Neural quantum embedding: Pushing the limits of quantum supervised learning

Tak Hur,[1] Israel F. Araujo,[1] and Daniel K. Park[1,2,*]

[1] *Department of Statistics and Data Science, Yonsei University, Seoul 03722, Republic of Korea*
[2] *Department of Applied Statistics, Yonsei University, Seoul 03722, Republic of Korea*

(Received 7 January 2024; revised 2 July 2024; accepted 8 July 2024; published 8 August 2024)

Quantum embedding is a fundamental prerequisite for applying quantum machine learning techniques to classical data and has substantial impacts on performance outcomes. In this study, we present neural quantum embedding (NQE), a method that efficiently optimizes quantum embedding beyond the limitations of positive and trace-preserving maps by leveraging classical deep-learning techniques. NQE enhances the lower bound of the empirical risk, leading to substantial improvements in classification performance. Moreover, NQE improves robustness against noise. To validate the effectiveness of NQE, we conduct experiments on IBM quantum devices for image data classification, resulting in a remarkable accuracy enhancement from 0.52 to 0.96. In addition, numerical analyses highlight that NQE simultaneously improves the trainability and generalization performance of quantum neural networks, as well as of the quantum kernel method.

DOI: 10.1103/PhysRevA.110.022411

I. INTRODUCTION

Machine learning (ML) is ubiquitous in modern society, owing to its capability to identify patterns from data. For well-behaved data, simple learning algorithms such as linear regression and support vector machines are often sufficient to capture the underlying data distribution. In contrast, intricate and high-dimensional data typically require advanced learning algorithms, substantial computational power, and extensive training data. The overarching objective of machine learning is to construct models that can effectively learn the underlying distributions of complex real-world data. However, achieving this objective presents a substantial challenge.

Recent advances in quantum computing (QC) have led to the development of quantum machine learning (QML). QML aims to efficiently process complex data distributions by leveraging the computational benefits of quantum algorithms [1–5]. One potential benefit of QC relevant to ML is its ability to efficiently sample from certain probability distributions that are exponentially difficult for classical counterparts [6–8], as validated in several experiments [9–11]. Quantum sampling algorithms typically impose lower requirements on physical implementations, making them an attractive pathway for demonstrating the quantum advantage using noisy intermediate-scale quantum (NISQ) devices [12]. One of the primary rationales underpinning the potential of QML is as follows: if a quantum computer can efficiently sample from a computationally hard probability distribution, it is plausible that quantum computers can efficiently learn from data drawn from such distributions. This implies a potential quantum advantage, especially for data distributions that are computationally infeasible for classical models but easily tractable for quantum models.

While quantum data are naturally suited for QML tasks [5], most contemporary data-science challenges involve classical data. Consequently, the exploration of the effectiveness of QML algorithms in learning from classical data constitutes a critical research focus. Notable examples of QML models tailored for classical data include the quantum neural network (QNN) and quantum kernel method (QKM), both of which are specialized for supervised learning problems. QNN utilizes a parameterized quantum circuit where the parameters are optimized through the variational method [13–16]. In contrast, QKM utilizes a quantum kernel function to effectively capture the correlations within the data [17,18].

In QML tasks involving classical data, an essential initial step is quantum embedding, which maps classical data into quantum states that a quantum computer can process. Quantum embedding is of paramount importance because it can significantly impact the performance of the learning model, including aspects such as expressibility [19], generalization capability [20], and trainability [21]. Therefore, selecting an appropriate quantum-embedding circuit is crucial for the successful learning of the data with quantum models. To achieve a quantum advantage in machine learning, prevailing research emphasizes designing quantum-embedding circuits that are computationally challenging to simulate classically [17,22]. In this work, we redirect attention to the data separability of embedded quantum states, utilizing trace distance, a tool in quantum information theory used to measure the distinguishability between quantum states [23,24], as a figure of merit. Subsequent sections will show that the choice of a quantum-embedding circuit inherently dictates a lower bound of empirical risk, independent of any succeeding trainable quantum circuits. Specifically, in the context of binary classification employing a linear loss function, the empirical risk is bounded from below by the trace distance between two ensembles of data-embedded quantum states representing different classes. Therefore, opting for a

[*] Contact author: dkd.park@yonsei.ac.kr

quantum embedding that maximizes trace distance—and thus, enhances distinguishability of states—facilitates improved training performance. Furthermore, a larger trace distance enhances resilience to noise, as the data-embedded quantum states reside farther from the decision boundary.

Conventional quantum-embedding schemes are generally data agnostic and do not guarantee high levels of data separability for a given dataset. To achieve a large trace distance, the use of trainable quantum embeddings is essential. Some efforts have explored trainable quantum embeddings by employing parameterized quantum circuits in both QNN [25] and QKM [26] frameworks. However, incorporating these quantum circuits increases the quantum circuit depth and the number of gates, making it less compatible with NISQ devices. Furthermore, the inclusion of trainable quantum gates during the quantum-embedding phase increases the model's susceptibility to barren plateaus [27], thereby adversely affecting the efficient training.

Given these considerations, we present neural quantum embedding (NQE), an efficient method that leverages the power of classical neural networks to learn the optimal quantum embedding for a given problem. NQE can enhance the quantum data separability beyond the capabilities of quantum channels, thereby extending the fundamental limits of quantum supervised learning. Our approach avoids the critical issues present in existing methods, such as the increased number of gates and quantum circuit depth and the exposure to the risk of barren plateaus. Numerical simulations and experiment with IBM quantum devices confirm the effectiveness of NQE in enhancing QML performance in several key metrics in machine learning. These improvements extend to training accuracy, generalization capability, trainability, and robustness against noise, surpassing the capabilities of existing quantum-embedding methods.

II. RESULTS

A. Lower bound of empirical risk in quantum binary classification

In supervised learning, the primary objective is to identify a prediction function f that minimizes the true (expected) risk $R(f)=\mathbb{E}\{l[f(X),Y]\}$ with respect to some loss function l, where X and Y are drawn from an unknown distribution D. Given a collection of N sample data $\{(x_i,y_i)\}$, the goal of learning algorithms is to find the optimal function f^* that minimizes the empirical risk $R_N(f)=(1/N)\sum_{i=1}^{N} l[f(x_i),y_i]$ among a fixed function class F, i.e., $f^*=\arg\min_{f\in F} R_N(f)$. Quantum supervised learning algorithms aim to efficiently find prediction functions with improved performance by exploiting the computational power of the quantum device.

A QNN is a widely used method for quantum supervised learning. In QNN, a classical input data x is first embedded into a quantum state by applying a quantum-embedding circuit to an initial ground state, resulting in $|x\rangle=\Phi(x)|0\rangle^{\otimes n}$. Next, a parameterized unitary operator, denoted as $U(\theta)$, is applied to transform the embedded quantum states, and the state is measured with an observable O. The measurement outcome serves as a prediction function for supervised learning algorithms, expressed as $f(x;\theta)=\langle x|U^\dagger(\theta)OU(\theta)|x\rangle$. Subsequently, using gradient descent or one of its variants, we search for the optimal parameter θ^* that minimizes the empirical risk. For a binary classification task with input data $x\in\mathbb{R}^m$ and its associated label $y\in\{-1,1\}$, we can predict the label of the new data x_{new} using the rule $y_{\text{new}}=\text{sign}[f(x_{\text{new}};\theta^*)]$.

Alternatively, we can consider this procedure as a quantum state discrimination problem involving two parameterized positive operator-valued measures (POVMs), denoted as $E_\pm(\theta)=[I\pm U^\dagger(\theta)OU(\theta)]/2$. With these POVMs, the probabilities of obtaining measurement outcomes ±1 given an input data x are computed as $P[E_\pm(\theta)|x]=\langle x|E_\pm(\theta)|x\rangle$. Subsequently, the decision rule for the new data is determined as $y_{\text{new}}=\text{sign}\{P[E_+(\theta)|x_{\text{new}}]-P[E_-(\theta)|x_{\text{new}}]\}$. In such a scenario, a natural loss function is the probability of misclassification, which can be expressed as $l[f(x;\theta),y]=P[E_{\neg y}(\theta)|x]$. Considering a dataset of N samples, $S=\{x_i^-,-1\}_{i=1}^{N^-}\cup\{x_i^+,1\}_{i=1}^{N^+}$; the empirical risk becomes

$$\begin{aligned} L_s &= \frac{1}{N}\left\{\sum_{i=1}^{N^-} P[E_+(\theta)|x_i^-] + \sum_{i=1}^{N^+} P[E_-(\theta)|x_i^+]\right\} \\ &\geqslant \frac{1}{2} - D_{\text{tr}}(p^-\rho^-, p^+\rho^+), \end{aligned} \tag{1}$$

where $\rho^\pm=\sum|x_i^\pm\rangle\langle x_i^\pm|/N^\pm$, $p^\pm=N^\pm/N$, and $D_{\text{tr}}(\cdot,\cdot)$ denotes the trace distance [28]. It is important to note the contractive property of the trace distance given by

$$D_{\text{tr}}[\Lambda(\rho_0),\Lambda(\rho_1)]\leqslant D_{\text{tr}}(\rho_0,\rho_1), \tag{2}$$

for any positive and trace-preserving (PTP) map Λ [29]. Based on the above, we now emphasize two crucial points.

(i) The empirical risk is lower bounded by the trace distance between two data ensembles $p^-\rho^-$ and $p^+\rho^+$. This bound is completely determined by the initial quantum-embedding circuit, regardless of the structure of the parameterized unitary gates $U(\theta)$ applied afterwards.

(ii) The minimum loss is achieved when $\{E_-(\theta),E_+(\theta)\}$ is a Helstrom measurement. Therefore, the training of a quantum neural network can be viewed as a process of finding the Helstrom measurement that optimally discriminates between the two data ensembles.

Designing a quantum embedding that maximizes the trace distance is of paramount importance since it minimizes the lower bound of the empirical risk. This becomes especially important in NISQ applications, as nonunitary quantum operations, such as noise, strictly reduce the trace distance between two quantum states [23,24]. Therefore, there is a clear need for a trainable, data-dependent embedding that can maximize the trace distance.

Several works have proposed combining a set of parameterized quantum gates and a conventional quantum-embedding circuit as a means to create a trainable unitary embedding [25,26,30]. However, the use of parameterized quantum gates comes with several drawbacks. First, it results in an increase in the number of gates and the depth of the quantum circuit. This not only increases computational costs but also makes the quantum embedding more susceptible to noise. Furthermore, the method is prone to encountering barren plateaus, which pose a fundamental obstacle to scalability [21,27]. Second, the trainable unitary embedding is highly

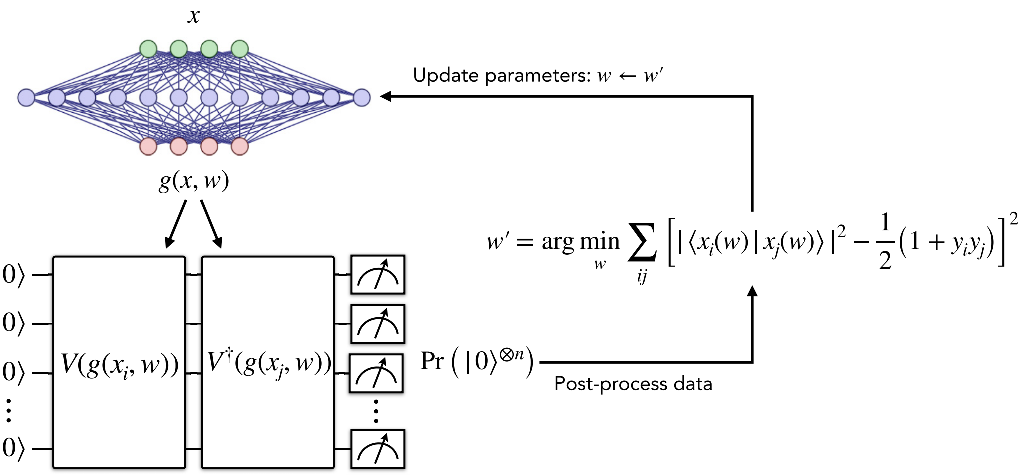

x
Update parameters: $w \leftarrow w'$
$g(x, w)$
$w' = \arg\min_{w} \sum_{ij} \left[|\langle x_i(w) | x_j(w) \rangle|^2 - \frac{1}{2}(1 + y_i y_j) \right]^2$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$V(g(x_i, w))$
$V^\dagger(g(x_j, w))$
$\Pr(|0\rangle^{\otimes n})$
Post-process data

FIG. 1. Overview of the NQE training. The unitary transformation that maps x_i to the quantum feature space is determined by the output of a classical neural network denoted by $g(x_i, w)$, where w represents trainable parameters. The resulting quantum state is $|x_i(w)\rangle = V[g(x_i, w)]|0\rangle^{\otimes n}$. The goal of the training is to produce mapping functions that can separate the two classes of data into two orthogonal subspaces. Efficient calculation of the fidelity between the two quantum states produced by the feature map is performed using a quantum computer.

restricted in enhancing the maximum trace distance of embedded quantum states (see Sec. II C 2). It is crucial to note that none of the existing quantum embeddings can guarantee the effective separation of two data ensembles in the Hilbert space with a large distance.

