Machine Learning: Science and Technology

PAPER • OPEN ACCESS

Variational quantum one-class classifier

To cite this article: Gunhee Park *et al* 2023 *Mach. Learn.: Sci. Technol.* **4** 015006

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
5 October 2022

REVISED
19 December 2022

ACCEPTED FOR PUBLICATION
3 January 2023

PUBLISHED
20 January 2023

PAPER

Variational quantum one-class classifier

Gunhee Park[1], **Joonsuk Huh**[2,3,4,*] **and Daniel K Park**[5,6,*]

[1] Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, United States of America
[2] Department of Chemistry, Sungkyunkwan University, Suwon 16419, Republic of Korea
[3] SKKU Advanced Institute of Nanotechnology, Sungkyunkwan University, Suwon 16419, Republic of Korea
[4] Institute of Quantum Biophysics, Sungkyunkwan University, Suwon 16419, Republic of Korea
[5] Department of Applied Statistics, Yonsei University, Seoul 03722, Republic of Korea
[6] Department of Statistics and Data Science, Yonsei University, Seoul 03722, Republic of Korea
[*] Authors to whom any correspondence should be addressed.

E-mail: joonsukhuh@gmail.com and dkd.park@yonsei.ac.kr

Keywords: quantum computing, quantum machine learning, one-class classification, semi-supervised learning, anomaly detection

Abstract

One-class classification (OCC) is a fundamental problem in pattern recognition with a wide range of applications. This work presents a semi-supervised quantum machine learning algorithm for such a problem, which we call a variational quantum one-class classifier (VQOCC). The algorithm is suitable for noisy intermediate-scale quantum computing because the VQOCC trains a fully-parameterized quantum autoencoder with a normal dataset and does not require decoding. The performance of the VQOCC is compared with that of the one-class support vector machine (OC-SVM), the kernel principal component analysis (PCA), and the deep convolutional autoencoder (DCAE) using handwritten digit and Fashion-MNIST datasets. The numerical experiment examined various structures of VQOCC by varying data encoding, the number of parameterized quantum circuit layers, and the size of the latent feature space. The benchmark shows that the classification performance of VQOCC is comparable to that of OC-SVM and PCA, although the number of model parameters grows only logarithmically with the data size. The quantum algorithm outperformed DCAE in most cases under similar training conditions. Therefore, our algorithm constitutes an extremely compact and effective machine learning model for OCC.

1. Introduction

With the growing demand for efficient and effective methods to extract useful knowledge from data, quantum machine learning (QML) has emerged as a promising application of quantum technology [1–3]. Many pattern recognition problems in data science can be formulated as a classification problem, which can be addressed via supervised machine learning (ML). Several theoretical works showed that QML can be advantageous for classification in terms of runtime [4–10], trainability, model capacity [11, 12], generalization [13], and prediction accuracy [14].

While the majority of existing works on QML for classification addresses binary problems, this work focuses on one-class classification (OCC). OCC has a wide range of applications, such as anomaly detection in finance [15], bioinformatics [16], manufacturing [17], computer vision [18], and high energy physics [19]. The goal of OCC is to train an ML model that distinguishes normal data from anomalous ones. In OCC, instead of having input–output example pairs as in the usual setup for supervised learning, only the input information is provided. Since the training example does not contain the class labels, the OCC is often called semi-supervised learning and is more difficult than the binary or multinomial classification with the label information. Moreover, a multinomial classifier can be constructed with multiple one-class classifiers.

OCC problems have been tackled by statistical ML approaches, such as principal component analysis (PCA) [20], one-class support vector machine (OC-SVM) [21–23], and deep learning based algorithms

[24, 25]. In particular, an autoencoder, a feed-forward neural network that aims to copy its input to its output [26–28], is widely used in OCC. An autoencoder consists of an encoder, which extracts the essential feature of data and reduces dimension, and a decoder, which reconstructs the data. Given a training dataset, an autoencoder is trained to act as an identity function with respect to the training dataset and the mean squared reconstruction error is subject to minimization. For OCC, after training a neural network as an autoencoder with normal class data, the reconstruction error can be used as a decision function [29, 30]. Alternatively, the autoencoder can be used as a feature extractor of other statistical ML techniques like OC-SVM [31–33].

As a classical autoencoder is able to learn the efficient representation of low dimensional latent space, a quantum autoencoder (QAE) is proposed for efficient quantum data compression. The QAE utilizes a parameterized quantum circuit (PQC) [34], which is central in variational quantum algorithms [35]. In addition to quantum data compression, several applications of QAE have been explored including denoising quantum data [36], quantum error correction [37], quantum error mitigation [38], and quantum metrology [39]. The QAE has also been explored for detecting anomalous phases in the context of quantum Hamiltonian problems [40], and detecting anomalous events at the Large Hadron Collider in high energy physics [41].

Motivated by the success of classical autoencoders for OCC problems, we present a variational quantum one-class classifier (VQOCC) algorithm based on the QAE that applies to classical data. The VQOCC is composed of data encoding, PQC, and quantum measurements for classical post-processing. In the past, anomaly detection algorithms based on the quantum OC-SVM and quantum PCA that could achieve exponential speedup were proposed [6]. However, these quantum algorithms require expensive subroutines, such as the quantum linear solver [42] and matrix exponentiation [5] that are not suitable for noisy intermediate-scale quantum (NISQ) computing [43]. In contrast, training a shallow-depth PQC with a classical optimizer is regarded as a promising approach for near-term QML [44]. This work focuses on taking the NISQ-friendly approach that constructs a variational quantum algorithm for OCC with classical data, and verifying whether a quantum advantage can be attained.

Numerical experiments are performed on handwritten digits and the Fashion-MNIST dataset with open-source Python API Qibo [45] for quantum circuit simulation. The performance of VQOCC is evaluated via the area under a receiver operating characteristic (ROC) curve (AUC), and compared to classical methods including OC-SVM, kernel PCA, and deep convolutional autoencoder (DCAE). We benchmark the performance of VQOCC with various structures of the QAE. The structure of the QAE is determined by selecting data encoding, the number of PQC layers, and the size of the latent feature space. The general result of VQOCC shows comparable performance to the classical methods despite having the number of model parameters grow only logarithmically with the data feature size. Notably, the performance of VQOCC is better than DCAE under similar training conditions.

The remainder of the paper is organized as follows. Section 2 describes the OCC and reviews some of the well-known approaches to the problem. Section 3 explains the QAE, which is the basis of the quantum one-class classifier proposed in this work. Section 4 explains the application of QAE for OCC and constructing different models via modifying the ansatz (i.e. structure of the PQC) and cost functions. Numerical experiments performed using scikit-learn and Qibo with handwritten digits and Fashion-MNIST datasets are explained in section 5. This section also compares the AUC of ROC curves of our algorithm with a one-class SVM, a kernel PCA, and a DCAE. Section 6 provides conclusion and suggestions for future work.

2. One-class classification

Assigning an input data to one of a given set of classes is a canonical problem in pattern recognition and can be formally described as a classification problem. Classification aims to predict the class label of an unseen (test) data $\tilde{\boldsymbol{x}} \in \mathbb{R}^N$, given a labeled (training) dataset

$$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_M, y_M)\} \subset \mathbb{R}^N \times \mathbb{Z}_l,$$

where l is the number of classes. The binary classification problem corresponds to the case when $l = 2$, which aims to distinguish between two specific classes. In contrast, the OCC is a special case of the aforementioned problem when $l = 1$ [25, 46], meaning that only a single class is provided during training. In this case, the training dataset is $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$, which is referred to as normal in the context of anomaly detection, and the goal is to identify whether test data $\tilde{\boldsymbol{x}}$ belongs to the class observed during training. Since anomalous data is not used in training, this is known as semi-supervised learning. It is also possible to perform OCC with unsupervised methods with an unlabeled dataset under the assumption that most of the test dataset is composed of normal data [24, 25].

