Unified framework for quantum classification

Nhat A. Nghiem,[1] Samuel Yen-Chi Chen,[2] and Tzu-Chieh Wei[3,1]
[1]*Department of Physics and Astronomy, State University of New York at Stony Brook, Stony Brook, New York 11794-3800, USA*
[2]*Computational Science Initiative, Brookhaven National Laboratory, Upton, New York 11973, USA*
[3]*C. N. Yang Institute for Theoretical Physics, State University of New York at Stony Brook, Stony Brook, New York 11794-3840, USA*

(Received 4 November 2020; accepted 21 May 2021; published 16 July 2021)

Quantum machine learning is an emerging field that combines machine learning with advances in quantum technologies. Many works have suggested great possibilities of using near-term quantum hardware in supervised learning. Motivated by these developments, we present an embedding-based framework for supervised learning with trainable quantum circuits. We introduce both explicit and implicit approaches. The aim of these approaches is to map data from different classes to separated locations in the Hilbert space via a parametrized quantum circuit. We will show that the implicit approach is a generalization of a recently introduced strategy, so-called *quantum metric learning*. Furthermore, we discuss an intrinsic connection between the explicit approach and those previously proposed quantum classification models. The implicit and explicit approaches, together, provide a unified framework for quantum classification. The utility of our framework is demonstrated by our noise-free and noisy numerical simulations. Moreover, we have conducted classification testing with both implicit and explicit approaches using several IBM Q devices.

DOI: 10.1103/PhysRevResearch.3.033056

I. INTRODUCTION

Quantum computation has been intensively studied over the past few decades and is expected to outperform its classical counterpart in certain computational tasks [1–3]. In this novel approach for computation, information is stored in the quantum states of an appropriately chosen and designed physical system, which resides in a complex Hilbert space $\mathcal{H}$, and quantum bits (qubits) are used as the underlying building blocks and processing units. The power of a quantum computer is in its ability to store and process information coherently in the tensor-product Hilbert space [1], with entanglement being a characteristic by-product or even a potential resource for information processing [4,5]. Quantum computations have been shown to provide dramatic speedup in solving some important computational problems, such as factorization of a large number via Shor's algorithm [6] and the unstructured search using Grover's algorithm [7], which are two prominent examples among many that have been discovered.

At the same time, machine learning (ML) has become a powerful tool in modern computation. For example, ML has been successful in computer vision [8–10], natural language processing [11], and drug discovery [12]. Building on this history, a natural application of quantum computers also may provide substantial speedup [13–16]. Several previous works have revealed potential quantum advantages in the field of unsupervised learning [17–21]. For example, in Ref. [20], the authors provide quantum algorithms for clustering problems, which could, in principle, yield an exponential speedup. In Ref. [21], the authors introduce a quantum version of k-means clustering, namely, q means, and present an efficient quantum procedure.

Using quantum computation in supervised learning also has garnered increased attention [22–24]. For example, the authors of Ref. [25] present a quantum version of support vector machines (SVMs) that shows possible exponential speedup. For near-term applications, variational strategies have been proposed to classify real-world data [22,26–29]. Classification is among the standard problems in supervised learning [30,31], and variational methods using short-depth quantum circuits with trainable parameters have given rise to a quantum-classical hybrid optimization procedure. Such frameworks have proven to be capable of performing complex classification tasks [26,27,29,32–34], and many more likely will appear. It is probable that such variational methods will be able to learn complex representations while still being robust to noise in so-called noisy intermediate-scale quantum (NISQ) devices [29,35–37].

"Traditional" quantum classification models rely on the encoding of classical data x into some quantum state $|\psi(x)\rangle$. This state then undergoes a parametrized quantum circuit $U(\theta)$. At the end, the state is measured. The outcome of the measurement usually is interpreted as the output of the learning model. Although the structures of previously proposed works [15,27–29] appear similar, the motivation underlying their strategies seems varied. For example, the quantum circuit learning algorithm proposed in Ref. [29] is inspired by classical neural networks. Meanwhile, in Refs. [28,38], the authors exploit and formally establish the connection between

quantum computation and the kernel method, where they interpret the step of encoding classical data x into the quantum state as a quantum feature map. Thus a clear picture emerges: Classical data x are embedded into some quantum state $|x\rangle$, i.e., a data point in Hilbert space $\mathcal{H}$. (This space also is called the *quantum feature space*, analogous to the feature space in classical ML.) Then, a decision boundary is learned by training the variational circuit to adapt the measurement basis, which is analogous to the classical approach where a decision boundary is learned to separate classes.

Metric learning is a well-known method in the classical ML context [39]. The aim is to learn an appropriate distance function over data points. This method recently has been extended to the quantum context by Lloyd *et al.* [40]. Instead of focusing on training the variational layer that adapts the measurement basis, the authors propose to train the embedding circuit and proffer a remarkable notion of "well separation" of data points. Per their argument, a significant amount of computational power spent on processing classically embedded data can be eased using such a strategy.

Aside from such an advantage, we pose that the ability to use a quantum circuit to represent data in a complex Hilbert space and the idea of well separation have further remarkable consequences. Our framework is built upon the geometrical aspect of the quantum classifier, where we consider the whole quantum circuit as a means to map data from input space to some Hilbert space. Therefore our framework conceptually involves all the previous approaches [15,27–29,40]. We then argue that previous traditional quantum classification methods [15,27–29] essentially achieve a certain well separation of data points, which implies that quantum classification models could be conceptually merged into a common framework. Here, we provide a unified, generic framework and categorize approaches into two different types, *implicit* and *explicit*, which are described in detail, backed up by numerical simulations, and tested on real quantum devices. The map from the data input space $\mathcal{X}$ to the featured Hilbert space $\mathcal{H}$ is enacted by a parametrized quantum circuit. The goal is to train the embedding circuit to produce clusters of data from different classes. In the *implicit* approach, the "centers" of these clusters are random. In the *explicit* approach, the cluster centers are constrained to lie in or nearby some predetermined subspaces of the Hilbert space. We show that both explicit and implicit approaches exhibit promising classification ability. Particularly, the method proposed by Lloyd *et al.* [40] is a binary version of the implicit approach. We point out that the explicit approach can conceptually unify traditional quantum classification methods, such as those of Refs. [15,27,29] (see Sec. II C 2). These two approaches then constitute our unified framework for quantum classification.

The following summarizes the contributions of this work:

(i) We introduce two approaches for quantum classification, implicit and explicit, that constitute a generic embedding-based framework.