B. Neural quantum embedding

Neural quantum embedding utilizes a classical neural network to maximize the trace distance between two ensembles $D_{\text{tr}}(p^- \rho^-, p^+ \rho^+)$. It can be expressed as $\Phi_{\text{NQE}} : x \rightarrow |x\rangle = V[g(x, w)]|0\rangle^{\otimes n}$, where V is a general quantum-embedding circuit and $g : \mathbb{R}^m \times \mathbb{R}^r \rightarrow \mathbb{R}^{m'}$ is a classical neural network that transforms the input data x using r trainable parameters.

By choosing $m' < m$, we can bypass additional classical feature reduction methods, such as principal component analysis (PCA) or autoencoders, typically employed prior to quantum embedding due to the current limitations on the number of reliably controllable qubits in quantum devices. Ideally, the loss function should directly contain the trace distance. However, calculating it is computationally expensive, even with the quantum computer. Therefore, we used an implicit loss function derived from a fidelity measure, which is expressed as

$$l_{\text{fid}}[(x_i, y_i), (x_j, y_j)] = \left[|\langle x_i | x_j \rangle|^2 - \tfrac{1}{2}(1 + y_i y_j) \right]^2. \quad (3)$$

This fidelity loss can be efficiently computed using the SWAP test [31] or directly measuring the state overlap (see Fig. 1). The relationship between the state fidelity and the trace distance, as well as how minimizing l_{fid} corresponds to enhancing the trace distance, are detailed in Appendix C.

While NQE is not restricted by the choice of the quantum-embedding circuit, we specifically focus on improving the *ZZ* feature embedding [17]. The unitary operator corresponding to this embedding is expressed as

$$V(\phi(x)) = \left\{ \exp\left[i \sum_i \phi_i(x) Z_i + i \sum_{i,j} \phi_{i,j}(x) Z_i Z_j \right] H^{\otimes n} \right\}^L, \quad (4)$$

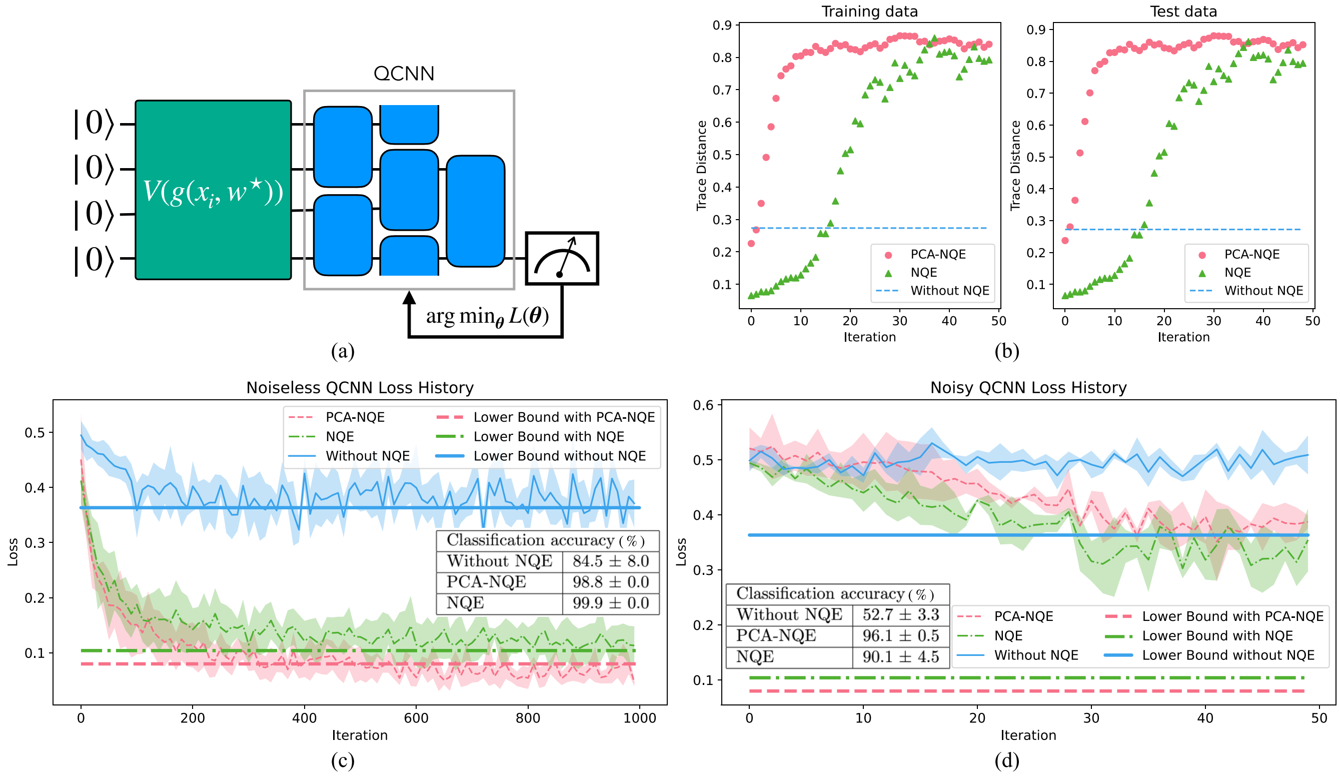
where $L > 1$. The use of this embedding is prevalent due to the conjectured intractability of computing its kernel classically when $L > 1$ [17]. It has been extensively explored in the field of quantum machine learning, including theoretical investigations [15,32,33] as well as practical applications in areas such as drug discovery [34,35], high-energy physics [36,37], and finance [38,39]. The most commonly used functions for ϕ are $\phi_i(x) = x_i$ and $\phi_{i,j}(x) = (\pi - x_i)(\pi - x_j)/2$ [15,17], but these choices are made without justifications. Although Ref. [40] numerically illustrates that the choice of ϕ can significantly impact the performance of QML algorithms, it does not provide guidelines for selecting an appropriate ϕ for the problem at hand. NQE effectively solves this limitation by replacing mapping functions with a trainable classical neural network.

While optimizing the *ZZ* feature embedding through NQE requires the use of a quantum computer due to the computational hardness of the loss function, there are certain embeddings that can be optimized solely on a classical computer. An instance of this is the amplitude encoding [41–45], where the corresponding loss function for NQE can be computed by taking the dot product of two vectors.

C. Experimental results

1. NQE versus fixed unitary embedding

This section presents experimental results that demonstrate the effectiveness of NQE in enhancing QML algorithms. To this end, we employed a four-qubit quantum convolutional neural network (QCNN) [13,46–49] for the task of classifying images of 0 and 1 within the MNIST dataset [50], a



Classification accuracy (%)	
Without NQE	84.5 ± 8.0
PCA-NQE	98.8 ± 0.0
NQE	99.9 ± 0.0

Classification accuracy (%)	
Without NQE	52.7 ± 3.3
PCA-NQE	96.1 ± 0.5
NQE	90.1 ± 4.5

FIG. 2. (a) Schematic representation of the quantum circuit used in the experiments. The green rectangle indicates the neural quantum embedding (NQE), which transforms classical data x_i into a quantum state $|x_i\rangle$. The blue rectangles represent two-qubit parameterized quantum gates of the quantum convolutional neural network (QCNN), designed for binary classification tasks. (b) Plot depicting the evolution of the trace distance between two ensembles of quantum states embedded by the NQE models during training on the ibmq_toronto device, compared to the trace distance from conventional quantum embedding without NQE. (c) Noiseless QCNN simulation results. (d) The results from QCNN experiments conducted on IBM quantum devices. In (c) and (d), the blue solid, red dashed, and green dash-dotted lines represent the mean training loss histories for conventional *ZZ* feature embedding, PCA-NQE, and NQE, respectively. The shaded regions in the figure represent one standard deviation from the mean. These values are acquired from five repetitions of each QCNN training with random initialization of parameters. The thicker versions of these lines indicate the theoretical lower bounds for each method.

well-established repository of handwritten digits. We conducted experiments using both noiseless simulations and quantum devices accessible through the IBM cloud service. The experiment unfolds in three main phases: the application of NQE, the training of the QCNN with and without NQE models, and the assessment of classification accuracies for the trained QCNN with and without NQE models.

We compared NQE against the conventional*ZZ* feature embedding with the aforementioned function $\phi_i(x) = x_i$ and $\phi_{i,j}(x) = (\pi - x_i)(\pi - x_j)/2$. Due to the limited number of qubits that can be reliably manipulated in current quantum devices, it is often necessary to reduce the number of features in the original data before embedding it as a quantum state. To address this issue, we tested two different NQE structures for incorporating dimensionality reduction. In the first approach, which we refer to as PCA-NQE, we applied PCA to reduce the number of features before passing them to the neural network. In the second approach, which we simply refer to as NQE, dimensionality reduction was directly handled within the neural network by adjusting the number of input and output nodes accordingly. For both PCA-NQE and NQE, classical neural networks produce eight dimensional vectors, which are then used as rotational angles for the four-qubit *ZZ* feature embedding. These two methods differ in their input requirements: PCA-NQE takes four-dimensional vectors as input, requiring classical feature reduction, while NQE accepts the original 28 × 28 image data, bypassing the need for additional classical preprocessing. Further details on the structure of the *ZZ* feature embedding circuit and the classical neural networks used for NQE methods are provided in Appendix A 1.

The goal of NQE training is to maximize the trace distance between the two sets of data ensembles, thereby minimizing the lower bound of the empirical risk. Figure 2(b) displays the trace distance as a function of NQE training iterations, acquired from ibmq_toronto. It is evident that both PCA-NQE and NQE effectively separate the embedded data ensembles and enhance the distinguishability of the quantum state representing different classes, even in the presence of noise in the real quantum hardware. After optimization, PCA-NQE and NQE yield trace distances of 0.840 and 0.792, respectively, which are significantly improved compared to the conventional *ZZ* feature embedding with a distance of 0.273. Consequently, the lower bound of the empirical risk is significantly reduced to 0.08 and 0.104, respectively, from the original value of 0.364.

The training of the QCNN circuit provides additional evidence supporting the effectiveness of NQE in QML.

Figure 2(c) presents the results of noiseless QCNN simulations. In this figure, the blue solid, red dashed, and green dash-dotted lines represent the mean training loss histories for the conventional *ZZ* feature embedding, PCA-NQE, and NQE, respectively. The thicker versions of these lines indicate the theoretical lower bounds for each method. The mean values are calculated from five repetitions of each QCNN training with random initialization of parameters. The shaded regions in the figure illustrate one standard deviation from the mean. The empirical risks for all models converge toward their respective theoretical minima, affirming that the trained QCNN adequately approximates optimal Helstrom measurements. Classification accuracies achieved on the test dataset are shown in the table within the figure. NQE methods led to significant enhancements, as demonstrated by reduced empirical risks and improved classification accuracies. Improvements in classification accuracies are expected as NQE embeds the training data into a specific subspace within the given Hilbert space such that the state distinguishability is maximized. The localization of embedded data facilitates the use of ML models with reduced complexity, thereby implying an enhancement in generalization capability. Numerical results supporting this intuition are presented in Secs. II D, II E, and Appendix F. Note that in some instances, the training loss falls below the theoretical lower bound of empirical risk. This occurs because the training loss is computed from a randomly sampled minibatch of data in each iteration.