Given a training dataset of normal class $\mathcal{D}$, a decision function $f(\boldsymbol{x};\mathcal{D})$ is attained from a OCC algorithm, which quantifies how far the input data $\boldsymbol{x}$ is from the training dataset. If the decision function $f(\tilde{\boldsymbol{x}};\mathcal{D})$ is smaller than a threshold value C_{th} (i.e. $f(\tilde{\boldsymbol{x}}) < C_{\mathrm{th}}$), then $\tilde{\boldsymbol{x}}$ is close enough to the normal class to be classified as normal. If $f(\tilde{\boldsymbol{x}}) > C_{\mathrm{th}}$, then the test data is too far from the normal class and is classified as anomalous. If $f(\tilde{\boldsymbol{x}}) = C_{\mathrm{th}}$, then the decision is made at random.

Two well-known statistical approaches for addressing OCC problems are PCA [20] and SVM [21–23]. PCA is a dimensionality reduction technique that projects data $\boldsymbol{x}_i$ into a lower dimensional subspace such that the projections have the largest variances. The projected space provides reconstructed data $\hat{\boldsymbol{x}}_i$. The lower dimensional subspace is determined to minimize the reconstruction error $\sum_i \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2$. Once the lower dimensional subspace is chosen, the reconstruction error $f(\boldsymbol{x}) = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$ can be considered as a decision function for OCC, since it will be small for normal data and large for anomalous one. The kernel trick can be utilized in PCA to include non-linearity [47].

The SVM is a supervised learning model that aims to find a hyperplane that separates two classes of training data with the maximum margin. Thus it is commonly used in binary classification. The SVM can be modified for OCC by finding a maximum-margin hyperplane that separates normal data from the origin. This is known as the one-class SVM (OC-SVM) [21, 22]. The decision function of OC-SVM is

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle - b, \tag{1}$$

where $\boldsymbol{w}$ and b describes the hyperplane and Φ is the feature map. If the decision function is positive (negative), the corresponding test data is classified as normal (anomalous).

Alternatively, the SVM can be modified for OCC by finding the smallest hypersphere that encapsulates normal data. This is known as the support vector data description (SVDD) [23]. After finding the optimal hypersphere, the data located outside of the hypersphere is classified as anomalous. In this case, the decision function can be expressed as

$$f(\boldsymbol{x}) = \|\Phi(\boldsymbol{x}) - \boldsymbol{a}\|^2 - R, \tag{2}$$

where $\boldsymbol{a}$ is a center of the hypersphere, and R is a radius of the hypersphere. Note that when the data is normalized to unit norm, the OC-SVM and SVDD become equivalent [48]. Intuitively, these methods can be understood as a process of learning the boundary for the normal data and identifying the data outside of the boundary to be anomalies.

3. Quantum autoencoder

QAE is the quantum-analog of classical autoencoder, for which a PQC learns to reduce the dimensionality of data [34]. The training is carried out through a classical optimization process; hence, it is the classical-quantum hybrid algorithm. The dimensionality reduction means that a QAE compresses quantum data into a smaller number of qubits than the input qubits. Following the convention used in classical ML, we refer to the set of qubits to which the data is compressed as latent qubits. A QAE is composed of an encoding part and a decoding part as depicted in figure 1(a). The encoding part applies a parameterized unitary gate $U(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a set of trainable parameters, aiming to compress data into latent qubits. Other qubits are discarded after this step (i.e. traced out) and are called trash qubits. The number of trash qubits is denoted by n_t. For example, the QAE circuit in figure 1(a) uses four latent qubits and two trash qubits. The decoder applies $U^\dagger(\boldsymbol{\theta})$ on the latent qubits and a reference state $|0\rangle^{\otimes n_t}$ to reconstruct the initial quantum state. For a QAE to be successful, the parameterized unitary gates for encoding should learn to disentangle latent qubits and trash qubits to put them into a product state. This guarantees the reconstruction of the initial quantum state via decoding with a proper ancillary state. The PQC is trained by minimizing a cost function defined with the quantum state fidelity or Hamming distance between the trash qubit system and the target state $|0\rangle^{\otimes n_t}$ [34, 49, 50]. More details on the cost function used in this work will be described in the next section.

4. Variational quantum one-class classifier

The QAE lays the ground for VQOCC. The structure of a QAE can be simplified if it is applied to a OCC. Namely, only the encoder part of the QAE is needed. In Romero *et al* [34], two cost functions based on the

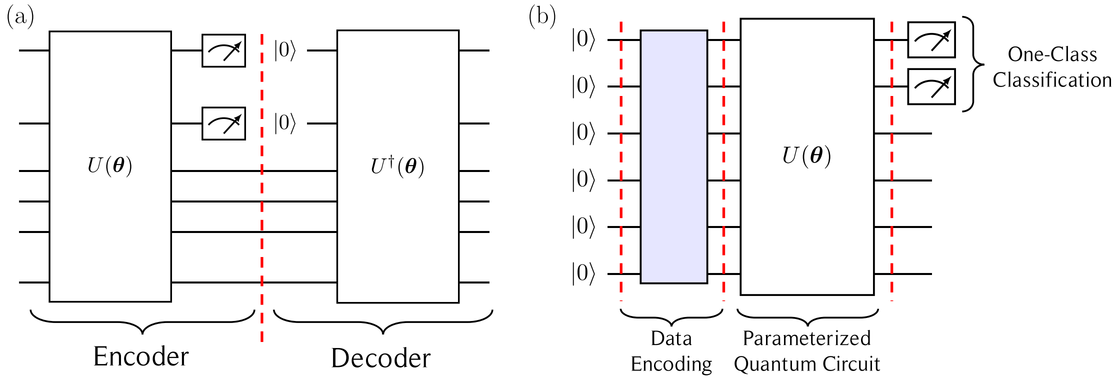


Figure 1. Graphical representation of (a) QAE and (b) VQOCC with number of trash qubits $n_t = 2$ and total qubits $n = 6$. Quantum autoencoder is composed of encoder and decoder parts, which are represented as parameterized quantum circuit $U(\boldsymbol{\theta})$ and $U^\dagger(\boldsymbol{\theta})$, respectively. The VQOCC quantum circuit consists of three parts: data encoding (blue rectangle), parameterized quantum circuit $U(\boldsymbol{\theta})$, and measurement of trash qubits for one-class classification. Note that the parameterized quantum circuit and measurement from the VQOCC quantum circuit is directly taken from the encoder part of quantum autoencoder.

trash state fidelity and the decoded state fidelity were analyzed. It shows that the trash state fidelity is the upper bound of the decoded state fidelity, and when the trash state fidelity equals one, the decoded state fidelity also equals one. The cost function based on the decoded state requires the access of two identical copies of the input state, whereas the cost function based on the trash state does not. Thus formulating the optimization problem with the cost function that only uses the trash state is more advantageous in terms of computational resources.

After training the PQC to minimize the cost function for the normal class training dataset, the cost function for anomalous data is expected to yield values far from zero. Hence by setting a threshold value to the cost function, normal and anomalous data can be discriminated. In this case, the cost function can be understood as a decision function f of OCC, which is analogous to using the reconstruction error as a decision function in the classical autoencoder.

In the following, the essential steps of the QAE-based VQOCC, namely data encoding, parameterized unitary gates, and measurement of n_t qubits from which the cost function is evaluated, are explained in detail. Hereinafter, the structure of quantum gates in a PQC is referred to as ansatz. A pictorial representation of the quantum circuit for VQOCC is shown in figure 1(b).