(ii) We show that the implicit approach is the generalization of the metric quantum learning method proposed in Ref. [40]. Such generalization allows us to manage the multiclass classification problem. The number of separated classes (or labels) is independent of qubits used in the quantum circuit. In principle, this approach can work even with a single qubit. Therefore it sheds light on constructing a universal quantum classifier.

(iii) We demonstrate that the explicit approach can conceptually unify other models for quantum classification. Along with the generalization provided by the implicit approach, our work provides a complete unification of quantum classification frameworks.

(iv) We implement both learning approaches on NISQ devices and compare the results with noisy simulations. We demonstrate the framework's success and clarify the cases where the results on real devices and noisy simulations do not agree well.

The structure of the paper is as follows: Sec. II A presents the main conceptual tool of our framework. In Secs. II B and II C, we discuss the implicit and explicit approaches. In Sec. III, we present results from numerical experiments and runs from real devices, and provide numerical evidence that the implicit approach is especially robust with small training size. Some discussions regarding our framework's prospects in the near-term era are presented in Sec. IV. Section V concludes the primary work. Appendix B provides an additional example to illustrate the unification. Appendix C discusses a remarkable consequence of focusing on the embedding part instead of the measuring part in quantum supervised learning (QSL), which could avoid a systematic issue of misclassification of the one-versus-all strategy.

II. GENERAL FRAMEWORK

A. Basic concept

We first introduce the basic concept of classification, which can be illustrated by a simple map:

$$x \longrightarrow \vec{f}(x,\theta), \tag{1}$$

where

$$\vec{f}(x,\theta) = \begin{bmatrix} f_0 \\ \vdots \\ f_i \\ \vdots \\ f_{L-1} \end{bmatrix}, \tag{2}$$

$\in \mathbb{R}^L$, is called a classifying vector of some input data x and θ refers to the network's or circuit's parameters. $\{f_i\}$ is generally of the form

$$f_i = \langle x|\mathcal{M}_i|x\rangle = \mathrm{Tr}(|x\rangle\langle x|\,\mathcal{M}_i), \tag{3}$$

where $|x\rangle$ is the corresponding quantum state of classical data x and $\mathcal{M}_i$, in general, is some Hermitian operator. The value of f_i depends on circuit parameters θ and the input feature x.

In the supervised learning problem with L separated labels, we are given a training set together with corresponding labels $X \times Y = \{x, i\}$, where $i \in \{0, 1, \ldots, L-1\}$ is the label of the data point x. We need to predict the label for some other unseen data x. In the quantum setting, we simply use its representation by a quantum state $|x\rangle$ instead of the classical data x. The number of components $\{f_i\}$, or equivalently the dimension of $\vec{f}(x,\theta)$, is denoted by L. The value f_i (for convenience, assumed to be in the range $0 \leqslant f_i \leqslant 1$) quantifies the likelihood that some input x have any of labels $\{i\}$. In this sense, the

method is somehow similar to the classical neural network, where the information is fed forward from the input layer to the output layer. There have been numerous works that explore the relation between quantum computation and neural networks [26,27,41,41–44]. The key relation extends from the quantum gates, as they carry out unitary transformation on the input quantum state, which is a vector in some Hilbert space $\mathcal{H}$. In the graphical representation (distinctively illustrated in Ref. [27]), the action of a quantum gate on the input state $|\psi\rangle$ produces an output state $|\rho\rangle$ and can be represented as a fully connected two-layer network. Hence a full quantum circuit generally can be represented by such a fully connected network with a certain number of layers. Measuring quantum states then corresponds to a nonlinear activation function.

Thus, to make a prediction, we "forward" x to such a classifying vector $\vec{f}(x,\theta)$ and assign to it a label according to the highest value of $\{f_i\}$. The accuracy of correct assignment depends on circuit parameters θ. Now, we provide a strategy to train the circuit. For each label i, assume that there are N_i training points with such a label and there are a total of N data points, where $N=\sum_i N_i$. Let $\vec{y}_i$ be the real, so-called *label vector*, of class i (which has dimension L) with components $\{y_i^j\}_{j=1}^L=\delta_{ij}$, where δ_{ij} is the Kronecker delta function. Let $\vec{f}_i^j$ be the classifying vector of the jth data. We minimize the following cost or so-called loss function,

$$C=\frac{1}{L}\sum_{i=1}^{L}\frac{1}{N_i}\sum_{j=1}^{N_i}\left|\vec{f}\left(x_i^j,\theta\right)-\vec{y}_i\right|, \tag{4}$$

where x_i^j is the jth data point in class i.

State overlaps. Given two *pure* quantum states represented by density matrices $\rho\equiv|\rho\rangle\langle\rho|$ and $\phi\equiv|\phi\rangle\langle\phi|$, the overlap, i.e., a similarity measure, on these two states is given by $\mathrm{Tr}(\rho\phi)=|\langle\rho|\phi\rangle|^2$. State overlap plays an important role in our subsequent construction of the implicit and explicit approaches. In the general case of mixed states, the SWAP test quantum procedure [1] can be used to evaluate $\mathrm{Tr}(\rho\phi)$ up to an additive error ϵ. In the special case of pure states, the *inversion test* [40] can be used to evaluate the overlaps between two quantum states $|\langle\phi|\cdot|\rho\rangle|^2$, provided that the circuit U to create either $|\phi\rangle$ or $|\rho\rangle$ can be efficiently inverted. Both schemes require only shallow circuits. In our subsequent experiments, we also will implement both schemes for classification on real devices.

B. Implicit approach

1. Construction

In the implicit approach, the data from the same class, after going through the quantum circuit $\Phi(x,\theta)$, produce clusters (closed data points) in the Hilbert space $\mathcal{H}$. Clusters corresponding to different classes should become maximally separated after minimizing the cost function (see Figs. 1 and 2).

