and 9(b) offer significant insights into the expressivity of QML compared to deep classical machine learning, as theoretically derived earlier. Classical neural networks are characterized by their strong expressivity, unlike conventional QML methods, which frequently exhibit a more constrained expressivity range during target function learning. This restricted range enhances trainability but concomitantly increases the probability of failing to capture the learning target fully. However, when the optimal solution is within reach, lower expressivity frequently translates to better performance. The proposed PEQAD addresses this issue by achieving superior expressivity with a minimal number of parameters. When the complexity of the dataset challenges the expressivity of QML methods, the results demonstrate that the dynamic adjustment of trainable parameters in PEQAD, coupled with appropriate postprocessing techniques, can extend the expressivity of QML algorithms. This enables them to encompass the learning target while minimizing the expressivity space. Furthermore, the low parameter requirement of our approach makes deployment on quantum hardware feasible. Under the constraint of limiting the number of trainable parameters to fewer than 1000, PEQAD consistently exhibits superior performance, often surpassing DSVDD in most scenarios. As demonstrated in Fig. 9(a), PEQAD attains comparable outcomes to DSVDD with a mere 200 parameters, utilizing a mere 20 parameters. Moreover, augmenting the trainable

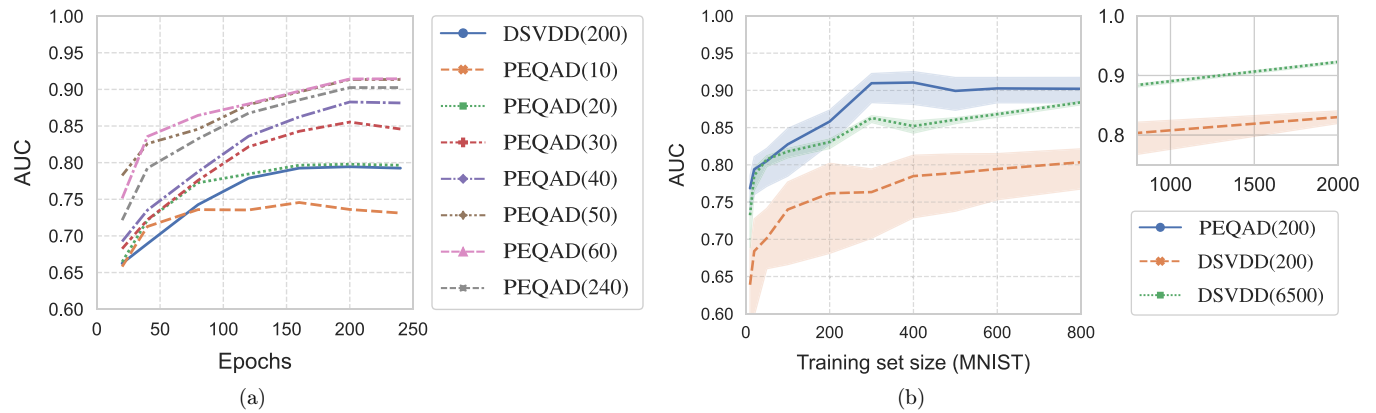


FIG. 9. Comparison of PEQAD and DSVDD on FashionMNIST and MNIST datasets. Panel (a) shows the AUC trends with different trainable parameters on FashionMNIST. As the number of trainable parameters in PEQAD increases from 10 to 70, its AUC shows an upward trend. However, when the number of parameters becomes excessively large, leading to overexpressivity, optimization becomes more challenging. Specifically, this is reflected in the AUC dropping at the same epoch. For instance, as shown in the figure, when the parameter count reaches 240, the AUC of PEQAD performs worse than with 60 parameters. Panel (b) highlights the performance under varying training set sizes on MNIST. When the number of samples in the training set is small (less than 400 in this case), PEQAD outperforms DSVDD significantly, even though DSVDD has more trainable parameters.

parameters in PEQAD amplifies its efficacy by enhancing its expressivity. As shown in Fig. 9(a), as the number of trainable parameters in PEQAD rises from 10 to 70, its AUC exhibits an upward trend. However, this enhancement is constrained. When the parameter count becomes excessively large, leading to overexpressivity, optimization becomes more challenging, as evidenced by the AUC dropping at the same epoch. For instance, when the parameter count reaches 240, the AUC of PEQAD performs worse than with 60 parameters.