4.1. Data encoding

To handle classical data, a QML algorithm must be preceded by a procedure that encodes classical data into quantum states [11, 51–53]. The quantum data encoding is a map $\Phi : D \rightarrow W$, where $D \subset \mathbb{R}^{N_c}$ and $W \subset \mathbb{C}^{N_q}$ are the subsets of the real and complex vector spaces in which the classical and quantum data are represented, respectively. The map can be implemented by applying a unitary transformation that is determined by classical data to the initial state $|0\rangle^{\otimes n}$, where $n = \log(N_q)$ is the number of qubits. Typically, n ranges from $\log(N_c)$ to N_c depending on the encoding map Φ [54–56]. In this work, two quantum data encoding schemes are used. One is amplitude encoding, which encodes classical data into the amplitudes of quantum states. Another one is called flexible representation of quantum images (FRQI) [57], which encodes classical data into rotation angles. Under these encoding schemes, $n = O(\log(N_c))$. The latter uses one more qubit than the former, but its resource overhead for the classical pre-computation is significantly smaller. More details on these quantum data encoding methods are given in appendix A.

The ansatz of a QAE serves as a structure for the sequence of quantum gates that are trained to disentangle trash qubits and latent qubits,

$$U(\boldsymbol{\theta})|\psi\rangle \approx |0\rangle^{\otimes n_t} \otimes |\phi\rangle. \tag{3}$$

The ansatz depicted in figure 2, which is adapted from reference [50], is constructed to achieve this. It is composed of layers with parameterized single-qubit y-axis rotations $R_y(\theta_j) = e^{-i\theta_j Y/2}$, followed by controlled-Z (CZ) gates. The CZ gates are applied between trash–trash qubit pairs and latent–trash qubit pairs. Each layer applies a sequence of $CZ_{\text{latent–trash}} \cdot CZ_{\text{trash–trash}} \cdot \bigotimes_{j=1}^{n} R_y(\theta_j)$ multiple times within which a different combination of latent–trash qubit pairing is used. In contrast, there are no CZ gates between latent

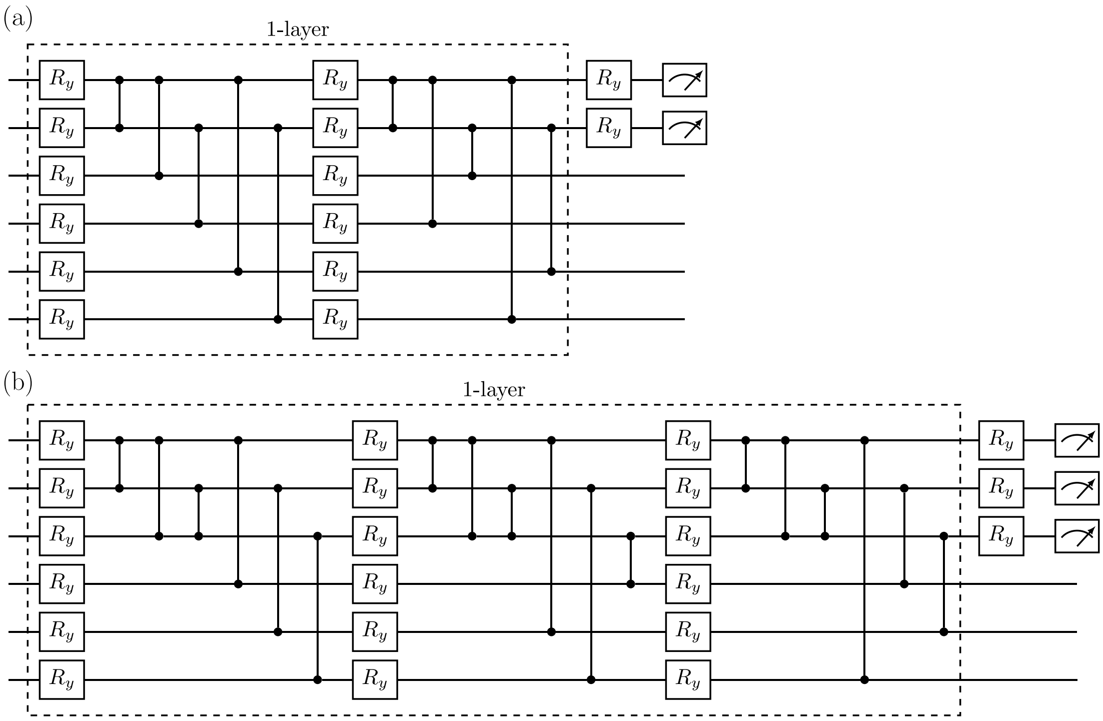


Figure 2. Parameterized quantum circuit ansatz for one layer with number of trash qubits (a) $n_t = 2$, (b) $n_t = 3$ and total qubits $n = 6$. Dashed boxes correspond to the ansatz of one layer that has R_y rotations and *CZ* gates between two trash qubits and between a trash qubit and a latent qubit. *CZ* gates are applied to different combinations of trash–latent qubit pairs after R_y rotations.

qubits since it does not contribute to disentangling trash qubits from latent qubits. For example, in figure 2(a), *CZ* gates are applied between the first latent qubit and either the first or second trash qubit.

In this work, it will be shown that increasing the number of layers can enhance the performance of OCC. However, using more layers increases the number of model parameters and gates and the quantum circuit depth. Evaluating these and studying the tradeoff between classification performance and computational resources is of critical importance for the practical application of VQOCC. The number of parameters (p) and two-qubit gates (g_2), and the circuit depth (d) used for VQOCC are

$$p = n_t(nL + 1), \tag{4}$$

$$g_2 = \left(\frac{n_t^3 - 3n_t^2}{2} + nn_t\right)L, \tag{5}$$

and

$$d = 1 + \left(\frac{n_t^3 - 3n_t^2}{2} + nn_t + n_t\right)L, \tag{6}$$

respectively, where L is the number of layers.

Because $n = O(\log(N_c))$ when amplitude encoding or FRQI encoding is utilized, the number of optimization parameters shown in equation (4) increases logarithmically with the size of data, which is in stark contrast to a large number of parameters required in classical deep learning algorithms. A potential quantum advantage is rooted in the fact that VQOCC can learn to discriminate anomalous data using only a logarithmic number of model parameters with respect to the number of features that describe the data.

4.2. Cost function

The choice of cost function is critical in training PQCs. As described in section 3, minimizing the cost function for a QAE is equivalent to making the trash qubit state as close as possible to the reference state $|0\rangle^{\otimes n_t}$. PQCs are trained such that only normal data can provide the reference state $|0\rangle^{\otimes n_t}$ on trash qubits, whereas anomalous data will provide different states on the trash qubits. In the early developments of QAE [34], the quantum state fidelity between measured trash qubit state and $|0\rangle^{\otimes n_t}$ was chosen as a cost function. However, reference [49] showed that cost functions with global observables induce exponentially vanishing gradients, so called barren plateaus [58], even in shallow quantum circuits. Consequently, the fidelity-based cost function used in the initial work is subject to the unwanted barren plateau effect. One way to avoid this is to construct a cost function with local observables [49]. The localized cost function used in this paper is based on the Hamming distance between the measurement outcome in the computational basis and the reference state $|0\rangle^{\otimes n_t}$ [50]. More specifically, the local cost function based on the Hamming distance can be written as

$$C = \frac{1}{2}\sum_{j=1}^{n_t}(1 - \langle Z_j\rangle), \tag{7}$$

where $\langle Z_j\rangle$ is an expectation value of the Pauli-Z operator for the jth trash qubit. This cost function is zero when the trash qubit state is disentangled from latent qubits and equals to the reference state $|0\rangle^{\otimes n_t}$, which indicates the compression of the quantum state to latent qubits.

While keeping the local property, a cost function can be constructed in a different manner. For instance, a cost function can be formulated in terms of a log loss function as

$$C = \sum_{j=1}^{n_t}\log\left(\frac{1 - \langle Z_j\rangle}{2}\right). \tag{8}$$

5. Numerical experiments

This section reports the benchmarking results for VQOCC, obtained by classical simulations carried out with open source framework Qibo [45]. The simulation is performed with two datasets, an 8 × 8 handwritten digits dataset available in scikit-learn [59], and the Fashion-MNIST dataset [60] downsampled to 16 × 16 pixels. Both datasets have ten different classes. Samples of the datasets are depicted in appendix D. In each numerical simulation, one class is treated as the normal class, and the training sample size is 100. After training, both normal and anomalous data samples are used as a test dataset. An equal number of samples from each class is used as test samples. A test dataset is composed of 70 and 100 test samples from each class, and consequently, the total number of test datasets used in the experiment is 700 and 1000 for handwritten digits and Fashion-MNIST datasets, respectively.