Figure 2(d) presents the mean QCNN training loss histories obtained using IBM quantum devices. The mean values are calculated from three independent trials on ibmq_jakarta, ibmq_perth, and ibmq_toronto devices with random initialization of parameters. The shaded regions in the figure illustrate one standard deviation from the mean. The presence of noise and imperfections in the quantum devices prevent the empirical risks from reaching their theoretical lower bounds. Nonetheless, the training performance is significantly improved by NQE. Notably, for both NQE methods, the empirical risk rapidly approaches or even falls below the theoretical limit of the conventional method. This result underscores that even on the current noisy quantum devices, our method can outperform the theoretical optimum of the *ZZ* feature map without any additional quantum error mitigation. Moreover, a substantial improvement in classification accuracy was recorded, reaching 96% (PCA-NQE) and 90% (NQE), whereas the conventional method achieved only 52%. These findings demonstrate that NQE enhances the noise resilience of QML models, improving the utility of NISQ devices. Comprehensive details of these experiments are provided in Appendix A 2.

2. *NQE versus trainable unitary embedding*

We also conduct numerical comparisons between NQE and trainable unitary embedding to further investigate the advantage of NQE. Trainable unitary embedding utilizes trainable unitary $U_{\text{tra}}(\theta)$ to find the quantum embedding that separates the data well. More specifically, one can implement a trainable unitary embedding $\Phi(x;\theta)$ as

$$\Phi(x;\theta) : |0\rangle^{\otimes n} \to |x;\theta\rangle = U_{\text{emb}}(x)U_{\text{tra}}(\theta)|0\rangle^{\otimes n}. \quad (5)$$

In this case, the data-embedded ensembles are expressed as

$$\begin{aligned}\rho^{\pm}(\theta) &= \frac{1}{N^{\pm}}\sum_i U_{\text{emb}}(x_i^{\pm})U_{\text{tra}}(\theta)|0\rangle\langle 0|^{\otimes n}U_{\text{tra}}^{\dagger}(\theta)U_{\text{emb}}^{\dagger}(x_i^{\pm})\\ &= \mathcal{E}^{\pm}[|\psi(\theta)\rangle\langle\psi(\theta)|],\end{aligned} \quad (6)$$

where $|\psi(\theta)\rangle = U_{\text{tra}}(\theta)|0\rangle^{\otimes n}$ and $\mathcal{E}^{\pm}(\cdot)$ are quantum channels that maps $\rho \to \sum_i K_i^{\pm}\rho K_i^{\pm\dagger}$, with $K_i^{\pm} = U_{\text{emb}}(x_i^{\pm})/\sqrt{N^{\pm}}$. Now the maximum trace distance between two data ensembles is upper bounded by the diamond distance,

$$\begin{aligned}&\max_{\theta} D_{\text{tr}}[p^{+}\rho^{+}(\theta), p^{-}\rho^{-}(\theta)]\\ &\quad= \max_{\theta}\|p^{+}\mathcal{E}^{+}[|\psi(\theta)\rangle\langle\psi(\theta)|] - p^{-}\mathcal{E}^{-}[|\psi(\theta)\rangle\langle\psi(\theta)|]\|_1\\ &\quad\leqslant D_{\diamond}(p^{+}\mathcal{E}^{+}, p^{-}\mathcal{E}^{-}).\end{aligned} \quad (7)$$

This presents a significant limitation since $\mathcal{E}^{\pm}$ are entirely predetermined by the choice of quantum-embedding circuit $U_{\text{emb}}(\cdot)$, without any guarantee that the diamond distance will be large. The trainable unitary $U_{\text{tra}}(\theta)$ does not contribute to improving the upper bound of the maximum trace distance.

Alternatively, one may consider the data re-uploading technique in which the trainable unitary and quantum-embedding circuit are repeatedly applied multiple times,

$$\Phi(x;\theta) : |0\rangle^{\otimes n} \to |x;\theta\rangle = \prod_{l=1}^{L} U_{\text{emb}}(x)U_{\text{tra}}(\theta_l)|0\rangle^{\otimes n}. \quad (8)$$

However, Ref. [51] demonstrates that data–re-uploading quantum embedding can be exactly transformed into a form where all the trainable unitaries follow the quantum-embedding circuits by introducing ancilla qubits. Upon such transformation, the embedding can be expressed as

$$U'_{\text{tra}}(\boldsymbol{\theta})U'_{\text{emb}}(x)|0\rangle^{\otimes n+n'}, \quad (9)$$

where $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_L]$ and $n' \in O[L\log(L)]$. Importantly, the embedding circuit is independent of the parameters of the trainable unitary gates. Consequently, the maximum trace distance is once again determined by the choice of quantum-embedding circuit, which, in turn, constrains the data distinguishability. Furthermore, employing multiple layers of trainable unitary and quantum-embedding circuit significantly increases the total circuit depth, making it not only unsuitable for NISQ applications, but also susceptible to the barren plateaus problem.

Figure 3 displays a comparison of the NQE and trainable unitary embedding under noiseless and noisy environments employing eight qubits. The PCA-NQE and NQE are implemented with adjustments made in both PCA and the classical neural network to accommodate the use of eight qubits (see Appendix A 1). The QCNN circuits are trained utilizing either trainable unitary embedding or NQE techniques. For the trainable unitary embedding, we used the following parameterized quantum circuit:

$$\prod_{l=1}^{L}\left\{V[\phi(x)]\exp\left(i\sum_i \theta_i^l Y_i + i\sum_{i,j}\theta_{i,j}^l Y_i Y_j\right)\right\}|0\rangle^{\otimes n}. \quad (10)$$

Here, $V[\phi(x)]$ is the *ZZ* feature map explained in Eq. (4) and θ_i^l ($\theta_{i,j}^l$) are the trainable parameters of the embedding.

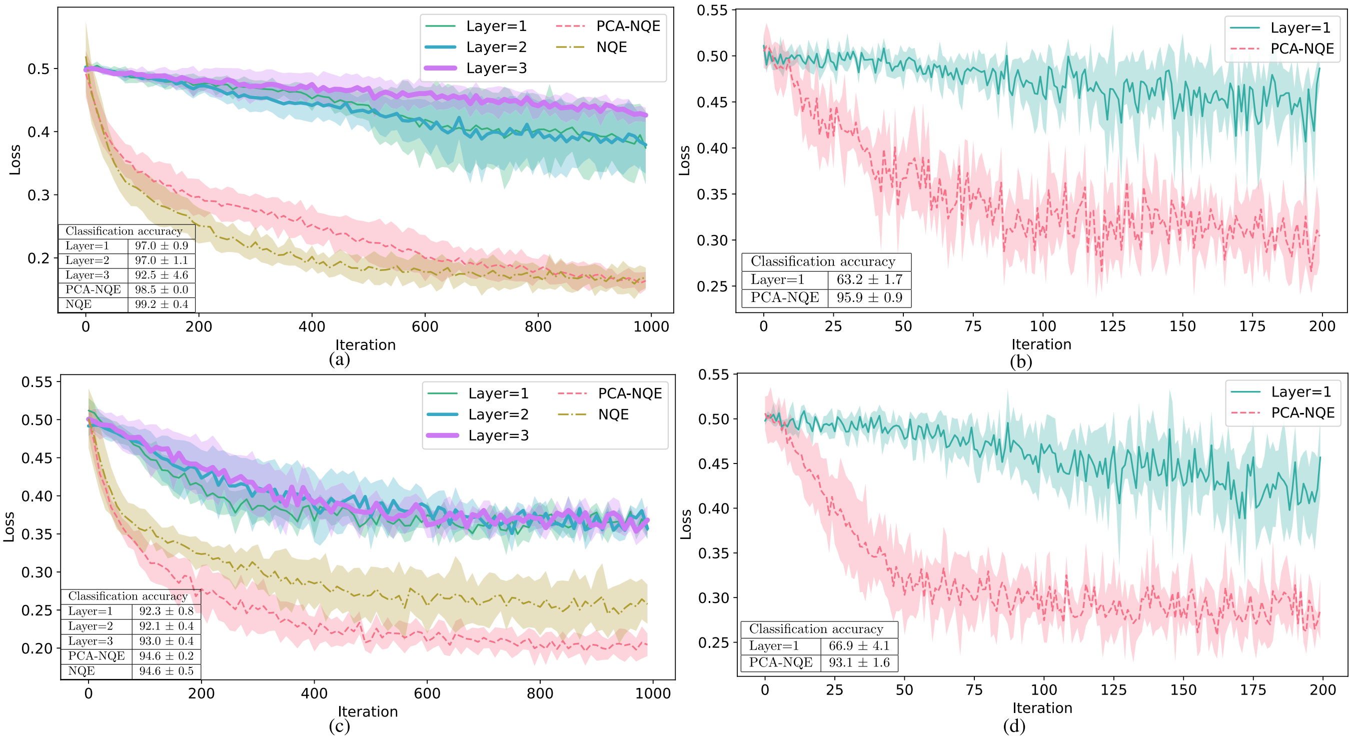


FIG. 3. Comparative analysis between neural quantum embeddings and trainable unitary embeddings with one, two, and three trainable layers. The numerical simulations were conducted under noiseless (left) and noisy (right) environments, utilizing MNIST (top) and Fashion-MNIST (bottom) datasets. For noiseless simulations, we used 1000 iterations, a learning rate of 0.01 learning rate, and batches of 128 data points per iteration. For noisy simulations, we used 200 iterations, a learning rate of 0.05, and batches of 15 data point per iteration. The noisy model simulations utilized the IBM QISKIT FakeGuadalupe environment. The classification accuracies were evaluated using a sample size of 2115 and 2000 data points for the MNIST and Fashion-MNIST datasets, respectively. The mean and one standard deviation from five independent iterations are shown for the loss history.

Figures 3(a) and 3(c) depict the training loss history and classification accuracies in a noiseless environment for the MNIST ({0, 1}) and Fashion-MNIST ({T-shirt/Top, Trouser}) datasets [52], respectively. In this experiment, we explored the impact of one, two, and three trainable unitary layers in comparison to PCA-NQE and NQE. The results indicate that the NQE methods achieve a notably lower training loss than the trainable unitary embedding, suggesting superior efficacy of NQE in improving data separability (by increasing the trace distance). Additionally, the NQE methods resulted in higher classification accuracies.

Figures 3(b) and 3(d) depict the training loss history and classification accuracies in a noisy environment for the MNIST and Fashion-MNIST datasets, respectively. A single-layer trainable unitary embedding was evaluated against PCA-NQE. The choice of a single layer was based on its minimal circuit depth, rendering it less susceptible to noise interference. The results indicate that NQE is more effective in enhancing data separability under the presence of noise. Additionally, PCA-NQE yields considerably higher classification accuracies. These advantages stem from the larger initial trace distance and the reduced circuit depths associated with PCA-NQE. We employed a noisy simulation using the IBM FakeGuadalupe device. This simulator mimics the essential characteristics of the ibmq_guadalupe device, including its basis gates, qubit connectivity, qubit relaxation (T_1) and rephrasing (T_2) times, and readout error rates.

D. Effective dimension

As demonstrated thus far, NQE has proven to be an efficient method for reducing the lower bound of empirical risk. While this advancement enhances the ability to learn from data, another essential measure of successful machine learning is the generalization performance, that is, the ability to make accurate predictions on unseen data based on what has been learned. The simulation and experimental results presented in the previous section indicate an improvement in prediction accuracy on test data with the implementation of NQE. In this section, we provide additional evidence of improved generalization performance by analyzing the effective dimension (ED) [15,53,54] of a QML model constructed with and without NQE. Intuitively, ED quantifies the number of parameters that are active in the sense that they influence the outcome of its statistical model.