Beyond the number of parameters, PEQAD demonstrates a particular advantage in handling small datasets compared to classical deep learning algorithms. As shown in Fig. 9(b),

when the number of samples in the training set is small (fewer than 400 in this case), PEQAD significantly outperforms classical DSVDD, even though DSVDD has more trainable parameters. As the training set size increases and reaches 400 samples, the performance of PEQAD stabilizes with no further improvement. On the other hand, DSVDD, with its larger number of trainable parameters, continues to improve as the training set size grows. Since the performance of PEQAD has already stabilized, we have not reported its performance beyond 800 samples, but we continue to track the performance of DSVDD.

It is worth noting that PEQAD incorporates an additional postprocessing step, contributing to increased expressivity.

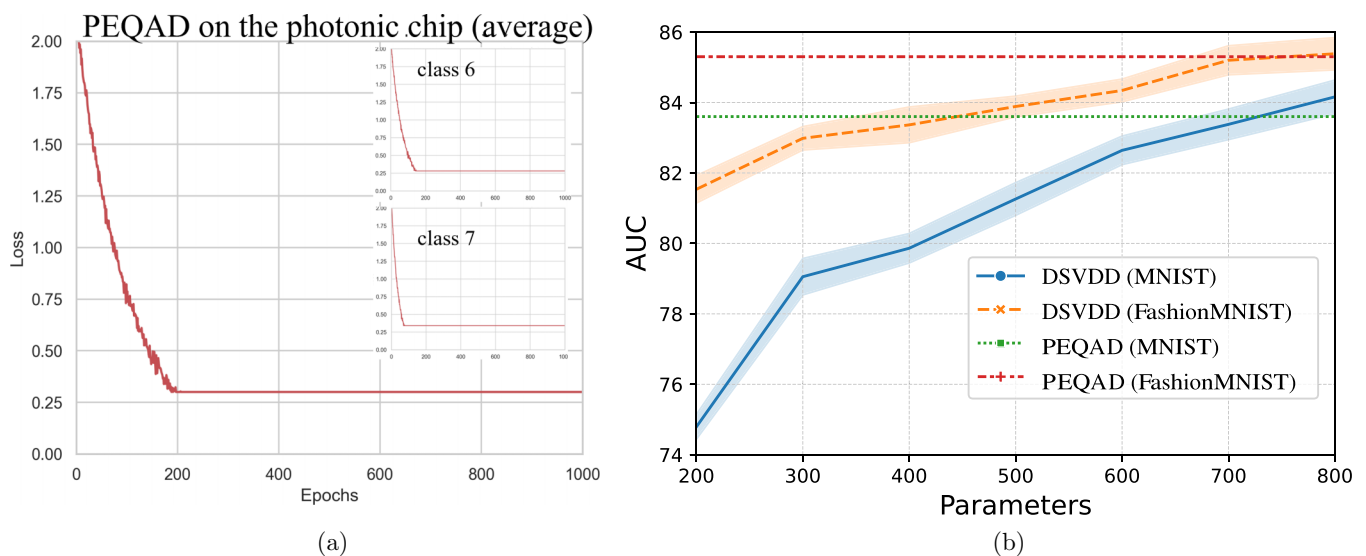


FIG. 10. (a) The optimization process of PEQAD on the photonic processor demonstrates a stable convergence with a higher final loss than that of the superconducting processor. (b) The parameter efficiency comparison between PEQAD and DSVDD on photonic quantum processors. DSVDD requires significantly more parameters to match PEQAD performance.