Besides the datasets, there are three variables in the numerical experiments: quantum data encoding methods, the number of trash qubits denoted by n_t, and the number of PQC layers denoted by L. Amplitude encoding and FRQI encoding are used for quantum data encoding, as described before. For amplitude encoding, 6-qubit states and 8-qubit states are used for handwritten digits and Fashion-MNIST datasets, respectively. For FRQI encoding, 7-qubit states and 9-qubit states are used for handwritten digits and Fashion-MNIST datasets, respectively. Three different numbers of trash qubits, $n_t = 2, 3, 4$, are used in the simulation. For each n_t, different values of L are used. More specifically, $L \in \{2, 4, 8, 12, 16\}$ for $n_t = 2$, $L \in \{2, 4, 6, 8, 10\}$ for $n_t = 3$, and $L \in \{2, 4, 6, 8\}$ for $n_t = 4$. The maximum L for each n_t is chosen in a way that the maximum circuit depths for all n_t are approximately the same for a fair comparison among different n_t. The evaluation metric used for numerical experiments is the AUC, which is commonly used for one-class classifiers [25].

The optimization of the PQC was performed with the Adam optimizer [61] in Tensorflow [62]. Tensorflow was set to be the simulation backend of Qibo, which enables the automatic differentiation for the computation of gradients. We used mini-batch gradient descent with a batch size of 10. The learning rate was set to be 0.1, and the number of iterations was 150.

The results presented in this section are based on minimizing the Hamming distance cost function shown in equation (7). Experiments based on the logarithmic cost function shown in equation (8) tested on the handwritten digit and the Fashion-MNIST datasets with amplitude encoding produced similar results that are reported in appendix C.

We compared the performance of VQOCC with various classical methods, such as the kernel PCA, OC-SVM, and deep neural network. For kernel PCA and OC-SVM, the Gaussian radial basis function kernel was used. We selected the inverse length parameter γ from $\gamma \in \{2^{-10}, 2^{-9}, \ldots, 2^{-1}\}$ via grid search using the performance on a small holdout set (10% of randomly drawn normal and anomalous test samples), and for

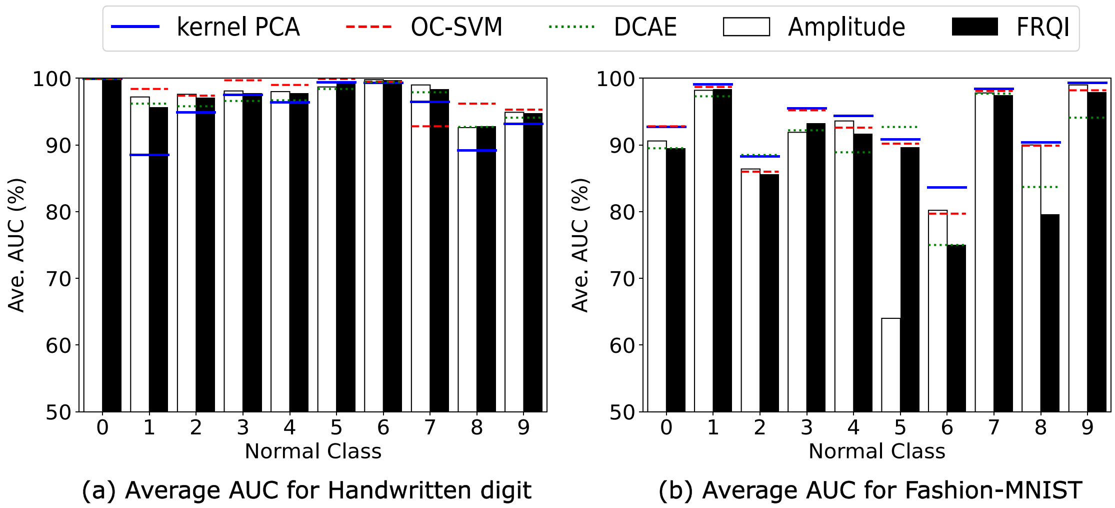


Figure 3. Average AUCs (over 10 seeds) of VQOCC on handwritten digit and Fashion-MNIST dataset for ten different normal classes. We report the best results from different numbers of trash qubits n_t and layers L of parameterized quantum circuits with two quantum data encoding schemes, amplitude encoding (open) and FRQI (filled). The results of kernel PCA (blue solid), OC-SVM (red dashed), and DCAE (green dotted) are illustrated as baselines.

OC-SVM, we chose the hyperparameter $\nu \in \{0.01, 0.1\}$, which represents the lower (upper) bound for the fraction of support vectors (errors), and report the better result.

As a deep learning method, a DCAE [32, 63] was chosen as a baseline. The size of the deep architecture was constrained so that the number of model parameters is comparable to that of VQOCC for a fair comparison. A detailed description of the DCAE architecture is described in appendix E. The number of parameters used in DCAE is 158 and 254 for the handwritten digit and the Fashion-MNIST datasets, respectively.

5.1. Results overview

The main results are presented in figure 3 and table 1. Each value in the figure and table represents the result obtained with the best n_t and L. In figure 3, two data encoding methods, amplitude encoding and FRQI encoding, are shown as open and filled bars, respectively. The results obtained from kernel PCA and OC-SVM, and DCAE are displayed as blue solid, red dashed, and green dotted lines, respectively. In table 1, the average and best AUCs of the OCC on handwritten digits and Fashion-MNIST datasets are provided. In general, the results from both encoding methods are on par with the classical algorithms and show better performances than DCAE. The cases from the class 2, 6, and 7 of the handwritten digit dataset even show that the performances from the quantum algorithm outperform the performances from all three classical algorithms. Also note that the best AUCs from VQOCC with either of the encoding methods at the class 0, 1, 2, 4, 6, 7, and 9 of handwritten digit and the class 8 and 9 of Fashion-MNIST show better performances than the AUCs from both OC-SVM and kernel PCA. The best AUCs from DCAE are generally better than the best AUCs from VQOCC in the Fashion-MNIST dataset, but the standard deviations of DCAE are larger than that of VQOCC. The two quantum data encoding methods performed comparably, except one case at the class 5 of Fashion-MNIST with the significantly lower performance of 64.0%, due to the data normalization in amplitude encoding (appendix F). Note that even though FRQI encoding requires one more qubit than amplitude encoding, quantum state preparation of image data is simpler with FRQI encoding (see appendix A).

5.2. Effects of the QAE structure

The performances of VQOCC are evaluated for various QAE structures determined by different values of n_t and L. Figure 4 illustrates the average AUCs of VQOCC for various n_t and L with standard deviations as colored shade on the handwritten digit and Fashion-MNIST datasets. We chose a few classes to illustrate the dependencies. Class 1 and 5 of handwritten digits and class 2 and 5 of Fashion-MNIST were chosen with both encoding schemes. The circuit depths at each L are also shown at the top of each plot. The general trend shows that the average AUC (i.e. the classification performance) increases with L. Furthermore, not only does

Table 1. The average and best AUCs in % and one standard deviation (over 10 seeds) of the one-class classification on handwritten digit and Fashion-MNIST dataset with variational quantum one-class classifier (VQOCC) with both amplitude and FRQI encoding, and deep convolutional autoencoder (DCAE). For VQOCC results, the better results are reported while varying the number of trash qubits n_t and layers of parameterized quantum circuits L. The results from kernel PCA and OC-SVM are also shown as a competing method.