To investigate this further, we focus on the local effective dimension (LED) introduced in Ref. [54], as it takes into account the data and learning algorithm dependencies and is computationally more convenient. Importantly, the LED exhibits a positive correlation with the generalization error, allowing for straightforward interpretation of the results: a smaller LED corresponds to a smaller generalization error, and vice versa. Our numerical investigation employs a four-qubit QNN (see Appendix A 3) and the results are presented in Fig. 4. In the figure, the green solid and purple dashed

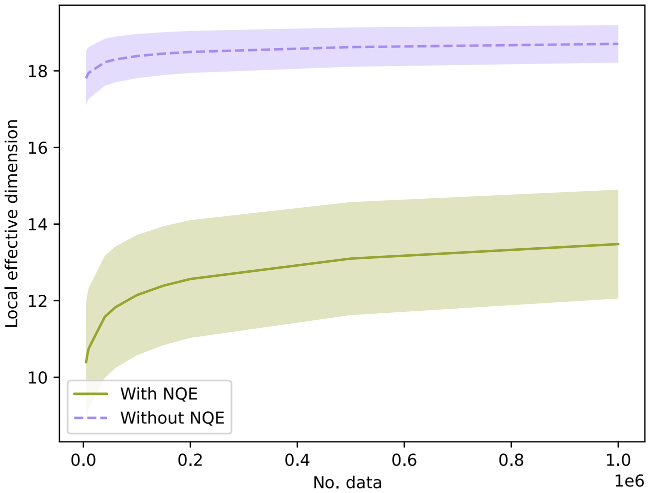


FIG. 4. The local effective dimension for the circuit with (solid green) and without (dashed blue) NQE. These results are based on 10 sets of experiments, each on a distinct artificial dataset with 20 repetitions with random initialization of parameters. The reported values represent the average across all 200 experiments.

lines represent the mean local effective dimension of the tested QML model with and without NQE, respectively. The mean values are computed from 200 trials, where each trial consists of 10 artificial datasets and, for each dataset, the experiment is repeated 20 times with random initialization of parameters. The shaded areas in the figure represent one standard deviation. The simulation results unequivocally demonstrate that NQE consistently reduces the LED in all 200 instances that are tested and across a wide range of training data sizes, signifying an improvement in generalization performance.

The effective dimension can also be interpreted as a measure of the volume of the solution space that a specific model class can encompass. A smaller effective dimension in a QML model implies a reduced volume of the solution space. This observation is particularly relevant as it suggests that models with smaller effective dimensions are less prone to encountering barren plateaus [55]. Consequently, the simulation results further indicate that NQE not only enhances generalization performance, but also improves the trainability of the model. Further evidence substantiating this improvement for both QNN and QKM will be presented in a later section.

E. Generalization in quantum kernel method

Up to this point, the investigation of NQE has primarily focused on its application within the context of quantum neural networks. In this section, we extend the analysis to demonstrate that NQE also enhances the performance of the quantum kernel method. Given a quantum embedding, the kernel function can be defined as

$$k^Q(x_i, x_j) = |\langle x_i|x_j\rangle|^2, \tag{11}$$

which can be efficiently computed on a quantum computer. The quantum kernel method refers to an approach that uses the kernel matrix K^Q, of which each entry is the kernel of the corresponding data points, in a method like a classical support vector machine [17,32]. The potential quantum advantage of such approach is based on the hardness to compute certain quantum kernel functions classically [17,18,56].

The quantum kernel method attempts to determine the function $f(x;W) = \mathrm{Tr}(W|x\rangle\langle x|)$ to predict the true underlying function $h(x)$ for unseen data x. The optimal parameters W^* are obtained by minimizing the cost function,

$$W^* = \underset{W\in\mathbb{C}^{2^n\times 2^n}}{\arg\min}\ \frac{1}{N}\sum_{i=1}^{N}[f(x_i;W)-h(x_i)]^2 - \lambda||W||_F^2, \tag{12}$$

where $||\cdot||_F$ is the Frobenius norm. The second term is the regularization term with a hyperparameter λ. The purpose of including the regularization term is to reduce the generalization error at the expense of the training error. Specifically, considering the true error $R(W) = \mathbb{E}_x|f(x;W)-h(x)|$ and the training error $R_N(W) = \sum_{i=1}^{N}|f(x_i;W)-h(x_i)|/N$, the generalization error is upper bounded as

$$|R(W)-R_N(W)| \leqslant \mathcal{O}\left(\frac{||W||_F}{\sqrt{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{13}$$

with probability at least $1-\delta$ (see Ref. [32] Supplementary Information, Sec. 4.C). The optimal W^* can be expressed as $W^* = \sum_{i=1}^{N}\sum_{j=1}^{N} h(x_i)(K^Q+\lambda I)^{-1}_{i,j}|x_j\rangle\langle x_j|$. Here, the employment of NQE affects W^* as both K^Q and $|x_j\rangle$ vary with the quantum embedding.

We performed empirical evaluations to assess the effectiveness of NQE in reducing $G = ||W^*||_F/\sqrt{N}$ and thereby improving the upper bound of the generalization error. The analysis proceeded in three steps: loading the dataset, computing the quantum kernel matrix with and without NQE, and calculating the generalization error bound with and without NQE. The experiments tested both PCA-NQE and NQE, with neural network parameters optimized on the ibmq_toronto processor, as detailed in Appendix A 1. During the dataset loading phase, binary datasets containing $N = 1000$ samples from classes {0,1} were constructed from the MNIST dataset. As outlined in Appendix A 1, both PCA-NQE and the conventional quantum embedding without NQE were preceded by PCA to reduce the number of features to four. In contrast, the NQE utilized the original 28 × 28 image datasets. Subsequently, three quantum kernel matrices were constructed: one without NQE, one with PCA-NQE, and one with NQE. The (i, j)th entry of the quantum kernel corresponds to $k^Q(x_i, x_j)$, and the fidelity overlap between pairs of data-embedded quantum states. This fidelity overlap was computed using PENNYLANE [57] numerical simulation. Finally, for each of the quantum kernel matrices, the corresponding upper bound of the generalization error $G = \sqrt{||W^*||_F^2/N}$ was calculated. Here, $||W^*||_F^2$ can be expressed as $||W^*||_F^2 = \sum_i\sum_j[(K^Q + \lambda I)^{-1}K^Q(K^Q+\lambda I)^{-1}]_{i,j}y_iy_j$.

The experimental procedure was repeated five times, each time with different sets of 1000 samples of data. Figure 5 presents the mean and one standard deviation of the generalization error bound G. These values were obtained under the ZZ feature embedding, both with and without NQE, and were examined across various regularization parameters λ. The results clearly illustrate that NQE significantly decreases the upper bound of the generalization error in quantum kernel methods.

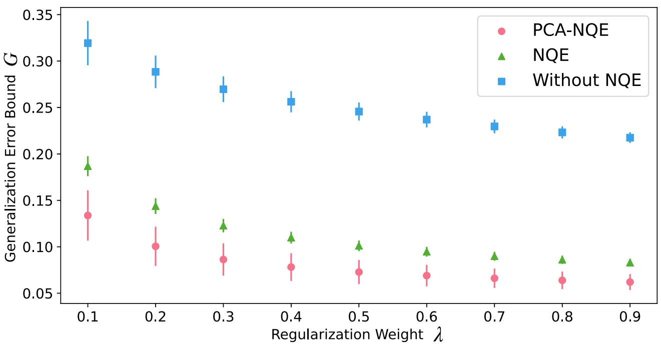


FIG. 5. A comparative analysis of the generalization error bound G with varying regularization weights λ. This plot illustrates the performance enhancement—lower generalization error bound—when employing NQE (green triangles) and PCA-NQE (red circles) over conventional methods without NQE (blue squares) in quantum kernel methods. PCA-NQE and NQE were optimized on the ibmq_toronto quantum hardware. The error bound G was determined based on five independent numerical simulations, presenting both the mean and one standard deviation of G.

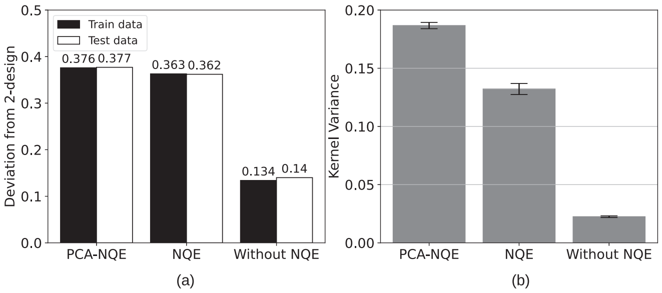


FIG. 6. (a) A comparative analysis of expressibility with and without NQE models. The deviation from the unitary 2-design is depicted, where a smaller deviation indicates higher expressibility. The deviation is derived from 12 665 (2115) MNIST training (test) data results. (b) A comparative analysis of the variance of quantum kernel elements with and without NQE models. The variance was computed from the off-diagonal elements of the quantum kernel matrix K^Q, constructed from 1000 samples of MNIST datasets. The mean and one standard deviation from five independent iterations are shown. For both (a) and (b), NQE models are optimized using the ibmq_toronto quantum device.

F. Expressibility and trainability

In both QNN and QKM, there exists a trade-off between expressibility and trainability. In the QNN framework, highly expressive quantum circuits often lead to barren plateaus, characterized by exponentially vanishing gradients, which severely hinders the trainability of the model [27,55]. In the QKM framework, highly expressive embedding induces a quantum kernel matrix whose elements exhibit an exponential concentration [21]. Specifically, the concentration of quantum kernel element $K_{i,j}^Q = k^Q(x_i, x_j)$ can be expressed by Chebyshev's inequality,

$$\Pr\big[\big|K_{i,j}^Q - \mathbb{E}\big(K_{i,j}^Q\big)\big| \geqslant \delta\big] \leqslant \frac{\mathrm{Var}\big(K_{i,j}^Q\big)}{\delta^2}, \tag{14}$$

for any $\delta > 0$. The quantum kernel element $K_{i,j}^Q$ arising from highly expressive quantum embedding displays an exponential reduction in variance as the number of qubits increases. Consequently, an exponentially large number of quantum circuit executions is necessary to accurately approximate the quantum kernel matrix K^Q. This poses a significant challenge in the efficient implementation of QKM.

NQE addresses this challenge by constraining quantum embedding to ensure large distinguishability. NQE strategically limits the expressibility of the embedded quantum states, thereby enhancing the trainability of QML models. This improvement is achieved by exploiting the prior knowledge that a quantum embedding with large distinguishability can effectively approximate the true underlying function.

Figure 6(a) illustrates how the expressibility varies as we apply NQE models. Here, we investigated the Hilbert-Schmidt norm of the deviation from unitary 2-design [55,58], as a measure of expressibility. More specifically, the deviation is given as

$$A = \int_{\mathrm{Haar}} (|\psi\rangle\langle\psi|)^{\otimes 2} d\psi - \int_{\mathcal{E}} (|\phi\rangle\langle\phi|)^{\otimes 2} d\phi, \tag{15}$$

where the first integral is taken over the Haar measure, and the second integral is taken over the ensemble of data-embedded quantum states, $\mathcal{E}$. We then define the deviation norm, $\epsilon = \sqrt{\mathrm{Tr}(A^\dagger A)}$, where a small ϵ indicates a highly expressive quantum embedding, and vice versa. In this experiment, we employed NQE and PCA-NQE, which had been previously optimized with ibmq_toronto hardware, as detailed in Sec. II C 1. The value of ϵ was numerically computed for ensembles of training and test datasets of classes $\{0,1\}$ from MNIST data, utilizing 12 665 instances for training data and 2115 instances for test data. We observe that for both training and test data, both NQE methods lead to reduction in the expressibility, consequently enhancing trainability.

Figure 6(b) demonstrates how the variance of the quantum kernel elements varies as we apply NQE models. To determine the variance, we initially computed quantum kernel matrix K^Q using PCA-NQE, NQE, and quantum embedding without NQE, following the procedures outlined in Sec. II E. The variance was calculated using the off-diagonal elements of K^Q. These experiments were repeated five times, each with different 1000 samples of data. The figure displays the mean and one standard deviation calculated from these results. A significant increase in the variance of quantum kernel elements is observed with the use of NQE models, implying that the quantum kernel matrix K^Q can be reliably approximated with fewer quantum circuit executions when NQE models are implemented. These results underscore the effectiveness of NQE in enhancing the trainability of QML models within both the QNN and QKM frameworks.