This enhancement in expressivity is likely the reason behind PEQAD's overall superior performance on the FashionMNIST dataset compared to VQOCC. Especially on data (classes 5 and 6) that VQOCC finds difficult to handle, PEQAD still performs well. Including the postprocessing step provides PEQAD with the capacity to capture intricate features and patterns in the data, resulting in enhanced anomaly detection accuracy. This result showcases the significance of careful model design in optimizing the performance of QML algorithms.

As shown in Fig. 6, the comparison between PEQAD and VQOCC suggests that greater expressivity can enhance the performance of quantum anomaly detection algorithms. PEQAD's superior expressive capacity enables it to capture more complex patterns, translating to better performance in practical anomaly detection tasks across diverse applications. By using PEQAD to gain more powerful expressivity, researchers can further harness the power of QML to address practical anomaly detection challenges in diverse applications. Combining theoretical insights and empirical validation helps advance the understanding and practical application of

PEQAD and other QML algorithms. The insights gained from this study pave the way for more robust and powerful QML techniques, bringing us closer to realizing the full potential of quantum computing in various domains.

APPENDIX H: PEQAD ON OTHER QUANTUM DEVICES

PEQAD can also be implemented on other quantum hardware types, such as photonic quantum processors [4]. We conducted preliminary experiments on a four-qudit photonic chip, where PEQAD achieved an AUC of over 80% using 20 parameters. Below, we present the optimization process of PEQAD on the photonic chip and compare its parameter efficiency to that of DSVDD in this setting (see Fig. 10). The optimization process of PEQAD on the photonic processor demonstrates a stable convergence with a higher final loss than that of the superconducting processor. However, due to the limited number of experimental runs, we provide these results as Appendixes. In future work, we aim to extend our validation to a broader range of quantum hardware platforms.

-
- [1] E. Bernstein and U. Vazirani, Quantum complexity theory, in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing* (ACM Press, New York, NY, 1993), pp. 11–20.
 - [2] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)* (ACM Press, New York, NY, 1996), pp. 212–219.
 - [3] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, New York, 1994), pp. 124–134.
 - [4] H.-S. Zhong *et al.*, Quantum computational advantage using photons, *Science* **370**, 1460 (2020).
 - [5] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
 - [6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature (London)* **549**, 195 (2017).
 - [7] J. Romero and A. Aspuru-Guzik, Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions, *Adv. Quantum Technol.* **4**, 2000003 (2021).
 - [8] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, *Nat. Commun.* **12**, 2631 (2021).
 - [9] J. Kübler, S. Buchholz, and B. Schölkopf, The inductive bias of quantum kernels, *Adv. Neural Inf. Process. Syst.* **34**, 12661 (2021).
 - [10] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, Challenges and opportunities in quantum machine learning, *Nat. Comput. Sci.* **2**, 567 (2022).
 - [11] A. Lavin and S. Ahmad, Evaluating real-time anomaly detection algorithms—The numanta anomaly benchmark, in *Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (IEEE, Piscataway, NJ, 2015), pp. 38–44.
 - [12] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, Anomaly-based network intrusion detection: Techniques, systems and challenges, *Comput. Secur.* **28**, 18 (2009).
 - [13] C. Phua, V. Lee, K. Smith, and R. Gayler, A comprehensive survey of data mining-based fraud detection research, [arXiv:1009.6119](https://arxiv.org/abs/1009.6119).
 - [14] T. Schlegel, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in *Proceedings of the International Conference on Information Processing in Medical Imaging, IPMI 2017*, edited by M. Niethammer *et al.*, Lecture Notes in Computer Science Vol. 10265 (Springer, Cham, 2017).
 - [15] J. An and S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, *Special Lecture on IE* **2**, 1 (2015).
 - [16] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, Deep one-class classification, in *Proceedings of the International Conference on Machine Learning* (PMLR, 2018), pp. 4393–4402.
 - [17] Q. Yu, C. Li, Y. Zhu, and T. Kurita, Convolutional autoencoder based on latent subspace projection for anomaly detection, *Methods* **214**, 48 (2023).
 - [18] J. J. Engelsma and A. K. Jain, Generalizing fingerprint spoof detector: Learning a one-class classifier, in *Proceedings of the 2019 International Conference on Biometrics (ICB)* (IEEE, Piscataway, NJ, 2019), pp. 1–8.
 - [19] G. Park, J. Huh, and D. K. Park, Variational quantum one-class classifier, *Mach. Learn.: Sci. Technol.* **4**, 015006 (2023).
 - [20] V. Belis, K. A. Woźniak, E. Puljak, P. Barkoutsos, G. Dissertori, M. Grossi, M. Pierini, F. Reiter, I. Tavernelli, and S. Vallecorsa, Quantum anomaly detection in the latent space of proton collision events at the LHC, *Commun. Phys.* **7**, 334 (2024).