To describe our supervised learning problem, we assume that for each label i, there are N_i training points that will be transformed to quantum states $\{|x_i^j\rangle\}_{j=1}^{N_i}$. The formula for $\mathcal{M}_i$ in this case is

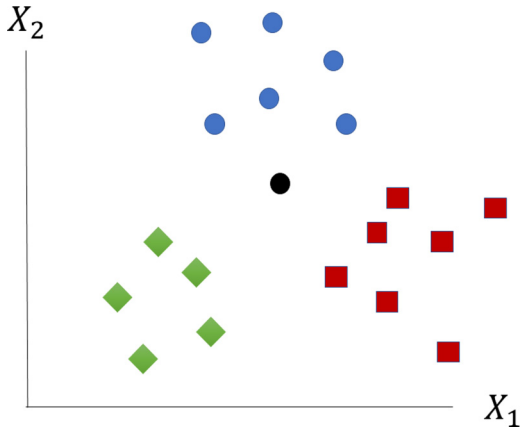

FIG. 1. An example supervised learning problem with $L=3$ labels. The black circle is some unseen data.

$$\mathcal{M}_i=\frac{1}{N_i}\sum_j|x_i^j\rangle\langle x_i^j|, \tag{5}$$

which is exactly an ensemble of quantum states: $\sigma_i=\frac{1}{N_i}\sum_j|x_i^j\rangle\langle x_i^j|$. This ensemble may be interpreted as the collection of the corresponding training points from class i on $\mathcal{H}$, and it can be obtained by sampling from the training set $\{x_i^j\}|_{j=1}^{N_i}$. The classifying vector $\vec{f}(x,\theta)$ now becomes

$$\begin{bmatrix}\mathrm{Tr}(|x\rangle\langle x|\sigma_0)\\ \mathrm{Tr}(|x\rangle\langle x|\sigma_1)\\ \cdot\\ \cdot\\ \mathrm{Tr}(|x\rangle\langle x|\sigma_{L-1})\end{bmatrix}. \tag{6}$$

We will focus on optimizing those quantities $\{f_i\}$ and using the values to assign a label to x.

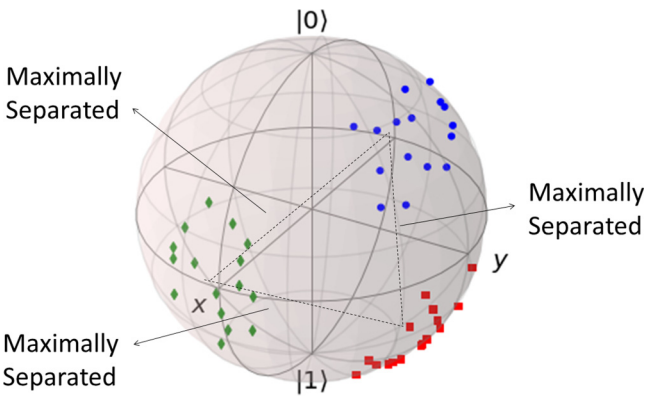

FIG. 2. Illustration of the implicit approach. After the training procedure, the "distance" between any two clusters (represented by dashed lines) becomes maximal. The "center" of each cluster is not fixed as the training process will move it so as to produce maximally separated clusters. Note that the data points in this picture are not related to the Iris data set in our subsequent experiment or to the data points in Fig. 3.

After applying Eq. (4), the cost function becomes

$$C = 1 - \frac{1}{L}\sum_{i=1}^{L} \mathrm{Tr}\sigma_i^2 + \frac{2}{L}\sum_{i<j} \mathrm{Tr}(\sigma_i\sigma_j). \tag{7}$$

Each cross term,

$$\mathrm{Tr}(\sigma_i\sigma_j) = \frac{1}{N_iN_j}\sum_{k=1}^{N_i}\sum_{p=1}^{N_j} \left|\langle x_i^k| \cdot |x_j^p\rangle\right|^2, \tag{8}$$

is a sum of the modulus square of overlaps. Hence, in the training procedure, we can use either the SWAP or inversion test to evaluate the cost.

Consider the binary classification problem ($L = 2$). The cost function is simply

$$C = 1 - \tfrac{1}{2}\left(\mathrm{Tr}\sigma_1^2 + \mathrm{Tr}\sigma_2^2\right) + \mathrm{Tr}\sigma_1\sigma_2 = 1 - \tfrac{1}{2}\mathrm{Tr}(\sigma_1 - \sigma_2)^2. \tag{9}$$

The optimization will be to minimize the Hilbert-Schmidt distance between two data clusters, which is highlighted in Ref. [40]. Therefore we have shown that this implicit approach is a generalization of the binary method discussed in Ref. [40].

2. Training with QRAM

If quantum random access memory (QRAM) [45] is available, the cost for the training and testing procedure will be reduced by a factor of approximately $\sum_{i\leqslant j} N_iN_j$ and $\sum_i N_i$, respectively, where N_i is the number of data points in each training set i. The calculation of the cost function and data classification can be done in time $\mathcal{O}(1)$. The data can be loaded corresponding to class i to a quantum state $|\psi_i\rangle = \frac{1}{\sqrt{N_i}}\sum_{j=1}^{N_i} |\text{“}\vec{x_i^j}\text{”}\rangle\,|j\rangle$, where the index j represents the address of the memory where x_i^j is residing. We use $|\text{“}\vec{x_i^j}\text{”}\rangle$ to denote the classical data (not the embedded feature state) loaded to a quantum register.

A third register is initialized in $|0\cdots0\rangle$ and is used to implement the quantum feature. The application of $R_y(x)$ (refer to Fig. 6) on this register with its argument being x_i^j can be done by performing the rotation conditioned on the first register with classical values x_i^j, i.e., a conditional rotation $c - R_y(x)$. In this way, at the expense of a more complicated circuit to implement the conditional rotation, we obtain the entangled state $|\psi_i'\rangle = \frac{1}{\sqrt{N_i}}\sum_{j=1}^{N_i} \otimes|\text{“}\vec{x_i^j}\text{”}\rangle \otimes |j\rangle \otimes |\Phi(x_i^j,\theta)\rangle$. Tracing over the first and second registers (i.e., without doing anything on them afterwards), the third register is in the state $\sigma_i = \sum_j |\Phi(x_i^j,\theta)\rangle\langle\Phi(x_i^j,\theta)|/N_i$. With QRAM, we do not need to repeat the circuits (with different rotations) to sample every data point individually from the training set to obtain an effective σ_i (about N_i times). However, a practical implementation of QRAM is currently not available.

Using the controlled-SWAP gate on this third register and another embedded state $|x\rangle$ for an unknown data point (i.e., the SWAP test), we can directly measure their fidelity $\langle x|\sigma_i|x\rangle$. However, computing the pairwise overlaps in Eq. (8) without QRAM will require using the SWAP test N_iN_j times. As such, it would be useful to design an efficient quantum subroutine of low-depth circuits to evaluate the cost in Eq. (7), directly exploiting QRAM and reducing the iterative evaluation steps.