III. CONCLUSIONS AND DISCUSSION

In this study, we investigated the crucial role of quantum embedding, an essential step in applying quantum machine learning to classical data. In particular, we highlighted how quantum data separability, namely, the distinguishability of quantum states representing different classes, determines the lower bound of training error and affects the noise resilience of quantum supervised learning algorithms. Motivated by these results, we introduced neural quantum embedding (NQE), which utilizes the power of classical neural network and deep learning to enhance data separability in the quantum feature space, thereby pushing the limits of quantum supervised learning. Integrating classical neural networks with parameterized quantum circuits to construct ML models has been explored in various studies, as referenced in [48,59–62]. However, there remains ambiguity regarding why and how classical neural networks should be incorporated, aside from addressing the limited size of quantum circuits executable on NISQ computers or naively attempting to extend the success of deep learning to the domain of QML. Additionally, determining the optimal strategy for interfacing classical and quantum neural networks, particularly for transferring information from classical to quantum systems, remains an important open problem. This work bridges these gaps by linking quantum supervised learning with the theory of quantum state discrimination, and contributes to developing an effective approach for applying QML to classical data. Quantum supervised learning for data classification can be understood as a process of learning the optimal POVM for minimizing error in discriminating density matrices representing data samples across different classes. In this regard, the parameterized quantum circuit solely plays the role of identifying the optimal measurement, whereas the optimal training performance is dictated by how the data are encoded as the quantum state. This optimal performance cannot be improved by any PTP map (e.g., quantum channels). However, classical neural networks can be utilized to learn the feature map from data samples, maximizing the distinguishability of states in the quantum feature space beyond the limits of the PTP maps for the given dataset. A quantum computer is also essential in this task because the objective function of the classical neural network is based on the fidelity of quantum states, which is conjectured to be hard to compute classically for the quantum feature maps of interest. In this respect, NQE differs from existing classical-quantum neural networks in that it trains a classical neural network with an objective function computed by a quantum computer specifically to maximize the separability of quantum data, which is the optimal embedding strategy for classification tasks.

NQE is versatile in the sense that it can be integrated into all existing quantum data-embedding methods, such as amplitude encoding, angle encoding, and Hamiltonian encoding. The training performance achieved by NQE is guaranteed to be at least as good as those that do not use it and rely on a fixed embedding function. This is because if the fixed embedding function happens to be the optimal one for the given data, NQE will learn to use it. Experimental results on IBM quantum hardware demonstrate that NQE significantly enhances quantum data separability, as quantified by increased trace distance between two ensembles of quantum states. Utilizing NQE led to a significant reduction in training loss and an improvement in accuracy and noise resilience in the MNIST data classification tasks. Notably, the experimental results achieved by NQE-enabled QML models outperformed the theoretical optimal of the conventional *ZZ* feature embedding that does not employ NQE.

Furthermore, we conducted numerical comparisons between NQE and three trainable unitary embedding circuits using both MNIST and Fashion-MNIST datasets. This study encompassed both noiseless and noisy simulations. The results demonstrate that NQE outperforms trainable unitary embeddings in terms of both training and classification accuracies across all scenarios.

A significant portion of current research in QML focuses on the trade-off between the expressibility and trainability of variational circuits within quantum neural networks. For a QML model to be effective, it must possess a high degree of expressibility, which ensures that it can approximate the desired solution with considerable accuracy. Concurrently, the model should be trainable, enabling optimization via a gradient descent algorithm or its variants. However, expressibility and trainability present a trade-off: high expressibility typically leads to reduced trainability [21,27,55]. This trade-off constitutes a significant challenge in advancing QML. To address this challenge, a strategic approach is to utilize problem-specific prior knowledge. For example, Refs. [63,64] deliberately construct variational circuits with limited expressibility, yet ensures the inclusion of the desired solution, by harnessing data symmetry. However, such method is not universally applicable to general datasets that do not present any symmetry or group structure. NQE offers an effective solution to this challenge by optimizing the quantum data-embedding process. As elucidated in Sec. II, a good approximation of the true underlying function can only be achieved with quantum embeddings that ensure high distinguishability of the data. By using this prior knowledge, NQE constrains the quantum embedding to those that allow large distinguishability between quantum states that represent the data. Consequently, the embedded quantum states from NQE are less expressive, resulting in an improvement in trainability.

The ultimate goal of machine learning is to construct a model that not only accurately classifies the training data (optimization), but also generalizes well to unseen data. In conventional machine learning, a trade-off typically exists between optimization and generalization [65]. However, our experimental results indicate that the incorporation of NQE markedly enhances both optimization and generalization metrics. This improvement is evidenced by reduced training error, reduced test error, diminished local effective dimension, and a reduced generalization upper bound in the quantum kernel method. Consequently, NQE presents a robust methodology for optimizing learning performance, while preserving strong generalization capabilities.

Appendixes D and E present additional experimental findings. The former exhibits noiseless and noisy simulation outcomes on the Fashion-MNIST dataset, illustrating the advantages of NQE with an alternative dataset. The latter delves into 8- and 12-qubit noiseless simulation results obtained from MNIST datasets. These results demonstrate that the benefits of NQE persist even for larger quantum systems. Overall, the supplementary findings consistently affirm that NQE surpasses traditional quantum data-embedding methods.

Further research is necessary to explore the impact of the type or architecture of neural networks on the performance of NQE and its optimization for specific types of target data. For instance, investigating the applicability of recurrent neural networks (RNNs) for handling sequential data and convolutional neural networks (CNNs) for image data remains an interesting avenue for future work. The incorporation of structure learning introduced in Ref. [66] with NQE is noteworthy as it can further improve the embedding. However, one must consider the trade-off between performance and the computational overhead introduced by the structure learning. As an alternative to NQE, enhancing quantum data separability can be achieved by implementing a probabilistic non-TP embedding [67]. Comparing this approach with NQE or exploring their combination for potential enhancements represents a valuable direction for future investigation.

The data and software that support the findings of this study can be found in the following repository of Ref. [68] .

ACKNOWLEDGMENTS

This work was supported by the Institute of Information & communications Technology Planning & evaluation (IITP) grant funded by the Korea government (Grant No. 2019-0-00003, Research and Development of Core technologies for Programming, Running, Implementing and Validating of Fault-Tolerant Quantum Computing System), the Yonsei University Research Fund of 2023 (Grant No. 2023-22-0072), the National Research Foundation of Korea (Grants No. 2022M3E4A1074591 and No. 2023M3K5A1094805), and the KIST Institutional Program (Grant No. 2E32241-23-010).

APPENDIX A: EXPERIMENTAL DETAILS

1. NQE structures and training

The PCA-NQE method employs the principal component analysis (PCA) to extract n features from the original data, where n is the number of qubits used for data embedding. These features are then passed to a fully connected neural network with two hidden layers. In the case of four-qubit experiments, each hidden layer contains 12 nodes, whereas in the eight-qubit cases, each hidden layer contains 24 nodes. The neural network has $2n$ output nodes, corresponding to $2n$ numerical values used as quantum gate parameters for the embedding. The rectified linear unit (ReLU) function serves as the nonlinear activation. In contrast, NQE (without PCA) utilizes a two-dimensional (2D) convolutional neural networks (CNN) which takes original data as an input. After each convolutional layer, we used 2D max pooling to reduce the dimension of the data. The dimension of the nodes in each layer is 28×28, 14×14, 7×7, and $2n$, respectively.

The classical neural networks of the NQE models are optimized by minimizing the implicit loss function $l_{\text{fid}}[(x_i, y_i), (x_j, y_j)]$ given in Eq. (3), where i and j are the indices of the randomly selected training data. For the four-qubit real device experiments, the NQE models were trained for 50 iterations using the stochastic gradient descent with a learning rate of 0.1 and a batch size of 10. The loss function was evaluated on ibmq_toronto with the selection of four qubits based on the highest CNOTfidelities. For the *ZZ* feature embedding circuits, we configured the total number of layers (L) to 1 and applied two qubit gates only on the nearest-neighboring qubits to avoid an excessive number of CNOTgates.

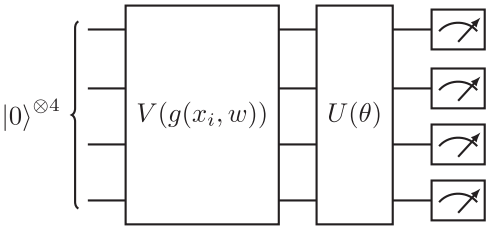


FIG. 7. Quantum circuit used for evaluating the local effective dimension. The feature map, denoted by $V[g(x_i, w)]$, acts on the initial state $|0\rangle^{\otimes 4}$ to encode the input vector x_i. Subsequently, a parameterized unitary transformation $U(\theta)$ is applied to evolve the state, with the parameters θ chosen to minimize a specific loss function.

2. Classification with QCNN

In the noiseless simulation setting, we utilized QCNN circuits featuring a general SU(4) convolutional ansatz (refer to Fig. 2(i) in Ref. [47]). The optimization of circuit parameters was performed over 1000 iterations using the Nesterov momentum algorithm with a learning rate of 0.01 and a batch size of 128. Each simulation was repeated five times with random parameter initialization.

For experiments on IBM quantum hardware, QCNN circuits were configured with a basic convolutional ansatz comprising two $R_y(\theta)$ gates, where $R_i(\theta)$ represent a single-qubit rotation around the i axis of the Bloch sphere by an angle θ, and a CNOT gate (refer to Fig. 2(a) in Ref. [47]). To minimize circuit depth, pooling gates were omitted. Moreover, the QCNN architecture was designed to allow only nearest-neighbor qubit interactions, eliminating the need for qubit swapping.

The training on quantum hardware consisted of 50 optimization iterations, using the Nesterov momentum gradient descent with a learning rate of 0.1 and a batch size of 10. Experiments were conducted on three distinct quantum devices: ibmq_jakarta, ibmq_toronto, and ibmq_perth.

Performance evaluation was carried out by assessing the classification accuracy of the trained QCNN models on a separate test set comprising 500 data points. This assessment was executed across three different quantum devices: ibmq_lagos, ibmq_kolkata, and ibmq_jakarta. The presented results were obtained from 1024 executions of quantum circuits.

3. Effective dimension

The analysis of the effective dimension utilizes the QNN architecture depicted in Fig. 7. The *ZZ* feature map, as explained in Eq. (4) and depicted in Fig. 8, is used for mapping classical data to quantum states. When NQE is not used, $g_j = x_{ij}$, which is the jth component of the input vector x_i,

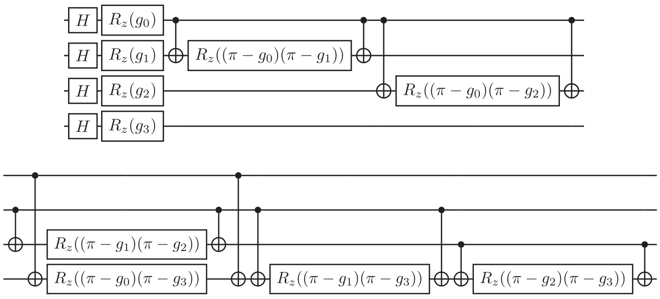

H
H
H
H
$R_z(g_0)$
$R_z(g_1)$
$R_z(g_2)$
$R_z(g_3)$
$R_z((\pi - g_0)(\pi - g_1))$
$R_z((\pi - g_0)(\pi - g_2))$
$R_z((\pi - g_1)(\pi - g_2))$
$R_z((\pi - g_0)(\pi - g_3))$
$R_z((\pi - g_1)(\pi - g_3))$
$R_z((\pi - g_2)(\pi - g_3))$

FIG. 8. Quantum circuit diagram for the trainable *ZZ* feature map $V[g(x_i, w)]$. The top and bottom segments collectively represent a single quantum circuit, split into two parts due to limited horizontal space. It uses mapping functions $g_j \in g(x_i, w)$ to encode the input vector x_i.

and the quantum embedding becomes equivalent to Eq. (4) with $L = 1$. When NQE is turned on, $g_j = g_j(x_i, w)$, which is the jth component of the output vector generated by a three-layer neural network. This output vector can be expressed as $g(x_i, w) = \sigma[w^{(1)}\sigma(w^{(0)}x_i + b^{(0)}) + b^{(1)}]$. In this equation, $w^{(0)} \in \mathbb{R}^{12\times4}$, $w^{(1)} \in \mathbb{R}^{4\times12}$, $b^{(0)} \in \mathbb{R}^{12}$, and $b^{(1)} \in \mathbb{R}^{4}$ are the trainable parameters of the network, and σ stands for the ReLU activation function.