- [21] A. Senokosov, V. Belis, M. Grossi, A. Hart, E. Kühn, M. Pierini, E. Puljak, F. Reiter, K. A. Wozniak, and S. Vallecorsa, Quantum-enhanced support vector machine for anomaly detection in high-energy physics, *Mach. Learn.: Sci. Technol.* **5**, 015040 (2024).
- [22] J. Romero, J. P. Olson, and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, *Quantum Sci. Technol.* **2**, 045001 (2017).
- [23] L. Lamata, U. Alvarez-Rodriguez, J. D. Martín-Guerrero, M. Sanz, and E. Solano, Quantum autoencoders via quantum adders with genetic algorithms, *Quantum Sci. Technol.* **4**, 014007 (2018).
- [24] Y. Ding, L. Lamata, M. Sanz, X. Chen, and E. Solano, Experimental implementation of a quantum autoencoder via quantum adders, *Adv. Quantum Technol.* **2**, 1800065 (2019).
- [25] M. Kieferová and N. Wiebe, Tomography and generative training with quantum Boltzmann machines, *Phys. Rev. A* **96**, 062327 (2017).
- [26] S. Jain, J. Ziauddin, P. Leonchyk, S. Yenkanchi, and J. Geraci, Quantum and classical machine learning for the classification of non-small-cell lung cancer patients, *SN Appl. Sci.* **2**, 1088 (2020).
- [27] P.-L. Dallaire-Demers and N. Killoran, Quantum generative adversarial networks, *Phys. Rev. A* **98**, 012324 (2018).
- [28] S. Lloyd and C. Weedbrook, Quantum generative adversarial learning, *Phys. Rev. Lett.* **121**, 040502 (2018).
- [29] J. Zeng, Y. Wu, J.-G. Liu, L. Wang, and J. Hu, Learning and inference on generative adversarial quantum circuits, *Phys. Rev. A* **99**, 052306 (2019).
- [30] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, *Phys. Rev. Lett.* **113**, 130503 (2014).
- [31] R. Mengoni and A. Di Pierro, Kernel methods in quantum machine learning, *Quantum Mach. Intell.* **1**, 65 (2019).
- [32] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill *et al.*, Quantum advantage in learning from experiments, *Science* **376**, 1182 (2022).
- [33] J. Cotler, H.-Y. Huang, and J. R. McClean, Revisiting dequantization and quantum advantage in learning tasks, [arXiv:2112.00811](https://arxiv.org/abs/2112.00811).
- [34] S. Chen, J. Cotler, H.-Y. Huang, and J. Li, A hierarchy for replica quantum advantage, [arXiv:2111.05874](https://arxiv.org/abs/2111.05874).
- [35] E. Parzen, On estimation of a probability density function and mode, *Ann. Math. Stat.* **33**, 1065 (1962).
- [36] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, Estimating the support of a high-dimensional distribution, *Neural Comput.* **13**, 1443 (2001).
- [37] D. M. J. Tax and R. P. W. Duin, Support vector data description, *Mach. Learn.* **54**, 45 (2004).
- [38] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [39] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [40] M. Schuld and N. Killoran, Quantum machine learning in feature Hilbert spaces, *Phys. Rev. Lett.* **122**, 040504 (2019).
- [41] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, Quantum embeddings for machine learning, [arXiv:2001.03622](https://arxiv.org/abs/2001.03622).
- [42] D. Silver, T. Patel, and D. Tiwari, Quilt: Effective multi-class classification on quantum computers using an ensemble of diverse quantum classifiers, in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI Press, Reston, VA, 2022), Vol. 36, pp. 8324–8332.
- [43] T. Patel, D. Silver, and D. Tiwari, Optic: A practical quantum binary classifier for near-term quantum computers, in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE, 2022), pp. 334–339.