3. Classification over a large number of classes and universal quantum classifier

In most current quantum ML models, the measurement outcome usually is interpreted as the outcome of learning models. In a binary classification problem, one-qubit measurement suffices to classify the data as there are only two possible outcomes, and one can draw inferences from such a measurement. For example, if the probability of obtaining class zero $\mathcal{P}(\text{outcome} = 0) \geqslant 0.5$, we then assign the data to class 0. Otherwise, we assign them to class 1. For multiclass classification, multiqubit measurement needs to be employed. A circuit with k qubits can classify up to 2^k different labels. Our implicit approach surpasses this because we only aim to get the classifying vector $\vec{f}$. The dimension of $\vec{f}$, or the number of classes, can be quite large. Real-world supervised learning problems may contain overwhelmingly numerous classes, such as in face recognition. Thus our framework may prove useful for these practical tasks.

Recall that our framework concerns the geometrical aspect of the quantum classifier, where the aim is to maximally distribute data points on Hilbert space. Even a single qubit is enough to execute this implicit approach, as it provides enough space for the separation, regardless of how many classes or labels are in the supervised learning problem (see Fig. 2). We further remark that a single qubit is enough for embedding purposes, as it could map data with arbitrary dimension to the Hilbert space of dimension 2.

Relevant work has been carried out in Ref. [46], showing that a single qubit is sufficient to construct a universal classifier and is able to deal with multidimensional input data and multilabel output of supervised learning problems. To handle multidimensional input, the authors propose a data reuploading strategy. They achieve the multiclassification by introducing a bias term in the measurement outcome. Our approach differs from that of Ref. [46] because we use the parametrized quantum circuit to represent the data in the Hilbert space and exploit its “vastness.” Data from different classes are “aligned” in separate locations. To handle multidimensional data, one may opt to follow the same strategy as in Ref. [46] or engage a different embedding routine. We do not attempt to provide a specific embedding ansatz here, since it is up to one’s choice.

C. Explicit approach

1. Construction

The implicit approach emphasizes training the embedding circuit to produce separated clusters on $\mathcal{H}$. However, the centers of these clusters are somewhat random. As long as relative distances among these clusters are maximal and those among data points within the same cluster are minimal, the method achieves its goal.

With the explicit approach, the cluster “positions” are designed to be fixed and separated into orthogonal subspaces (Fig. 3). The main intuition is that with enough qubits, the Hilbert space is large enough and can accommodate many

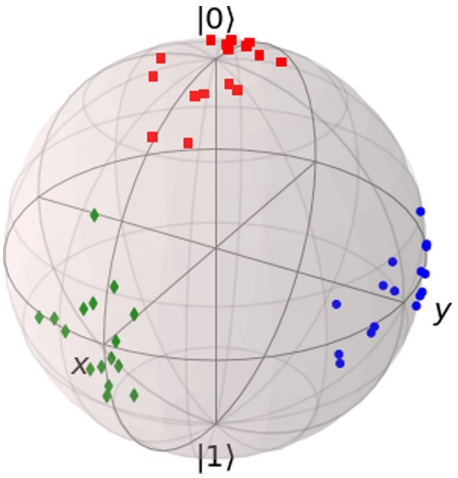

FIG. 3. Illustration of the explicit approach. The "center" of each cluster is fixed. For class 0 (red points), the position of the center is $v_0 = (0, 0, 1)$. For class 1 (green points), the position is $v_1 = (1, 0, 0)$. For class 2 (blue points), the position is $v_2 = (0, 1, 0)$. Notably, this separation is not exactly the same as desired in Eq. (10) because the Hilbert space associated with a single qubit has dim = 2, and it only can be decomposed into, at most, two orthogonal subspaces. Nevertheless, this figure illustrates the idea of the explicit approach.

smaller subspaces where data clusters can reside. These subspaces are well defined and well separated. If the data from different classes are "approximately" mapped to their proper subspaces, they are well separated by construction. The approximation here means that the embedded data might not exist completely within the desired subspace, instead possibly only in its vicinity. Then, we can "measure" the distance from a data point in $\mathcal{H}$ to different subspaces. Thus classification of such a data point can be done accordingly.

Again, consider the supervised learning problem with L labels; we decompose $\mathcal{H}$ into

$$\mathcal{H} = H_0 \oplus H_1 \oplus \cdots \oplus H_{L-1} \oplus \cdots , \tag{10}$$

assuming that $L \leqslant \dim(\mathcal{H})$. We can always achieve this condition by adding more qubits to the circuits.

Our aim is to approximately map a data point according to its "label subspace" $\{H_i\}$. Without loss of generality, let $\dim(H_i) = k$ and H_i be spanned by $\{|\psi_i^j\rangle_{j=1}^k\}$. Let a set of operators associated with label i, or equivalently, the subspace H_i, be $\{|\psi_i^j\rangle\langle\psi_i^j|\}_{j=1}^k$. The likelihood of given data $|x\rangle$ having some label i may be quantified by the projection of $|x\rangle$ onto the corresponding "label" subspace. Therefore $\mathcal{M}_i$ could have the following form of a projection operator:

$$\mathcal{M}_i = \sum_{j=1}^{k} |\psi_i^j\rangle\langle\psi_i^j| , \tag{11}$$

which essentially is the targeted ensemble density operator (up to a normalization) associated with label i. The same strategy is then followed as we minimize the cost in Eq. (4) and assign some unseen data x according to the value of f_i.

For example, we consider the binary supervised learning problem with $L = 2$ and 1-dim data set $X = X_A \cup X_B$, where

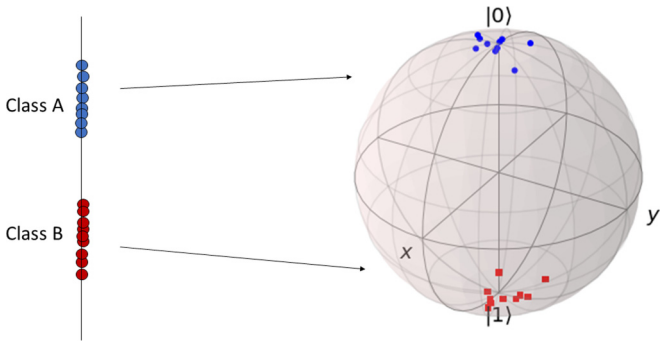

FIG. 4. Illustration of binary classification. After the training stage, data from A are mapped to the blue points, surrounding $|0\rangle$. Data from B are mapped to the red points, surrounding $|1\rangle$.