		AUC (%)							
		Amplitude		FRQI		DCAE			
Dataset	Normal class	Average	Best	Average	Best	Average	Best	Kernel PCA	OC-SVM
Handwritten digit	0	99.9 ± 0.1	100.0	99.9 ± 0.1	100.0	99.9 ± 0.1	100.0	99.9	99.9
	1	97.2 ± 0.8	98.7	95.7 ± 0.9	97.0	96.2 ± 1.2	97.9	88.5	98.4
	2	97.6 ± 0.9	99.0	97.1 ± 0.9	98.7	95.8 ± 2.3	99.4	94.9	97.4
	3	98.1 ± 0.3	98.7	97.8 ± 0.5	98.4	96.6 ± 1.1	99.5	97.5	99.7
	4	98.0 ± 0.6	99.2	97.8 ± 0.2	98.1	96.7 ± 1.1	99.0	96.4	99.0
	5	98.7 ± 0.5	99.5	99.3 ± 0.3	99.6	98.4 ± 1.1	99.8	99.4	99.9
	6	99.8 ± 0.1	100.0	99.7 ± 0.1	99.9	99.4 ± 0.3	100.0	99.3	99.5
	7	99.0 ± 0.5	99.7	98.4 ± 0.9	99.3	97.9 ± 1.0	99.4	96.5	92.8
	8	92.6 ± 1.8	95.2	92.8 ± 2.2	96.0	92.7 ± 2.1	96.7	89.2	96.2
	9	94.9 ± 1.3	96.5	94.8 ± 0.5	95.2	94.1 ± 1.5	96.6	93.2	95.3
Fashion-MNIST	0	90.6 ± 0.6	91.4	89.5 ± 1.1	90.4	89.5 ± 4.1	96.1	92.7	92.8
	1	98.2 ± 0.2	98.7	98.4 ± 0.1	98.8	97.3 ± 1.1	99.4	99.1	98.7
	2	86.4 ± 0.6	87.6	85.6 ± 0.6	87.3	88.5 ± 3.2	95.7	88.3	86.0
	3	91.9 ± 1.0	93.7	93.3 ± 0.6	93.9	92.2 ± 1.3	95.0	95.5	95.2
	4	93.6 ± 0.2	93.9	91.7 ± 1.4	92.7	88.9 ± 4.2	97.6	94.4	92.6
	5	64.0 ± 2.5	70.1	89.7 ± 0.5	90.4	92.7 ± 1.4	94.4	90.8	90.2
	6	80.2 ± 0.4	80.8	75.0 ± 2.0	77.5	75.0 ± 3.7	96.3	83.6	79.7
	7	97.8 ± 0.3	98.2	97.6 ± 0.1	97.9	97.5 ± 0.4	98.8	98.4	98.1
	8	90.0 ± 1.0	91.4	79.6 ± 3.3	84.0	83.7 ± 3.9	94.5	90.4	89.9
	9	99.0 ± 0.3	99.4	97.9 ± 0.6	98.6	94.1 ± 4.5	98.9	99.3	98.2

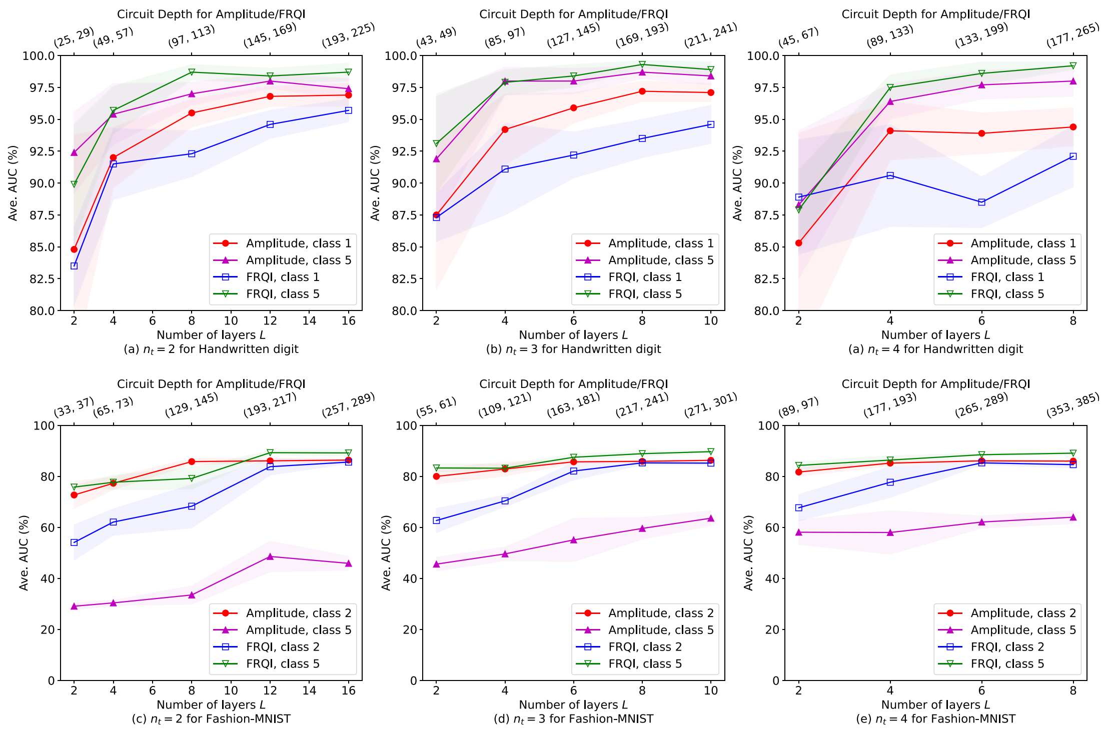


Figure 4. Average AUCs and their standard deviations of VQOCC varying the number of trash qubits $n_t = 2, 3, 4$ and layers L on handwritten digits (a), (b), (c) and Fashion-MNIST (d), (e), (f) dataset. Shaded areas indicate the standard deviations. For handwritten digits, the results from class 1 and 5 are shown, and for Fashion-MNIST, the results form class 2 and 5 are shown as representative examples. The plots include results for two different encoding schemes, amplitude encoding and FRQI encoding. The circuit depths at a number of layers L for different encoding schemes are indicated on the top of each plot.

the performance increase but also the standard deviation of AUCs decreases as L increases. This implies that the PQC with low depth is more likely to fall into a local minimum, which impedes the classification performance.

The results in figure 4 show that the performance saturates as L increases. In general, an average AUC increases with L, but in some cases, the performance hits a plateau when L is beyond a certain threshold L_{th}. For instance, L_{th} of class 5 of the handwritten digit dataset is $L_{\text{th}} = 8$ and $L_{\text{th}} = 4$ for $n_t = 2$ and 3, respectively, for both encoding schemes, whereas in the case of $n_t = 4$, the performance plateau was not observed in class 5 of the handwritten digit dataset. Such behaviors are also observed at the class 1 of the handwritten digit dataset for Amplitude encoding at $n_t = 4$, the class 5 of Fashion-MNIST dataset for both encoding schemes at $n_t = 2$, and the class 2 of Fashion-MNIST dataset for both encoding schemes at $n_t = 2, 3$, and 4. Recall that the number of model parameters increases linearly with L. This observation shows the relevance of the over-parameterization in variational QML [64].

6. Conclusion

We proposed a semi-supervised variational QML algorithm for OCC of classical data. In general, OCC problems are considered to be more difficult than traditional binary or multi-class classification problems due to the absence of labels in the dataset. Consequently, it requires a different approach. Our algorithm, dubbed VQOCC, utilizes a fully-parameterized QAE that learns to extract essential features of normal data. Unlike the conventional QAE, VQOCC only uses the encoding part, which is trained to recognize anomalous data by minimizing a loss function defined with trash qubits left out from the data compression. Since the algorithm is based on training a fully PQC and only uses half the circuit of the usual QAE, it is an excellent candidate for the NISQ application.