The parameterized unitary operator of the QNN used in this analysis, represented as $U(\theta)$ in Fig. 7, is shown in Fig. 9. This circuit design is attractive for several reasons. First, it offers a high degree of expressibility and entangling capability while maintaining a relatively small number of gates and parameters [58]. In addition, this design is hardware efficient [16,69], as it relies solely on single-qubit rotations and CNOT operations between adjacent qubits.

The analysis encompassed 10 artificial binary datasets, each containing 400 samples. These datasets, characterized by four features and four clusters per class, were generated through the make_classification function from the Scikit-Learn library [70]. The NQE model was trained for 100 iterations using the Adam optimizer [71] with a batch size of 25.

The local effective dimension [54] was determined using the get_effective_dimension function from the LocalEffectiveDimension class in QISKIT [72].

APPENDIX B: RELATION BETWEEN LINEAR AND MSE LOSS

The main text focuses on maximizing the trace distance, which sets the optimal lower bound for the linear loss. However, many conventional quantum machine learning (QML) routines employ mean squared error (MSE) loss. Here, we provide some relationship between linear loss and MSE loss. Consider the data $\{x_i, y_i\}_{i=1}^{N}$ and the vectors $Y = (y_1, y_2, \dots, y_N)$ and $f(X) = [f(x_1), f(x_2), \dots, f(x_N)]$. The linear and MSE loss are expressed as

$$L_{\text{linear}} = ||Y - f(X)||_1, \tag{B1}$$

$$L_{\text{MSE}} = ||Y - f(X)||_2^2. \tag{B2}$$

By the vector norm inequalities, we can both upper bound and lower bound the MSE loss with the linear loss,

$$\frac{1}{N}L_{\text{linear}}^2 \leqslant L_{\text{MSE}} \leqslant L_{\text{linear}}^2. \tag{B3}$$

Reducing the lower bound of empirical linear loss by maximizing the trace distance reduces both the upper and lower bounds of the empirical MSE loss. Hence, we can expect neural quantum embedding (NQE) to work favorably for MSE loss as well.

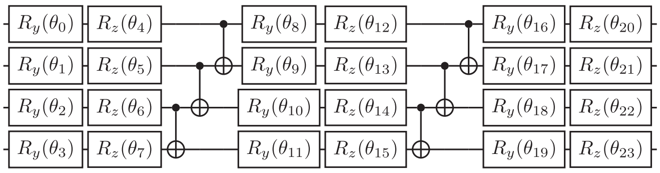


FIG. 9. Parameterized quantum circuit layout for the QNN, which is represented as $U(\theta)$ in Fig. 7.

APPENDIX C: RELATION BETWEEN IMPLICIT LOSS FUNCTION AND TRACE DISTANCE

During the NQE training, we optimize the classical neural network to maximize the trace distance between two data-embedded ensembles. Although using trace distance directly as a loss function is ideal, we utilized an implicit loss function due to the computational hardness of the trace distance calculation. The implicit loss function is delineated in Eq. (3).

When $y_i = y_j$, the loss function directs NQE to maximize the fidelity between $|x_i\rangle$ and $|x_j\rangle$ as much as possible. Due to the contractive property of the trace distance, $D(\rho^-, \rho^+) \leqslant D(|\psi^-\rangle, |\psi^+\rangle)$, where, $|\psi^-\rangle, |\psi^+\rangle$ are the purification of ρ^-, ρ^+, respectively. The equality holds when the two data ensembles are pure states. The purity of $\rho^\pm$ is

$$\text{Tr}[(\rho^\pm)^2] = \frac{1}{(N^\pm)^2}\sum_{i,j=1}^{N^\pm} |\langle x_i^\pm | x_j^\pm\rangle|^2.$$

Therefore, maximizing the fidelity when $y_i = y_j$ increases the purity of $\rho^\pm$, allowing the trace distance to achieve its upper bound.

Conversely, when $y_i \neq y_j$, the loss function directs NQE to minimize the fidelity between $|x_i\rangle$ and $|x_j\rangle$ as much as possible. For simplicity, let us consider a balanced set of data, $N^+ = N^- = N$. Due to strong convexity of the trace distance [24],

$$D\left(\frac{1}{N}\sum_{i=1}^{N}|x_i^-\rangle\langle x_i^-|, \frac{1}{N}\sum_{i=1}^{N}|x_i^+\rangle\langle x_i^+|\right) \leqslant \frac{1}{N}\sum_{i=1}^{N}\sqrt{1 - |\langle x_i^+|x_i^-\rangle|^2}. \tag{C1}$$

Hence, minimizing the fidelity when $y_i \neq y_j$ increases the upper bound of the trace distance. Therefore, it is evident that minimizing the implicit loss function contributes positively to maximizing the trace distance.

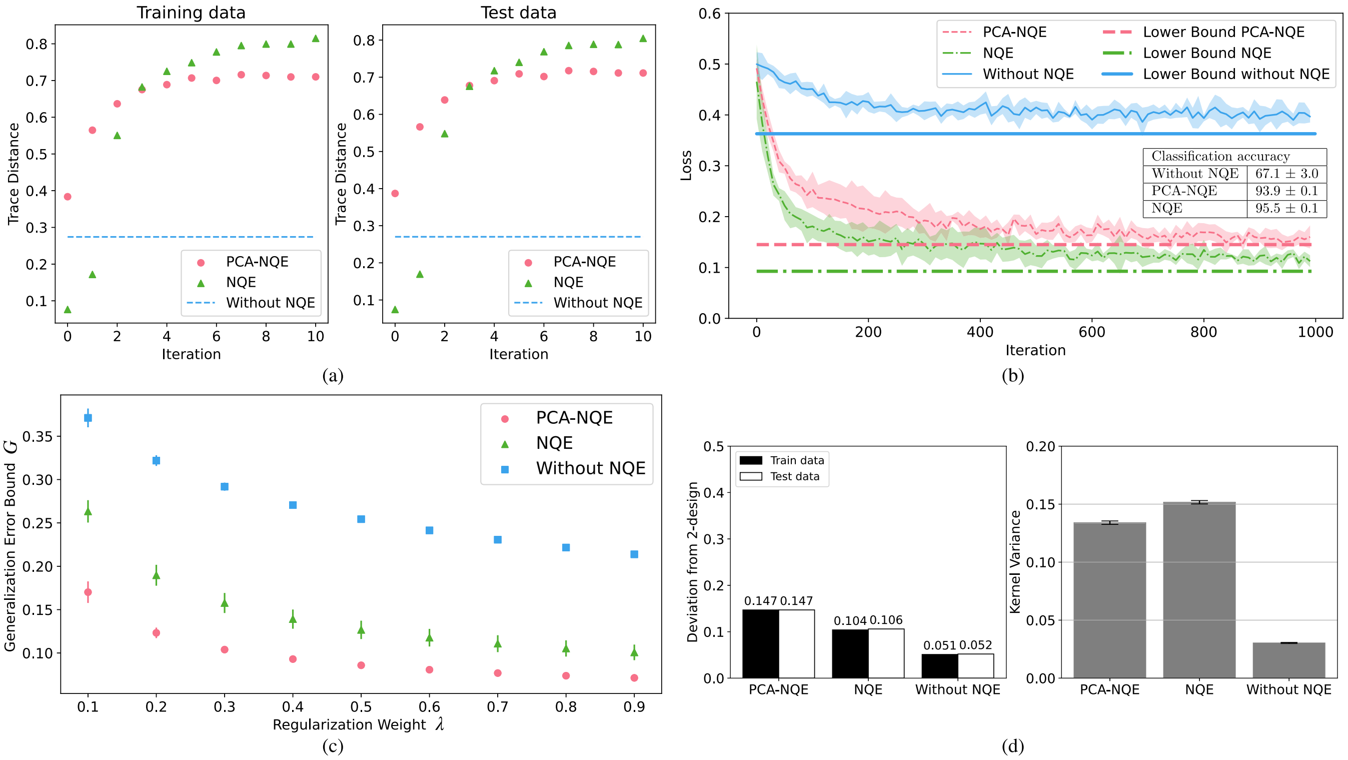


FIG. 10. Results for the additional experiments with Fashion-MNIST datasets using noiseless simulations. (a) Plot depicting the evolution of the trace distance between two ensembles of quantum states embedded by the NQE models during training, compared to the trace distance from conventional quantum embedding without NQE. (b) QCNN simulation results. The blue solid, red dashed, and green dash-dotted lines represent the mean training loss histories for conventional *ZZ* feature embedding, PCA-NQE, and NQE, respectively. The shaded regions in the figure represent one standard deviation from the mean. These values are acquired from five repetitions of each QCNN training with random initialization of parameters. The thicker versions of these lines indicate the theoretical lower bounds for each method. (c) A comparative analysis of the generalization error bound G with varying regularization weights λ. This plot illustrates the performance enhancement—lower generalization error bound—when employing NQE (green triangles) and PCA-NQE (red circles) over conventional methods without NQE (blue squares) in quantum kernel methods. The error bound G was determined based on five independent numerical simulations, presenting both the mean and one standard deviation of G. (d) Left: A comparative analysis of expressibility with and without NQE models. The deviation from unitary 2-design is depicted, where a smaller deviation indicates higher expressibility. The deviation is derived from 12 000 (2000) Fashion-MNIST training (test) data results. Right: A comparative analysis of the variance of quantum kernel elements with and without NQE models. The variance was computed from the off-diagonal elements of the quantum kernel matrix K^Q, constructed from 1000 samples of Fashion-MNIST datasets. The mean and one standard deviation from five independent iterations are shown.

APPENDIX D: SIMULATION RESULTS ON AN ADDITIONAL DATASET

In Sec. II C 1 of the main text, we presented how utilizing PCA-NQE and NQE improves QCNN performance when classifying MNIST datasets, by demonstrating trace distance history [Fig. 2(a)], as well as QCNN loss history and classification accuracies [Figs. 2(b) and 2(c)]. In Secs. II E and II F, we illustrated how employing NQE models can improve generalization performances and trainability.

In this section, we present additional experiments tested on classes {T-shirt/Top, Trouser} of the Fashion-MNIST dataset [52]. Figures 10 and 11 display results from noiseless and noisy simulations, respectively. In both figures, panel (a) depicts the trace distance history with and without NQE models, panel (b) presents QCNN loss history and classification accuracy with and without NQE models, panel (c) presents the upper bound of generalization error in QKM with and without NQE models, and panel (d) presents expressibility and trainability with and without NQE models. In alignment with the results in the main text, additional experiments with the Fashion-MNIST datasets indicate that employing NQE effectively improves training error, classification accuracy, generalization performance, and trainability.

The methodology for these experiments mirrors that of the experiments with MNIST datasets, which are detailed in Appendix A 1, and Secs. II E and II F. Due to the constraints in accessing IBM quantum hardware, we adapted our approach for the noisy experiments. Instead of direct hardware utilization, we employed a simulation environment using the IBM FakeVigo device. This simulator mimics the essential characteristics of the ibmq_vigo device, including its basis gates, qubit connectivity, qubit relaxation (T_1) and dephasing (T_2) times, and readout error rates.