- [44] Román Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, *Ann. Phys.* **349**, 117 (2014).
- [45] M. Wang, A. Huang, Y. Liu, X. Yi, J. Wu, and S. Wang, A quantum-classical hybrid solution for deep anomaly detection, *Entropy* **25**, 427 (2023).
- [46] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, General parameter-shift rules for quantum gradients, *Quantum* **6**, 677 (2022).
- [47] Y. Du, Z. Tu, X. Yuan, and D. Tao, Efficient measure for the expressivity of variational quantum algorithms, *Phys. Rev. Lett.* **128**, 080506 (2022).
- [48] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, *PRX Quantum* **3**, 010313 (2022).
- [49] V. Vapnik, *The Nature of Statistical Learning Theory* (Springer Science & Business Media, New York, 2013).
- [50] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning* (MIT Press, Cambridge, MA, 2018).
- [51] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, Spectrally-normalized margin bounds for neural networks, in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Red Hook, NY, 2017), Vol. 30.
- [52] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* **86**, 2278 (1998).
- [53] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms, [arXiv:1708.07747](https://arxiv.org/abs/1708.07747).
- [54] A. Krizhevsky and G. Hinton, Learning multiple layers of features from tiny images, Technical report, University of Toronto, 2009.
- [55] P. Perera, P. Oza, and V. M. Patel, One-class classification: A survey, [arXiv:2101.03064](https://arxiv.org/abs/2101.03064).
- [56] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi *et al.*, PennyLane: Automatic differentiation of hybrid quantum-classical computations, [arXiv:1811.04968](https://arxiv.org/abs/1811.04968).
- [57] D. P. Kingma, J. Ba, Y. Bengio, and Y. LeCun, *3rd International Conference on Learning Representation (ICLR)*, San Diego, 2015).
- [58] A. Sebastianelli, D. A. Zaidenberg, D. Spiller, B. L. Saux, and S. L. Ullo, On circuit-based hybrid quantum neural networks for remote sensing imagery classification, *IEEE J. Sel. Top. Appl.* **15**, 565 (2021).
- [59] Y. Zeng, H. Wang, J. He, Q. Huang, and S. Chang, A multi-classification hybrid quantum neural network using an all-qubit multi-observable measurement strategy, *Entropy* **24**, 394 (2022).

- [60] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, Transfer learning in hybrid classical-quantum neural networks, *Quantum* **4**, 340 (2020).
- [61] Institute of quantum Information and chinese academy of sciences quantum technology innovation, oned12: A 12-qubit superconducting quantum processor.
- [62] J. C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Trans. Autom. Control* **37**, 332 (1992).
- [63] <http://yann.lecun.com/exdb/mnist/>.
- [64] <https://github.com/zalandoresearch/fashion-mnist>.
- [65] <https://www.cs.toronto.edu/kriz/cifar.html>.
- [66] <https://github.com/UCL-CCS/PEQAD.git>.
- [67] T. Barthel and J. Lu, Fundamental limitations for measurements in quantum many-body systems, *Phys. Rev. Lett.* **121**, 080406 (2018).
- [68] D. Weingarten, Asymptotic behavior of group integrals in the limit of infinite rank, *J. Math. Phys.* **19**, 999 (1978).
- [69] Y. Yao and Y. Hasegawa, Direct gradient computation of parameterized quantum circuits, *arXiv:2503.05145*.