We explored the performance of the VQOCC algorithm for the handwritten digits and Fashion-MNIST datasets by varying the classifier structure, which includes the data encoding scheme, the number of layers in the ansatz, and the number of measured trash qubits. After tailoring the structure of the VQOCC algorithm, the performance was compared to classical OCC methods including OC-SVM, kernel PCA, and DCAE. The number of parameters in DCAE was matched to that in PQCs for a fair comparison. VQOCC generally performed better than DCAE and was comparable to OC-SVM and kernel PCA. However, it is unsure whether increasing the number of parameters in PQCs by having more layers will constantly improve the performance to significantly surpass classical methods. Numerical experiments show that increasing the number of circuit layers (and hence the number of model parameters) is deemed to be effective in enhancing performance until it hits the plateau when a certain value is reached.

Note that statistical methods like OC-SVM and kernel PCA are known to be less effective at complex high-dimensional datasets [65] and deep learning algorithms are shown to be very effective at this problem. VQOCC proposed in this work can be an alternative model of deep learning algorithms and its classification capabilities for complex high-dimensional datasets are left for future investigation. Since the number of qubits for encoding classical data can be exponentially smaller than the number of features, the model parameters in VQOCC can be exponentially smaller than classical deep learning algorithms. Moreover, OC-SVM and kernel PCA require at least $O(d)$ computational complexity for a d-dimensional dataset, in stark contrast to $O(\log d)$ number of qubits and model parameters required in VQOCC.

Interesting future work is to apply different ansatz of PQCs for VQOCC. One particular example of ansatz is quantum circuits with a hierarchical structure [8]. Previously, the hierarchical structure was successful for binary classification [8, 14], and hence it is natural to extend it to OCC. Another promising direction could be a combination of a QAE and OC-SVM, possibly with quantum kernels. A similar approach exists in classical ML in which an autoencoder is used as a feature extractor [31] and OC-SVM is used for anomaly detection on the compressed data. Appendix F shows that the classification results of OC-SVM and kernel PCA are affected under the data normalization, which is required in amplitude encoding. This implies that the VQOCC can be further improved by carefully choosing the data encoding scheme which must be preceded by classical data preprocessing. It is worth mentioning that our approach can be extended to the unsupervised learning of OCC. In the classical unsupervised setup, most of the data is assumed to be comprised of normal data and anomalous data will show a high false positive rate [24, 32]. This assumption can be directly applied to our VQOCC algorithm. Finally, an important future work, which is common in many variational quantum algorithms, is to verify whether the improvements continue to hold in larger systems.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgment

This research was supported by the Yonsei University Research Fund of 2022 (2022-22-0124), by the National Research Foundation of Korea (Grant Nos. 2021M3H3A1038085, 2019M3E4A1079666, 2022M3E4A1074591, and 2022M3H3A106307411), and by the KIST Institutional Program (2E31531-22-076).

Appendix A. Data encoding

A.1. Amplitude encoding

Amplitude encoding loads classical data into the amplitudes of a quantum state. To encode a $2^n \times 2^n$ pixel image into a quantum state, the input can be represented as $x = (x_1, x_2, \ldots, x_N)^\top$, where $N = 2^{2n}$, and it can be encoded in a $2n$-qubit quantum state $\phi(x)$ as follows,

$$|\phi(x)\rangle = \frac{1}{||x||}\sum_{i=1}^{N} x_i |i\rangle. \tag{A1}$$

A.2. FRQI: flexible representation of quantum images

The FRQI [57] is designed to encode classical image data into quantum states. Unlike amplitude encoding, which requires nonintuitive routine to prepare arbitrary amplitudes [66], FRQI encoding is composed of Hadamard and controlled rotation with rotation angles simply achieved from input image. When input is given by $x = (x_1, x_2, \ldots, x_N)^\top$, each pixels can be normalized to be in the range of $\theta_i \in [0, \frac{\pi}{2}]$, or $\theta = (\theta_1, \theta_2, \ldots, \theta_N)^\top$. These angles can be encoded in $(2n+1)$-qubit quantum state $I(\theta)$ as follows,

$$|I(\theta)\rangle = \frac{1}{2^n}\sum_{i=0}^{2^{2n}-1}(\cos(\theta_i)|0\rangle + \sin(\theta_i)|1\rangle)\otimes|i\rangle. \tag{A2}$$

It requires one more qubit to encode classical data, but it is more intuitive and flexible on encoding classical image data into quantum states.

Appendix B. Evaluation metrics

The performance of one-class classifier is evaluated based on how well normal and anomalous data in a test dataset is classified as normal and anomalous, respectively. Such classification depends on a threshold value, C_{th}, for the decision function. The ROC curve characterizes the performance of OCC at different threshold values. The ROC curve plots the relationship between the true positive rate (TPR) and the false positive rate (FPR). By labeling normal (anomalous) data as positive (negative), respectively, they are defined as follows,

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{B1}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}, \tag{B2}$$

where TP, TN, FP, and FN refer to the number of data that is true positive, true negative, false positive, and false negative, respectively.

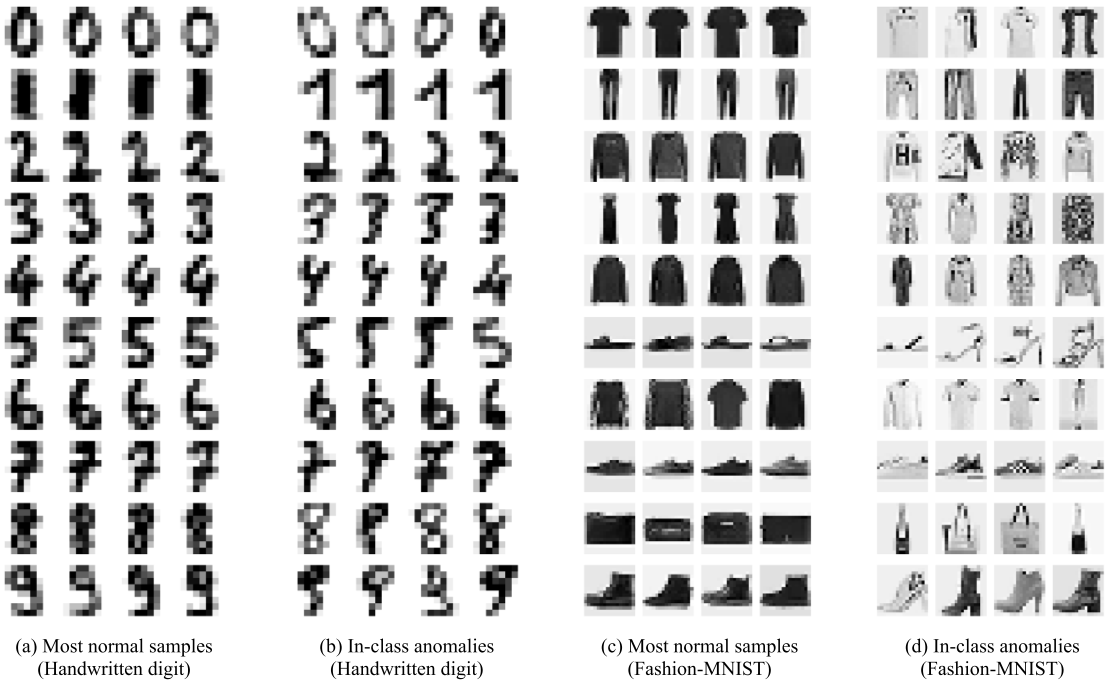
The AUC is commonly used evaluation metrics for one-class classifiers [25]. If the classifier can ideally separate the normal and anomalous data, the AUC is given by 1.0. In contrast, in the worst case when the data is classified at random, the AUC is given by 0.5. Furthermore, AUC can represent a performance of one-class classifier independently to a threshold value of a decision function.