X_A and X_B are the training sets with label 0 and 1, respectively, as depicted in Fig. 4. For simplicity, we use one qubit in the embedding circuit (we refer readers to the one-qubit toy model in Ref. [40]). Hence dim $\mathcal{H} = 2$. We then make the decomposition

$$\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1,$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ are spanned by $|0\rangle$ and $|1\rangle$, respectively. We note that the classifying vector $\vec{f}$ has the form

$$\begin{bmatrix} \mathrm{Tr}(|x\rangle\langle x|\sigma_0) \\ \mathrm{Tr}(|x\rangle\langle x|\sigma_1) \end{bmatrix} = \begin{bmatrix} \mathrm{Tr}(|x\rangle\langle x|\cdot|0\rangle\langle 0|) \\ \mathrm{Tr}(|x\rangle\langle x|\cdot|1\rangle\langle 1|) \end{bmatrix}. \tag{12}$$

Following the same procedure as provided in Eq. (4) helps obtain the cost value

$$C = 1 - \tfrac{1}{2}\mathrm{Tr}[\sigma_z(\rho_A - \rho_B)], \tag{13}$$

where ρ_A and ρ_B are state ensembles of the two training sets A and B, respectively, and σ_z is the Pauli Z matrix. Minimization of this cost C with respect to the circuit's parameters will give an embedding $\Phi(x, \theta)$ that maps the data from A to the vicinity of $|0\rangle$ in $\mathcal{H}$ and the data from B to the vicinity of $|1\rangle$ in $\mathcal{H}$, as illustrated in Fig. 4.

An alternative picture also can be drawn from the described 1-dim data set. If we choose the label space to be $\{H_0, H_1\}$, then the classifying vector in Eq. (12) can be obtained by simply performing the measurement on the embedded state $|x\rangle$ in the z basis. Choosing a different label space, e.g., by decomposing $\mathcal{H} = \mathcal{H}_+ \oplus \mathcal{H}_-$, where $\mathcal{H}_+$ and $\mathcal{H}_-$ are spanned by $(|0\rangle + |1\rangle)/\sqrt{2}$ and $(|0\rangle - |1\rangle)/\sqrt{2}$, respectively, measurement in the x basis (i.e., the observable σ_x) would need to be used instead. The classifying vector $\vec{f}$ in this latter case is a direct result of such a σ_x measurement. Instead of the north and south poles on the Bloch sphere (Fig. 4), the quantum circuit training would then make data points cluster around the "$+\hat{x}$ pole" and "$-\hat{x}$ pole," where $\hat{x}$ is the unit vector point along the positive x direction.

2. Connection to traditional quantum classification models

By closely examining $\vec{f}$ in Eq. (12), the value of $f_0 = \mathrm{Tr}(|x\rangle\langle x|\cdot|0\rangle\langle 0|) = |\langle 0|\cdot|x\rangle|^2$ turns out to be the probability of obtaining state $|0\rangle$ when measuring the state $|x\rangle$ in the computational basis. Most current quantum classifiers [27,28] rely on these measurement outcomes after applying a

general circuit $\Phi(x,\theta) = W(\theta)U(x)$ to some initial state $|0\rangle$ for classification. Hence, by choosing the appropriate label space (specifically, the standard computational-basis state), there is an intrinsic connection between this explicit approach and other traditional models [27,29,38]. More precisely, traditional approaches can be unified by this explicit approach.

To be more specific about the intrinsic connection, we consider the structure of these traditional approaches, which has been shown to be closely related to the kernel method in the classical context [38]. The first part of the circuit maps classical data to some quantum states; training the parameter layer followed by measurement could be interpreted as learning the decision boundary for classification on the Hilbert space. Under the scope of our explicit approach, the whole circuit (everything before the measurement) is used for encoding purposes, and the optimization process would align the data in separated subspaces. As mentioned, if we choose the appropriate label space, then the measurement outcomes on the encoded state could be understood as the "distance" to the corresponding label space, and hence the classification can be done. Such intrinsic connection merges all these quantum classification models and hence provides a unified picture. In Appendix B, we provide a further example to illustrate the unification.

Such unification offers a twofold advantage. First, the evaluation of overlaps between the input data $|x\rangle$ and $|0\rangle$ or $|1\rangle$, as in Eq. (12), can be done simply by letting x undergo the embedding circuit $\Phi(x,\theta)$ once and performing measurements instead of invoking the embedding circuit twice (in the SWAP test subroutine). Additionally, the cost evaluation in Eq. (13) does not necessarily need to be done in an iterative manner. In Refs. [47,48], the authors provide elegant and efficient methods to encode the cost evaluation directly into quantum circuits. Hence the training time can be reduced.

III. NUMERICAL SIMULATIONS AND REAL-DEVICE EXPERIMENTS

With each approach, we train on the ideal simulator and then use the optimized circuit to test on the ideal simulator, the noisy simulator, and several real devices.

A. Implicit approach experiment

Data sets. For illustration purposes, we target the Iris data set [49,50] with $L = 3$ labels (Fig. 5). There are 50 data points in each class for a total of $N = 150$ data points. Ten data points are taken from each class to serve as the training data, and the remaining 40 are used for testing. Aside from classification, our aim is to demonstrate the formation of clusters in the featured Hilbert space $\mathcal{H}$.

Quantum embeddings. We use the same so-called *quantum approximate optimization algorithm (QAOA)-like ansatz* as in Ref. [40] (Fig. 6) for embedding. The unit circuit $\Phi(x,\theta)$ is composed of a feature layer $U(x)$ followed by the parameter layer $W(\theta)$. Hence we have $\Phi(x,\theta) = W(\theta)U(x)$, and the model is compact. A possible useful design of this embedding unit is to mix the feature parameters x and tunable parameters θ to reduce the depth while maintaining the efficiency (see Ref. [46]). For instance, instead of $R_y(x_1)$, one can consider $R_y(\theta_1, x_1)$ or, generally, $R_y(g(\theta_1, x_1))$, where g is some function.