APPENDIX E: SIMULATION RESULTS WITH LARGER QUANTUM CIRCUITS

In Sec. II C 1 of the main text, we presented how utilizing NQE improves QCNN performances by demonstrating

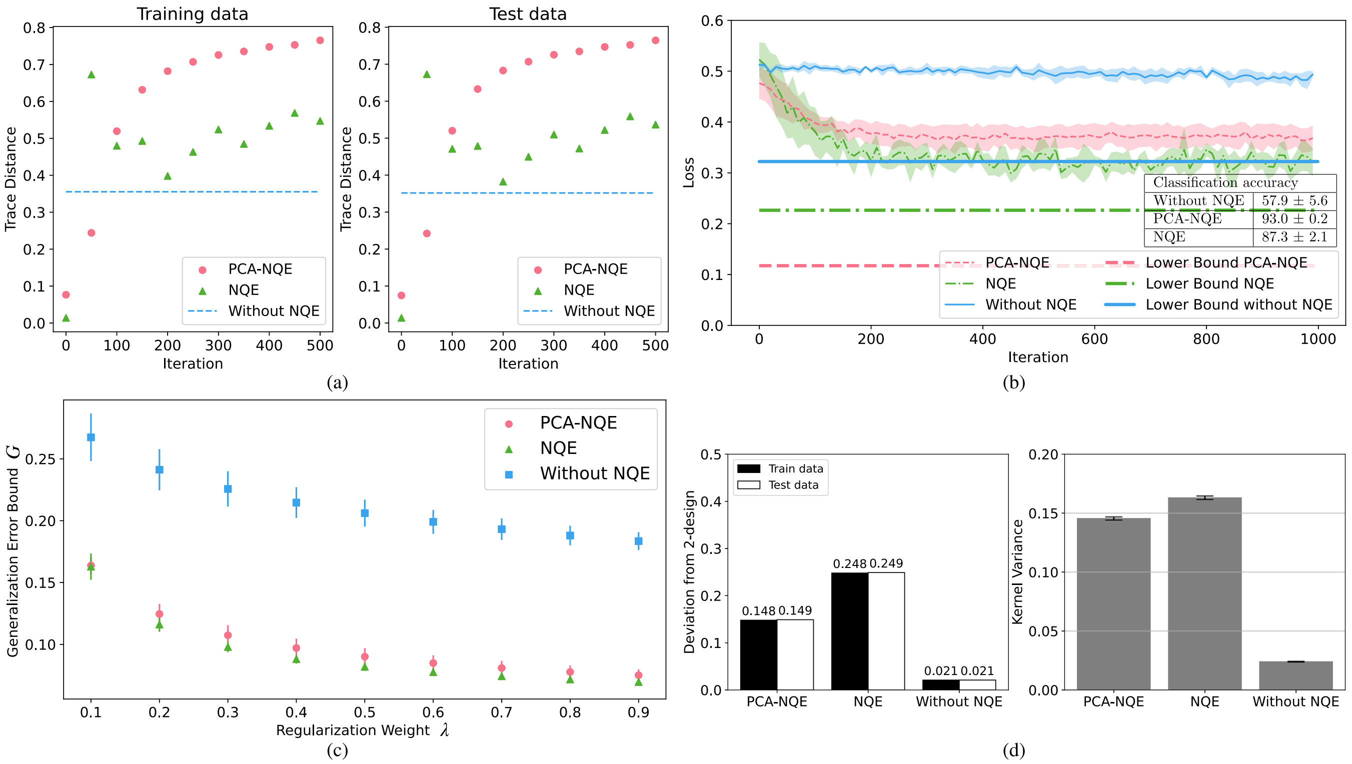


FIG. 11. Results for the additional experiments with Fashion-MNIST datasets using noisy simulations replicating the characteristics of the ibmq_vigo quantum computer. Descriptions for (a)–(d) are identical to those provided in the caption of Fig. 10.

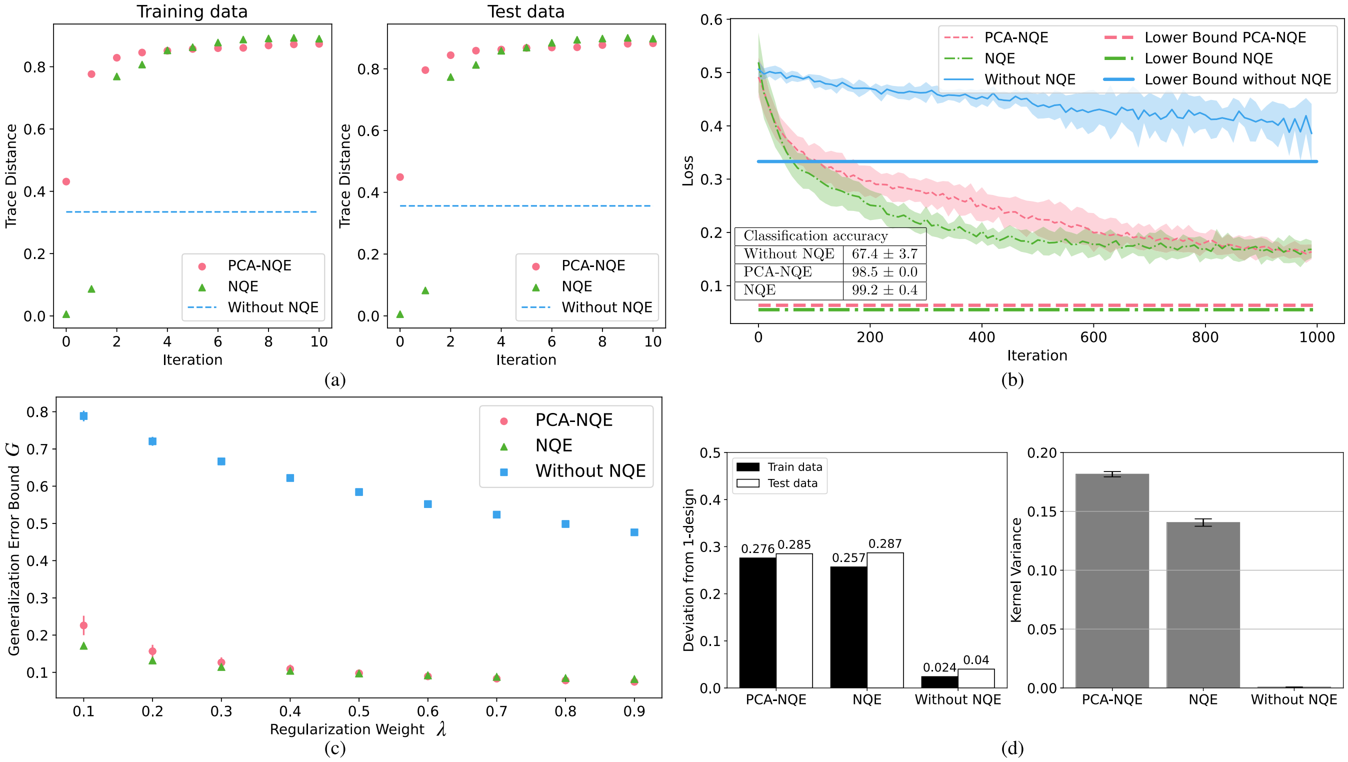
four-qubit experiments on both numerical simulation and IBM quantum hardware experiments. In this section, we illustrate the effectiveness of NQE methods in larger quantum systems, specifically those comprising 8 and 12 qubits. Consistent with the main text, we compare the effectiveness of NQE methods on trace distance history [Figs. 12(a), 13(a)], QCNN loss histories and classification accuracies [Figs. 12(b), 13(b)], upper bound of generalization in QKM [Figs. 12(c), 13(c)], and expressibility and trainability [Figs. 12(d), 13(d)] in 8 and 12 qubit setups. Due to limited computational resources and access to IBM quantum hardware, the experiments are only conducted with numerical simulation. The experimental methods are identical to the ones of Sec. II C 1 of the main text, except the expressibility is computed by deviation from 1-design (instead of 2-design) due to limited computational resources.

The experimental results further validate that NQE methods are effective at enhancing QML algorithms on larger quantum systems. Application of NQE yielded improvements in training loss, classification accuracy, generalization upper bounds, and trainability metrics. Unlike in four-qubit experiments, training losses did not reach their theoretical minima. This indicates that QCNN circuits did not accurately approximate the optimal Helstrom measure. Such behavior is expected as QCNN circuits are inexpressive due to its parameter-sharing and nearest-neighbor variational ansatz constraints. Nonetheless, NQE application significantly enhanced training loss, underscoring its utility in advancing QML algorithm performance. Additionally, in scenarios where NQE is not employed, there is a notable decline in trainability with 8 and 12 qubits [Figs. 12(d), 13(d)], as opposed to those with 4 qubits (Fig. 6). This reduction in trainability is evidenced by increased expressibility and kernel variance. Conversely, in systems utilizing NQE models, both expressibility and kernel variance maintain consistent levels, demonstrating enhanced trainability even in larger-scale quantum circuits.

APPENDIX F: ADDITIONAL ANALYSIS ON EXPRESSIBILITY AND TRAINABILITY

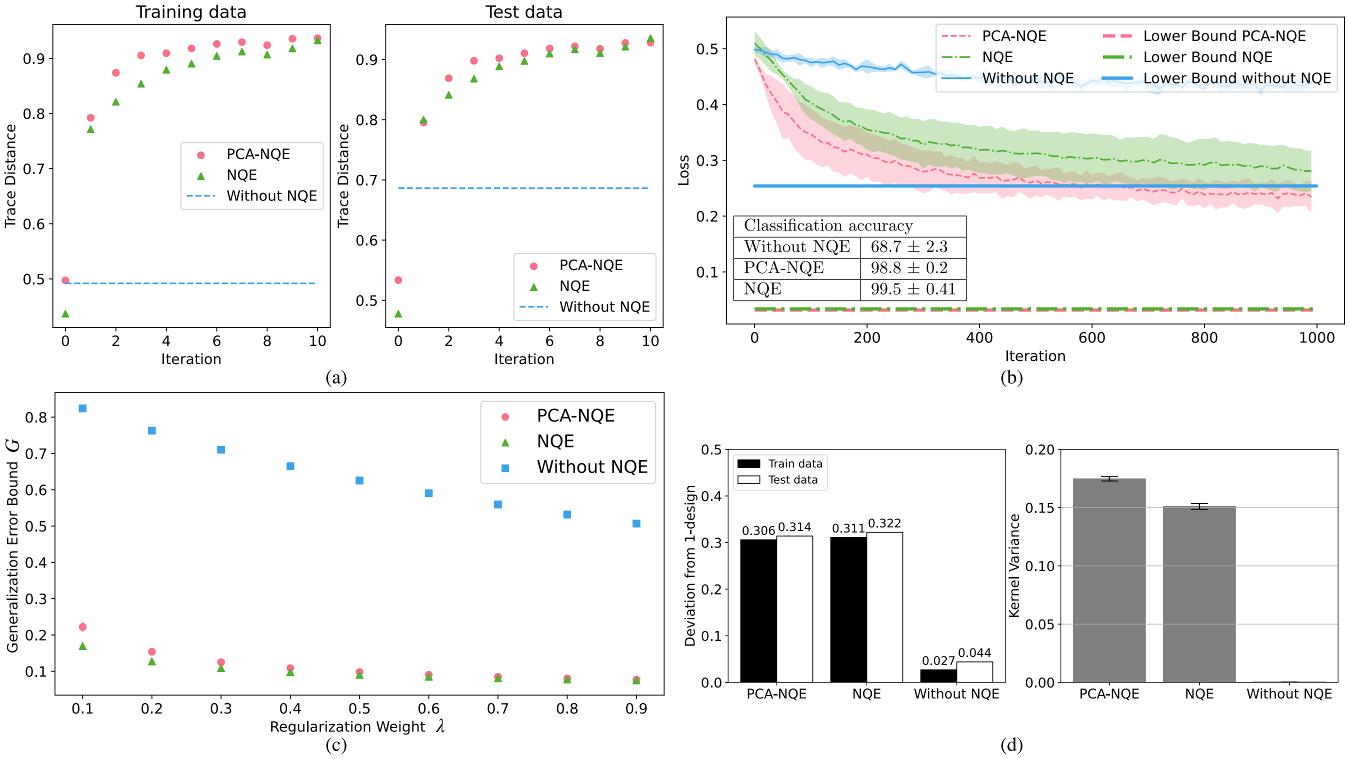
Let N be the number of samples drawn from a data distribution $\mathcal{D}$, and let $d = \dim(\mathrm{span}\{|x_i\rangle\}_{i=1,\dots,N}) \leqslant N$ represents the dimension of the subspace spanned by the quantum state representation of training data, determined by quantum embedding [32]. The dimension d serves as an indicator of the expressibility of the given quantum embedding and the complexity of the machine learning (ML) model necessary for learning from the quantum-embedded data. This relationship can be rigorously demonstrated in the context of the quantum kernel method (QKM) as follows. The QKM can be employed to learn a quantum model defined by $f(x) = \mathrm{Tr}[OU\rho(x)U^\dagger]$, where $\rho(x)$ is the density matrix representation of the data encoded as a quantum state, from N samples drawn from a data distribution $\mathcal{D}$. The expected risk (i.e., prediction error) of the prediction model $h(x)$ constructed from the learning procedure is bounded as

$$\mathbb{E}_{x\in\mathcal{D}}|h(x) - f(x)| \leqslant c\sqrt{\frac{\min[d, \mathrm{Tr}(O^2)]}{N}}, \tag{F1}$$