Appendix C. Logarithmic cost function

In section 4 of the main text, the local cost function was introduced that it can avoid barren plateaus. The cost function mainly used in the main text is the Hamming distance based cost function (7), but the local cost function can also be constructed in terms of logarithmic function, as (8). We here report the AUC result for the handwritten digit and Fashion-MNIST dataset with amplitude encoding and logarithmic cost function in table 2. The result for logarithmic cost does not show a big difference to the result from Hamming distance based cost function.

Table 2. Average AUCs in % and one standard deviation (over 10 seeds) of the one-class classification on the handwritten digit dataset with logarithmic cost function.

		Average AUC (%)		
Dataset	Normal class	Amplitude	Kernel PCA	OC-SVM
Handwritten digit	0	99.9 ± 0.1	99.9	99.9
	1	97.7 ± 0.7	88.5	98.4
	2	96.5 ± 1.2	94.9	97.4
	3	98.0 ± 0.8	97.5	99.7
	4	97.7 ± 0.3	96.4	99.0
	5	98.8 ± 0.6	99.4	99.9
	6	99.8 ± 0.1	99.3	99.5
	7	98.9 ± 0.5	96.5	92.8
	8	92.5 ± 1.3	89.2	96.2
	9	93.7 ± 2.2	93.2	95.3
Fashion-MNIST	0	90.9 ± 0.3	89.0	87.2
	1	98.2 ± 0.1	98.5	91.3
	2	86.8 ± 0.4	86.2	87.8
	3	92.4 ± 0.5	91.0	91.9
	4	92.9 ± 0.3	93.6	93.6
	5	65.0 ± 3.5	75.3	43.2
	6	79.9 ± 0.5	79.9	80.4
	7	97.9 ± 0.2	98.1	97.0
	8	89.8 ± 2.0	89.4	80.8
	9	99.0 ± 0.3	98.9	98.3



(a) Most normal samples (Handwritten digit)

(b) In-class anomalies (Handwritten digit)

(c) Most normal samples (Fashion-MNIST)

(d) In-class anomalies (Fashion-MNIST)

Figure 5. Most normal (a), (c) and most anomalous in-class (b), (d) samples in the handwritten digit and Fashion-MNIST dataset, respectively, classified by variational quantum one-class classifier (VQOCC).

Appendix D. Data samples

Figures 5(a)–(d) show the most normal and the most anomalous in-class samples in the handwritten digit data, and the most normal and the most anomalous in-class samples in the Fashion-MNIST data, respectively, determined by the cost function of the VQOCC algorithm.

Appendix E. Deep convolutional autoencoder (DCAE)

The deep learning architecture for DCAE is determined to match the number of parameter in deep learning model to be comparable to the number of parameter used in VQOCC quantum circuit model for fair comparison. The DCAE model is composed of encoder and decoder. For the encoder part of DCAE, we use LeNet-type convolutional neural networks (CNNs), where each convolutional layers are followed by leaky ReLu activation functions and 2×2 max-pooling. The decoder has a symmetrical structure to the encoder, where the max-pooling is substituted by upsampling. The encoder architecture is consisted of $2 \times (3 \times 3 \times 1)$ filters followed by $2 \times (3 \times 3 \times 1)$ filters, and a final dense layer of 4 units. For optimization, the Adam optimizer is used and the Batch Normalization is applied as [32]. We train the model with the learning rate η from $\eta \in \{1.0 \times 10^{-4}, 5.0 \times 10^{-4}, 1.0 \times 10^{-3}, 5.0 \times 10^{-3}\}$, the weight decay hyperparameter λ from $\lambda \in \{10^{-6}, 10^{-5}\}$, batch size of 10, and 250 epochs, and report the better results.

Appendix F. Normalized data

Amplitude encoding encodes data into the amplitude of quantum states, which naturally leads data to be normalized. In this section, we compare the VQOCC result from amplitude encoding to the result from kernel PCA and OC-SVM with the normalized input data in table 3. A few cases show a decrease in performance after normalization, such as the third and fifth class of the Fashion-MNIST dataset. This result implies that the normalization process that is implicitly included in the amplitude encoding can deteriorate the classification performance.

Table 3. Average AUCs in % and one standard deviation (over 10 seeds) of the one-class classification on the handwritten digit and Fashion-MNIST dataset under normalization.

		Average AUC (%)		
Dataset	Normal class	Amplitude	Kernel PCA	OC-SVM
Handwritten digit	0	99.9 ± 0.1	99.9	99.5
	1	97.2 ± 0.8	98.6	80.0
	2	97.6 ± 0.9	99.3	95.4
	3	98.1 ± 0.3	98.5	97.7
	4	98.0 ± 0.6	97.0	93.9
	5	98.7 ± 0.5	99.7	95.3
	6	99.8 ± 0.1	99.1	99.0
	7	99.0 ± 0.5	95.5	98.7
	8	92.6 ± 1.8	95.2	81.8
	9	94.9 ± 1.3	97.7	87.9
Fashion-MNIST	0	90.6 ± 0.6	89.0	87.2
	1	98.2 ± 0.2	98.5	91.3
	2	86.4 ± 0.6	86.2	87.8
	3	91.9 ± 1.0	91.0	91.9
	4	93.6 ± 0.2	93.6	93.6
	5	64.0 ± 2.5	75.3	43.2
	6	80.2 ± 0.4	79.9	80.4
	7	97.8 ± 0.3	98.1	97.0
	8	90.0 ± 1.0	89.4	80.8
	9	99.0 ± 0.3	98.9	98.3

ORCID iDs

Gunhee Park https://orcid.org/0000-0002-2818-488X
Joonsuk Huh https://orcid.org/0000-0002-8792-5641
Daniel K Park https://orcid.org/0000-0002-3177-4143

References

[1] Wittek P 2014 *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Boston: Academic)

[2] Schuld M and Petruccione F 2021 *Machine Learning With Quantum Computers* (*Quantum Science and Technology*) (Cham: Springer)

[3] Xiao T, Fan J and Zeng G 2022 Parameter estimation in quantum sensing based on deep reinforcement learning *npj Quantum Inf.* **8** 2

[4] Rebentrost P, Mohseni M and Lloyd S 2014 Quantum support vector machine for big data classification *Phys. Rev. Lett.* **113** 130503

[5] Lloyd S, Mohseni M and Rebentrost P 2014 Quantum principal component analysis *Nat. Phys.* **10** 631–3

[6] Liu N and Rebentrost P 2018 Quantum machine learning for quantum anomaly detection *Phys. Rev.* A **97** 042315

[7] Schuld M, Sinayskiy I and Petruccione F 2016 Prediction by linear regression on a quantum computer *Phys. Rev.* A **94** 022342

[8] Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, Green A G and Severini S 2018 Hierarchical quantum classifiers *npj Quantum Inf.* **4** 65

[9] Cong I, Choi S and Lukin M D 2019 Quantum convolutional neural networks *Nat. Phys.* **15** 1273–8

[10] Liu Y, Arunachalam S and Temme K 2021 A rigorous and robust quantum speed-up in supervised machine learning *Nat. Phys.* **17** 1013–7

[11] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12

[12] Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A and Woerner S 2021 The power of quantum neural networks *Nat. Comput. Sci.* **1** 403–9

[13] Caro M C, Huang H-Y, Cerezo M, Sharma K, Sornborger A, Cincio L and Coles P J 2022 Generalization in quantum machine learning from few training data *Nat. Commun.* **13** 4919

[14] Hur T, Kim L and Park D K 2022 Quantum convolutional neural network for classical data classification *Quantum Mach. Intell.* **4** 3

[15] Li S-H, Yen D C, Lu W-H and Wang C 2012 Identifying the signs of fraudulent accounts using data mining techniques *Comput. Hum. Behav.* **28** 1002–13

[16] Min S, Lee B and Yoon S 2016 Deep learning in bioinformatics *Briefings Bioinform.* **18** 851–69

[17] Martí L, Sanchez-Pi N, Molina Je M and Garcia A C B 2015 Anomaly detection based on sensor data in petroleum industry applications *Sensors* **15** 2774–97