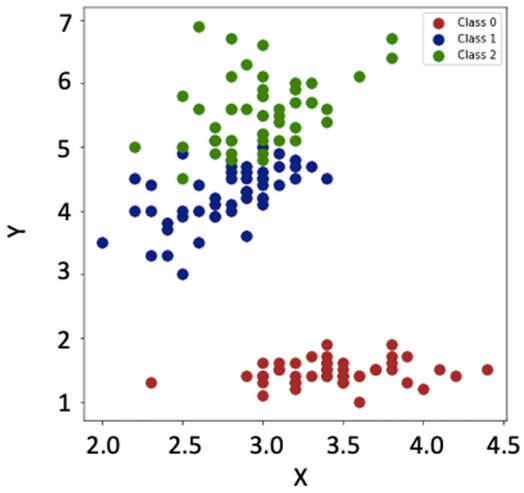

FIG. 5. The Iris data set. There are three classes distributed in a two-dimensional region. Different classes are represented by different colors. This data set is used in our implicit approach for classification.

Training stage. Because there are $L = 3$ labels in our problem, the cost function is

$$C = 1 - \frac{1}{3}\sum_{i=1}^{3} \mathrm{Tr}(\sigma_i)^2 + \frac{2}{3}\sum_{i<j} \mathrm{Tr}(\sigma_i\sigma_j). \tag{14}$$

The training procedure is as follows:

(i) Data from the training set are mapped to quantum states.

(ii) Define and use the SWAP test subprogram to evaluate $\mathrm{Tr}(\sigma_i^2)$ and $\mathrm{Tr}(\sigma_i\sigma_j)$. Later, we also use the inversion test.

(iii) Minimize the cost C in Eq. (14) over circuit parameters.

Our simulation uses the PENNYLANE software package [51], and the optimization of C over circuit parameters is done using the RMSPROP [52] optimizer with a learning rate of 0.01.

1. Results of noiseless simulations

Figure 7 shows the training curve. As minimization takes place, the embedded data in $\mathcal{H}$ are expected to form clusters, while those from different classes separate from each other. This is confirmed, as shown in Fig. 8, in the comparison of the overlap of embedded data before and after the training. In particular, the overlaps between the embedded data from

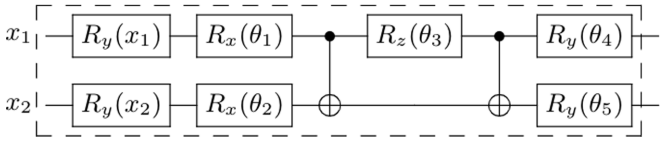

FIG. 6. Unit embedding circuit Φ. In our implementation, this unit is repeated four times. Hence the total number of trainable parameters is 20. In the end, the feature layer is repeated once more. The repetition of both feature and parameter layers has been used in Ref. [46] as a data reuploading strategy that yielded better classification ability.

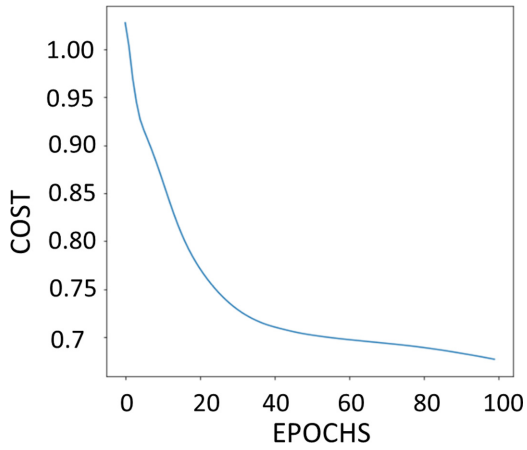

FIG. 7. Cost as a function of epochs. In this training, we use 100 epochs. There are 10 training points for each class, totaling 30 training points.

different classes become small after the training. This is especially the case for the overlap of class 0 with both class 1 and class 2. Thus we have verified the well separation of the embedded data from different classes.

After obtaining the optimized circuit parameters, we use the optimal circuit to perform a test on classifying the remaining unused data (i.e., the test data set). The overall accuracy obtained is 92.5%. Notably, only 30 data points (ten for each class) are used as the training set, which corresponds to 20% of the total data points. Such a result demonstrates that the classifier can classify unseen data with high accuracy, despite being trained with a relatively small training data set (more details in Sec. III C).

2. *Testing results from noisy simulations*

In real quantum hardware, noise and errors are important factors that reduce accuracy. To examine our method in the presence of noise, we test our classification with noisy models acquired from IBM Q "backends." The device noise model is generated from their device calibration and accounts for gate error probability, gate length, T_1 and T_2 (relaxation and dephasing times, respectively), and readout error probability. For convenience, Table I shows the average gate errors for the four backends considered in this paper.

We test the classification with the SWAP test circuit via the noisy simulations, and the results are tabulated in Table II. The accuracy seems to be unaffected by the noise, and the values from the noisy simulator using the four noisy models

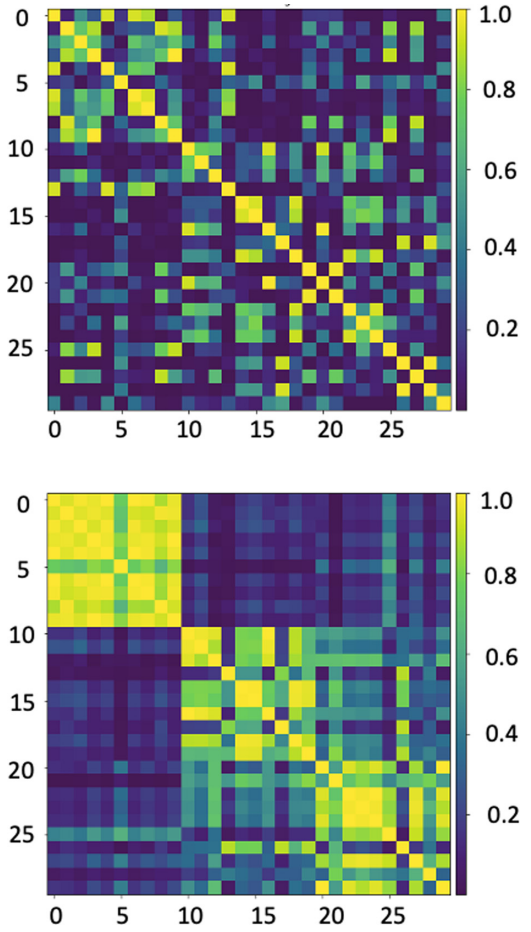

FIG. 8. Visualization of overlaps between training points (ten training points in each class). Top: Initial distribution of data in $\mathcal{H}$, in which the parameters in the variational quantum circuit are randomized. Bottom: After 100 training epochs, the data from the same class form a cluster in $\mathcal{H}$ as overlaps between their quantum states are high (brighter color). The visualization clearly shows that class 0 (red points) is more separated from the other two classes. Meanwhile, classes 1 (blue points) and 2 (green points) are less separated from each other. The observation is in agreement with the testing results because all testing points from class 0 are predicted with absolute accuracy, and false predictions only come from classes 1 and 2.