Classification accuracy	
Without NQE	67.4 ± 3.7
PCA-NQE	98.5 ± 0.0
NQE	99.2 ± 0.4

FIG. 12. Results for the additional experiments with eight qubit noiseless simulations. Descriptions for (a)–(c) are identical to those provided in the caption of Fig. 10. (d) Left: A comparative analysis of expressibility with and without NQE models. The deviation from unitary 1-design is depicted, where a smaller deviation indicates higher expressibility. The deviation is derived from 12 665 (2115) MNIST training (test) data results. Right: A comparative analysis of the variance of quantum kernel elements with and without NQE models. The variance was computed from the off-diagonal elements of the quantum kernel matrix K^Q, constructed from 1000 samples of MNIST datasets. The mean and one standard deviation from five independent iterations are shown.



Classification accuracy	
Without NQE	68.7 ± 2.3
PCA-NQE	98.8 ± 0.2
NQE	99.5 ± 0.41

FIG. 13. Results for the additional experiments with 12 qubit noiseless simulations. Descriptions for (a)–(d) are identical to those provided in the caption of Fig. 12.

TABLE I. The rank of the quantum kernel matrix (d) evaluated with and without NQE. Four different binary datasets, each containing 800 samples, were generated from two pairs of MNIST classes. For each pair of datasets, two PCA configurations were used to reduce the number of features to four and eight, respectively.

		d					
Dataset			NQE (5 runs)				
Classes	Input size	Untrained	1	2	3	4	5
0 and 1	4	175	15	14	19	15	13
	8	800	14	16	17	20	20
3 and 8	4	175	38	38	40	31	27
	8	800	55	54	87	50	127

where $c > 0$ is a constant. The quantity of interest here is d because it is affected by NQE and $\mathrm{Tr}(O^2)$ grows exponentially with the number of qubits in many cases (e.g., Pauli observables). This equation implies that the hardness of the learning problem depends on the quantum embedding that represents the set of N training data.

We conducted numerical experiments to assess the effectiveness of NQE in reducing the dimension d. To evaluate this, we employed a binary classification task involving the discrimination of digits 0 and 1 or 3 and 8 from the MNIST dataset. We computed the rank of the quantum kernel matrix both with and without NQE, and the dimension d was determined as $d = \mathrm{rank}(K^Q)$. The results, presented in Table I, clearly demonstrate that NQE effectively reduces d. As d represents the effective dimension of the quantum training data used for model training, its reduction indicates that simpler ML models with NQE can achieve comparable performance to more complex models applied to the original data without NQE. The findings suggest that by using NQE, we can constrain quantum embedding to those that allow for large data separability. This reduction in the expressibility of quantum embedding, conversely, improves the trainability of the model.

[1] P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd, Quantum singular-value decomposition of nonsparse low-rank matrices, Phys. Rev. A **97**, 012327 (2018).

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

[3] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411.

[4] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[5] 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).

[6] S. Aaronson and A. Arkhipov, The computational complexity of linear optics, in *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, USA, 2011), pp. 333–342.

[7] A. P. Lund, M. J. Bremner, and T. C. Ralph, Quantum sampling problems, boson sampling and quantum supremacy, npj Quantum Inf. **3**, 15 (2017).

[8] A. W. Harrow and A. Montanaro, Quantum computational supremacy, Nature (London) **549**, 203 (2017).

[9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[10] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[11] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins *et al.*, Quantum computational advantage with a programmable photonic processor, Nature (London) **606**, 75 (2022).

[12] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[13] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[14] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. **4**, 043001 (2019).

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

[16] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[17] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature (London) **567**, 209 (2019).

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

[19] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Phys. Rev. A **103**, 032430 (2021).

[20] M. C. Caro, E. Gil-Fuster, J. J. Meyer, J. Eisert, and R. Sweke, Encoding-dependent generalization bounds for parametrized quantum circuits, Quantum **5**, 582 (2021).

[21] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, Exponential concentration and untrainability in quantum kernel methods, arXiv:2208.11060.

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

[23] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, New York, 2011).

[24] M. M. Wilde, *Quantum Information Theory* (Cambridge University Press, Cambridge, 2013).

[25] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, Quantum embeddings for machine learning, arXiv:2001.03622.

[26] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, Peter-Jan H. S. Derks, P. K. Faehrmann, and J. J. Meyer, Training quantum embedding kernels on near-term quantum computers, Phys. Rev. A **106**, 042431 (2022).

[27] 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).

[28] J. Bae and L.-C. Kwek, Quantum state discrimination and its applications, J. Phys. A: Math. Theor. **48**, 083001 (2015).

[29] K. Siudzińska, S. Chakraborty, and D. Chruściński, Interpolating between positive and completely positive maps: A new hierarchy of entangled states, Entropy **23**, 5 (2021).

[30] J. R. Glick, T. P. Gujarati, A. D. Corcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, Covariant quantum kernels for data with group structure, arXiv:2105.03406.

[31] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Quantum fingerprinting, Phys. Rev. Lett. **87**, 167902 (2001).

[32] 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).

[33] S. Park, D. K. Park, and J.-K. K. Rhee, Variational quantum approximate support vector machine with inference transfer, Sci. Rep. **13**, 3288 (2023).

[34] K. Batra, K. M. Zorn, D. H. Foil, E. Minerali, V. O. Gawriljuk, T. R. Lane, and S. Ekins, Quantum machine learning algorithms for drug discovery applications, J. Chem. Inf. Model. **61**, 2641 (2021).

[35] S. Mensa, E. Sahin, F. Tacchino, P. K. Barkoutsos, and I. Tavernelli, Quantum machine learning framework for virtual screening in drug discovery: A prospective quantum advantage, Mach. Learn.: Sci. Technol. **4**, 015023 (2023).

[36] S. L. Wu, S. Sun, W. Guan, C. Zhou, J. Chan, C. L. Cheng, T. Pham, Y. Qian, A. Z. Wang, R. Zhang, M. Livny, J. Glick, P. K. Barkoutsos, S. Woerner, I. Tavernelli, F. Carminati, A. D. Meglio, A. C. Y. Li, J. Lykken, P. Spentzouris, Samuel Yen-Chi Chen, S. Yoo, and T.-C. Wei, Application of quantum machine learning using the quantum kernel algorithm on high energy physics analysis at the LHC, Phys. Rev. Res. **3**, 033221 (2021).

[37] T. Li, Z. Yao, X. Huang, J. Zou, T. Lin, and W. Li, Application of the quantum kernel algorithm on the particle identification at the BESIII experiment, J. Phys.: Conf. Ser. **2438**, 012071 (2023).

[38] M. Pistoia, S. F. Ahmad, A. Ajagekar, A. Buts, S. Chakrabarti, D. Herman, S. Hu, A. Jena, P. Minssen, P. Niroula, A. Rattew, Y. Sun, and R. Yalovetzky, Quantum machine learning for finance ICCAD special session paper, in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)* (IEEE Press, Munich, Germany, 2021), pp. 1–9.

[39] M. Grossi, N. Ibrahim, V. Radescu, R. Loredo, K. Voigt, C. V. Altrock, and A. Rudnik, Mixed quantum–classical method for fraud detection with quantum feature selection, IEEE Trans. Quantum Eng. **3**, 1 (2022).

[40] Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto, Analysis and synthesis of feature map for kernel-based quantum classifier, Quantum Mach. Intell. **2**, 9 (2020).

[41] R. LaRose and B. Coyle, Robust data encodings for quantum classifiers, Phys. Rev. A **102**, 032420 (2020).

[42] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).

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

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

[45] I. F. Araujo, D. K. Park, T. B. Ludermir, W. R. Oliveira, F. Petruccione, and A. J. D. Silva, Configurable sublinear circuits for quantum state preparation, Quantum Inf. Proc. **22**, 123 (2023).

[46] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, Absence of barren plateaus in quantum convolutional neural networks, Phys. Rev. X **11**, 041011 (2021).

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

[48] J. Kim, J. Huh, and D. K. Park, Classical-to-quantum convolutional neural network transfer learning, Neurocomputing **555**, 126643 (2023).

[49] H. Oh and D. K. Park, Quantum support vector data description for anomaly detection, arXiv:2310.06375.

[50] Y. LeCun, C. Cortes, and C. J. Burges, MNIST handwritten digit database, ATT Labs (Online); http://yann.lecun.com/exdb/mnist, 2, 2010.

[51] S. Jerbi, L. J. Fiderer, H. P. Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Quantum machine learning beyond kernel methods, Nat. Commun. **14**, 517 (2023).

[52] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms, arXiv:1708.07747.

[53] O. Berezniuk, A. Figalli, R. Ghigliazza, and K. Musaelian, A scale-dependent notion of effective dimension, arXiv:2001.10872.

[54] A. Abbas, D. Sutter, A. Figalli, and S. Woerner, Effective dimension of machine learning models, arXiv:2112.04807.

[55] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, PRX Quantum **3**, 010313 (2022).

[56] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Average-case complexity versus approximate simulation of commuting quantum computations, Phys. Rev. Lett. **117**, 080501 (2016).

[57] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, K. McKiernan, J. J. Meyer, Z. Niu, A. Száva, and N. Killoran, Pennylane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968.

[58] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, Adv. Quantum Technol. **2**, 1900070 (2019).

[59] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, R. Halavati, M. Y. Niu, A. Zlokapa

et al., Tensorflow quantum: A software framework for quantum machine learning, arXiv:2003.02989.

[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] W. Jiang, J. Xiong, and Y. Shi, A co-design framework of neural networks and quantum circuits towards quantum advantage, Nat. Commun. **12**, 579 (2021).

[62] M. Liu, J. Liu, R. Liu, H. Makhanov, D. Lykov, A. Apte, and Y. Alexeev, Embedding learning in hybrid quantum-classical neural networks, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Press, Broomfield, CO, USA, 2022), pp. 79–86.

[63] M. Larocca, F. Sauvage, F. M. Sbahi, G. Verdon, P. J. Coles, and M. Cerezo, Group-invariant quantum machine learning, PRX Quantum **3**, 030341 (2022).

[64] J. J. Meyer, M. Mularski, E. Gil-Fuster, A. A. Mele, F. Arzani, A. Wilms, and J. Eisert, Exploiting symmetry in variational quantum machine learning, PRX Quantum **4**, 010328 (2023).

[65] V. Vapnik, *The Nature of Statistical Learning Theory* (Springer Science & Business Media, New York, 1999).

[66] M. Incudini, F. Martini, and A. Di Pierro, Structure learning of quantum embeddings, arXiv:2209.11144.

[67] H. Kwon, H. Lee, and J. Bae, Feature map for quantum data: Probabilistic manipulation, arXiv:2303.15665.

[68] H. Tak, I. F. Araujo, and D. K. Park, qDNA-yonsei/neural-quantum-embedding: v1.0.0, Zenodo (2024), https://doi.org/10.5281/zenodo.12817965.

[69] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[70] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. **12**, 2825 (2011).

[71] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[72] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, Quantum computing with Qiskit, arXiv:2405.08810.