[18] Saleh B, Farhadi A and Elgammal A 2013 Object-centric anomaly detection by attribute-based reasoning *2013 IEEE Conf. on Computer Vision and Pattern Recognition* pp 787–94

[19] Fraser K, Homiller S, Mishra R K, Ostdiek B and Schwartz M D 2022 Challenges for unsupervised anomaly detection in particle physics *J. High Energy Phys.* **2022** 66

[20] Hoffmann H 2007 Kernel PCA for novelty detection *Pattern Recognit.* **40** 863–74

[21] Schölkopf B, Williamson R C, Smola A, Shawe-Taylor J and Platt J 2000 Support vector method for novelty detection *Advances in Neural Information Processing Systems* vol 12, ed S Solla, T Leen and K Müller (MIT Press)

[22] Schölkopf B, Platt J C, Shawe-Taylor J, Smola A J and Williamson R C 2001 Estimating the support of a high-dimensional distribution *Neural Comput.* **13** 1443–71

[23] Tax D M J and Duin R P W 2004 Support vector data description *Mach. Learn.* **54** 45–66

[24] Chalapathy R and Chawla S 2019 Deep learning for anomaly detection: a survey (arXiv:1901.03407)

[25] Perera P, Oza P and Patel V M 2021 One-class classification: a survey (arXiv:2101.03064)

[26] Bourlard H and Kamp Y 1988 Auto-association by multilayer perceptrons and singular value decomposition *Biol. Cybern.* **59** 291–4

[27] Kramer M A 1991 Nonlinear principal component analysis using autoassociative neural networks *AIChE J.* **37** 233–43

[28] Hinton G E and Zemel R 1993 Autoencoders, minimum description length and Helmholtz free energy *Advances in Neural Information Processing Systems* vol 6, ed J Cowan, G Tesauro and J Alspector (Morgan-Kaufmann)

[29] Sakurada M and Yairi T 2014 Anomaly detection using autoencoders with nonlinear dimensionality reduction *Proc. of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis* (*MLSDA 2014*) (*New York, NY, USA*) (Association for Computing Machinery) pp 4–11

[30] Chen J, Sathe S, Aggarwal C and Turaga D 2017 Outlier detection with autoencoder ensembles *SIAM Int. Conf. on Data Mining* pp 90–98

[31] Erfani S M, Rajasegarar S, Karunasekera S and Leckie C 2016 High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning *Pattern Recognit.* **58** 121–34

[32] Ruff L, Vandermeulen R, Goernitz N, Deecke L, Siddiqui S A, Binder A, Müller E and Kloft M 2018 Deep one-class classification *Proc. 35th Int. Conf. on Machine Learning* (*Proc. Machine Learning Research*) (*10–15*) vol 80, ed J Dy and A Krause (PMLR) pp 4393–402

[33] Chalapathy R, Menon A K and Chawla S 2018 Anomaly detection using one-class neural networks (arXiv:1802.06360)

[34] Romero J, Olson J P and Aspuru-Guzik A 2017 Quantum autoencoders for efficient compression of quantum data *Quantum Sci. Technol.* **2** 045001

[35] Cerezo M *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 625–44

[36] Bondarenko D and Feldmann P 2020 Quantum autoencoders to denoise quantum data *Phys. Rev. Lett.* **124** 130502

[37] Locher D F, Cardarelli L and Müller M 2022 Quantum error correction with quantum autoencoders (arXiv:2202.00555)

[38] Zhang X-M, Kong W, Farooq M U, Yung M-H, Guo G and Wang X 2021 Generic detection-based error mitigation using quantum autoencoders *Phys. Rev.* A **103** L040403

[39] Yuxuan D and Tao D 2021 On exploring practical potentials of quantum auto-encoder with advantages (arXiv:2106.15432)

[40] Kottmann K, Metz F, Fraxanet J and Baldelli Nò 2021 Variational quantum anomaly detection: unsupervised mapping of phase diagrams on a physical quantum computer *Phys. Rev. Res.* **3** 043184

[41] Ngairangbam V S, Spannowsky M and Takeuchi M 2022 Anomaly detection in high-energy physics using a quantum autoencoder *Phys. Rev.* D **105** 095004

[42] Harrow A W, Hassidim A and Lloyd S 2009 Quantum algorithm for linear systems of equations *Phys. Rev. Lett.* **103** 150502

[43] Preskill J 2018 Quantum computing in the NISQ era and beyond *Quantum* **2** 79

[44] Bharti K *et al* 2022 Noisy intermediate-scale quantum algorithms *Rev. Mod. Phys.* **94** 015004

[45] Efthymiou S, Ramos-Calderer S, Bravo-Prieto C, Pérez-Salinas Aan, García-Martín D, Garcia-Saez A, Latorre Je I and Carrazza S 2021 Qibo: a framework for quantum simulation with hardware acceleration *Quantum Sci. Technol.* **7** 015018

[46] Moya M M and Hush D R 1996 Network constraints and multi-objective optimization for one-class classification *Neural Netw.* **9** 463–74

[47] Schölkopf B, Smola A and Müller K-R 1998 Nonlinear component analysis as a kernel eigenvalue problem *Neural Comput.* **10** 1299–319

[48] Tax D M J 2001 One-class classification: concept learning in the absence of counter-examples *PhD Thesis* Technische Universiteit Delft

[49] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2021 Cost function dependent barren plateaus in shallow parametrized quantum circuits *Nat. Commun.* **12** 1791

[50] Bravo-Prieto C 2021 Quantum autoencoders with enhanced data encoding *Mach. Learn.: Sci. Technol.* **2** 035028

[51] Giovannetti V, Lloyd S and Maccone L 2008 Quantum random access memory *Phys. Rev. Lett.* **100** 160501

[52] Schuld M and Killoran N 2019 Quantum machine learning in feature Hilbert spaces *Phys. Rev. Lett.* **122** 040504

[53] Lloyd S, Schuld M, Ijaz A, Izaac J and Killoran N 2020 Quantum embeddings for machine learning (arXiv:2001.03622)

[54] de Veras T M L, de Araujo I C S, Park D K and da Silva A J 2021 Circuit-based quantum random access memory for classical data with continuous amplitudes *IEEE Trans. Comput.* **70** 2125–35

[55] Araujo I F, Park D K, Petruccione F and da Silva A J 2021 A divide-and-conquer algorithm for quantum state preparation *Sci. Rep.* **11** 6329

[56] Araujo I F, Park D K, Ludermir T B, Oliveira W R, Petruccione F and da Silva A J 2021 Configurable sublinear circuits for quantum state preparation (arXiv:2108.10182)

[57] Le P Q, Dong F and Hirota K 2011 A flexible representation of quantum images for polynomial preparation, image compression and processing operations *Quantum Inf. Process.* **10** 63–84

[58] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 4812

[59] Pedregosa F *et al* 2011 Scikit-learn: machine learning in python *J. Mach. Learn. Res.* **12** 2825–30

[60] Xiao H, Rasul K and Vollgraf R 2017 Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (arXiv:1708.07747)

[61] Kingma D P and Ba J 2017 Adam: a method for stochastic optimization (arXiv:1412.6980)

[62] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems (tensorflow.org)

[63] Masci J, Meier U and Schmidhuber J 2011 Stacked convolutional auto-encoders for hierarchical feature extraction *21st Int. Conf. on Artificial Neural Networks* Proc., Part I pp 52–59

[64] Larocca M, Ju N, García-Martín D, Coles P J and Cerezo M 2021 Theory of overparametrization in quantum neural networks (arXiv:2109.11676)

[65] Bengio Y and Lecun Y 2007 *Scaling Learning Algorithms Towards AI* (Cambridge, MA: MIT Press)

[66] Möttönen M, Vartiainen J J, Bergholm V and Salomaa M M 2005 Transformation of quantum states using uniformly controlled rotations *Quantum Info. Comput.* **5** 467–73