from the respective devices are 92.5, 90.83, 92.5, and 92.5%. The circuit parameters used are obtained from the noiseless optimization. The reason for not using noisy simulators to obtain the parameters is because we will perform the same testing on the actual hardware. Hence it would be impractical

TABLE I. Overall noise properties of four machines in our experiment. For each column (e.g., "$U1$ gate error"), we average over the values of all qubits in that device. The $U1$ gates have zero error because they are implemented with a frame change [53]. For "CNOT error," we average over all pairs of qubits. The last two columns show the results of noisy simulations. Circles, `make_circles` data set.

	$U1$ gate error	$U2$ gate error	$U3$ gate error	Readout error	CNOT error	Iris (%)	Circles (%)
`ibmq_16_melbourne`	0.0	0.00115	0.00229	0.06597	0.03157	92.5	97
`ibmq5_yorktown`	0.0	0.00084	0.00168	0.03494	0.02024	90.83	93
`ibmq_bogota`	0.0	0.00031	0.00062	0.03702	0.01171	92.5	96
`ibmq_rome`	0.0	0.00035	0.00071	0.02397	0.01344	92.5	95

TABLE II. Summary of testing results, including SWAP test and inversion test, on noisy simulators and real devices, for simulations and runs on actual backends for both the Iris and `make_circles` data sets. The result given in the top row for each backend corresponds to the noisy simulation (n.s.), and the results underneath correspond to the real device (r.d.). Values are percentages. ST, SWAP test; IT, inversion test.

	Iris	Circles
Ideal simulator	92.5	96
`ibmq_16_melbourne`	92.5(n.s.) 32.5(r.d. ST) 88.33(r.d. IT)	97(n.s.) 99(r.d.)
`ibmq_5_yorktown`	90.83(n.s.) 50(r.d. ST) 83.83(r.d. IT)	93(n.s.) 91(r.d.)
`ibmq_bogota`	92.5(n.s.) 75(r.d. ST) 91.67(r.d. IT)	96(n.s.) 95(r.d.)
`ibmq_rome`	92.5(n.s.) 28.33(r.d. ST) 91.67(r.d. IT)	95(n.s.) 94(r.d.)

and too time-consuming to perform the training directly on the hardware as the jobs queue could be long and execution of the training circuits would have to be split over many jobs.

3. Runs on quantum computers

With the noisy simulation, we also test our ideally trained model on real quantum backends. Table II summarizes the detailed results, including simulations and real devices. For the same classification, we run two different methods to obtain overlaps: the SWAP test, which uses five qubits, and the inversion test, which uses only two qubits.

The accuracy with the SWAP test ranges from 28 to 75% on various backends. However, the inversion test accuracy remains stable around 90%. This clearly shows substantial performance differences between the SWAP and inversion tests. Likely, the main factor that accounts for such a discrepancy is the controlled-SWAP (CSWAP) gate (Fig. 16) in the SWAP test circuit. The number of CSWAP gates required for two n-qubit states scales as $\mathcal{O}(n)$. Each CSWAP gate then is decomposed into many CNOT gates (which are noisy) as shown in Fig. 9. Despite the noisy simulations yielding accuracy around 90%, runs on the actual machines suffer accumulated errors not captured in the noise model used in the simulations.

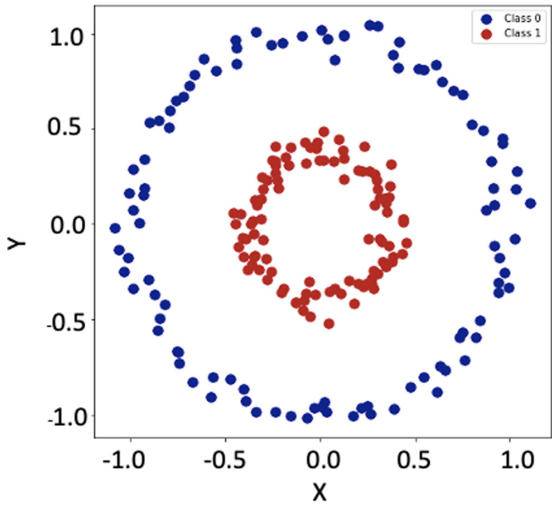

FIG. 10. The `make_circles` data set with $N = 2$ labels used in our explicit approach for classification.

On the other hand, the inversion test does not need the CSWAP gate and, hence, requires fewer CNOTs—but at the cost of doubling the quantum circuit depth. In our classification model, there is a trade-off between using the SWAP test and using the inversion test. As our small-size experiments have shown, the inversion test should be used for better classification on NISQ machines. However, it requires the ability to invert the embedding circuit and can only evaluate the overlaps between two data points (of pure states). Hence classifying unseen data (by obtaining the classifying vector $\vec{f}$) must be done in an iterative manner. Conversely, the SWAP test can handle mixed states, and the classification can be sped up with QRAM. In addition, the SWAP test performance can be further improved by using methods introduced in Ref. [54]. Such an approach is hardware dependent (as the authors examined on IBM Q and Rigetti separately), and it requires fewer CNOT gates [55].

B. Explicit approach experiment

Data sets. To illustrate this approach, we target the data set `make_circles` with $L = 2$ labels (Fig. 10). Fifteen points are taken from each class to serve as the training data. For testing, we generate an additional 50 points for each class.

Training stage. We choose two subspaces spanned by $|00\rangle$ and $|11\rangle$, respectively, as label spaces and train the circuit to map data from class 0 (blue points in Fig. 10) to H_{00} and from

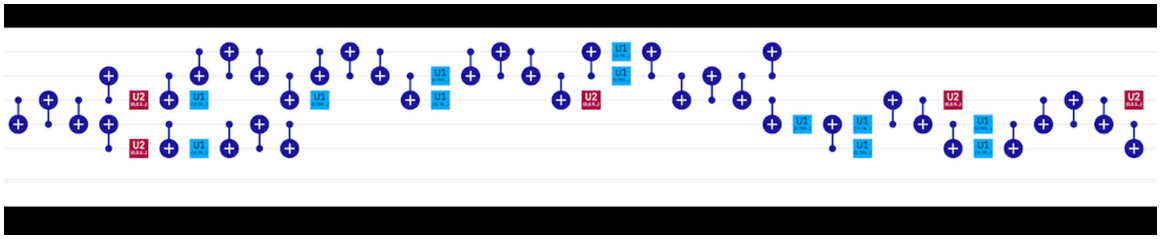

FIG. 9. Decomposition of controlled-SWAP gates into many CNOT and one-qubit gates. In this paper, classical data are embedded in two-qubit states. Hence there are five total qubits in the SWAP test circuit that uses a controlled-SWAP gate. Note that this diagram only shows the decomposition of CSWAP, not including the data embedding part. There are 38 CNOT gates used in the decomposition.

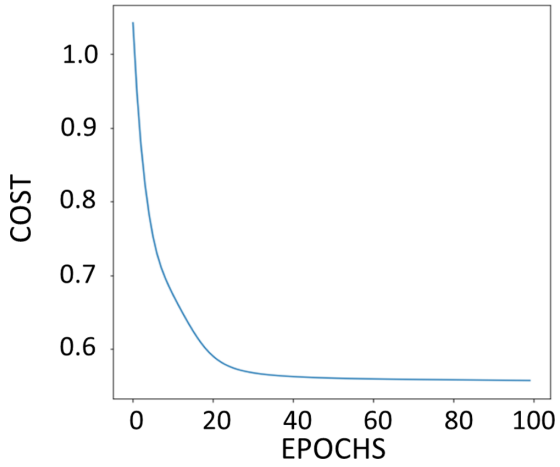

FIG. 11. Cost as a function of epochs. There are 100 epochs in this training with 15 training points for each class and 30 total training points.

class 1 (red points in Fig. 10) to H_{11}. Given some input data x, the classifying vector is then

$$\begin{bmatrix} \mathrm{Tr}(|x\rangle\langle x| \cdot |00\rangle\langle 00|) \\ \mathrm{Tr}(|x\rangle\langle x| \cdot |11\rangle\langle 11|) \end{bmatrix}. \tag{15}$$

The cost function becomes

$$C = 1 - \frac{1}{2}\{\mathrm{Tr}[\sigma_A(|00\rangle\langle 00| - |11\rangle\langle 11|)] - \mathrm{Tr}[\sigma_B(|00\rangle\langle 00| - |11\rangle\langle 11|)]\}, \tag{16}$$

where $\sigma_A = \frac{1}{N_A}\sum_A |x_A\rangle\langle x_A|$ and $\sigma_B = \frac{1}{N_B}\sum_B |x_B\rangle\langle x_B|$. A and B refer to class 0 and class 1, respectively.

1. Results of noiseless simulations

Figure 11 presents the training curve in the noiseless simulation. The overall accuracy is 96%. As in the implicit case, we visualize the overlaps between training points before and after training with 100 epochs (Fig. 12). This confirms the well separation of embedded data from different classes in our explicit approach.

2. Noisy simulations

As in the previous implicit case, we also use the trained parameters from the noiseless simulation for the quantum embedding but perform the noisy simulation to classify the testing data set. The accuracy of these noisy simulations is tabulated in Table II. Noise details can be found in Table I.

3. Runs on quantum computers

We perform the classification experiments of this explicit approach on the real quantum backends `ibmq_16_melbourne`, `ibmq_5_yorktown`, `ibmq_bogota`, and `ibmq_rome` and compare their accuracy with the noisy simulation with the noise model from the same backend. Table II summarizes the results.

The accuracies from the noisy simulators and real machines turn out to agree well with each other, achieving values above 90%. This indicates that the explicit approach is less affected by the noise compared with the implicit approach using the SWAP test. This is reasonable as the explicit approach requires fewer resources, such as fewer two-qubit gates, than the implicit approach with the SWAP test. We simply let the input data run through the circuit and perform computational-basis measurements as the classification only depends on the probability of obtaining $|00\rangle$ and $|11\rangle$.

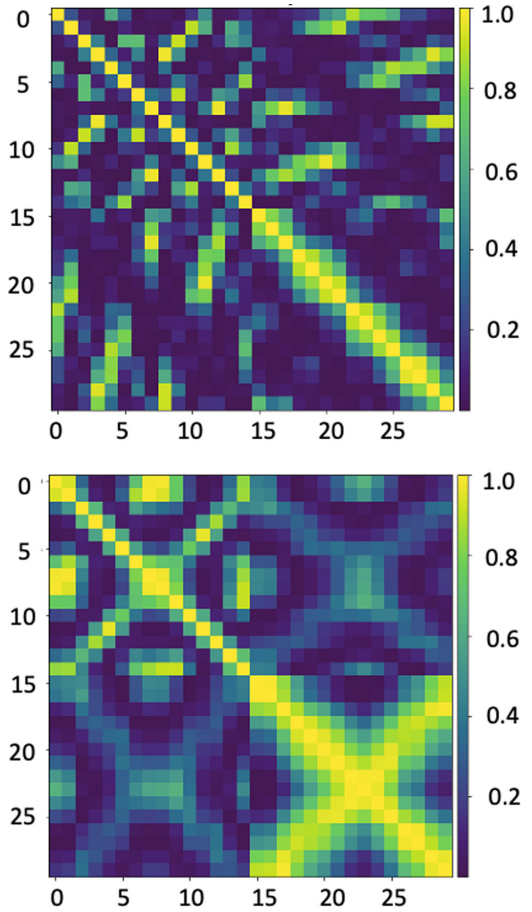

FIG. 12. Visualization of overlaps between training points (15 training points in each class). Top: Initial overlaps between data points (with randomized circuit parameters). Bottom: After training process, data from different classes become separated. Compared with the implicit approach, the "well separation" is less apparent. In other words, clusters are less tight. This may be reasonably explained by the way the two methods work. In the implicit approach, the optimization procedure focuses on *directly* separating data points from different classes. Meanwhile in the explicit approach, data get separated *indirectly*, i.e., through the predefined subspaces. Hence, in theory, the implicit approach has certain advantages over the explicit method in terms of attaining complete separation. In practice, both approaches have strong classification ability, demonstrated in our experiments.

C. Training over small samples

We observe that both approaches produce surprisingly good testing accuracy despite training with small data points. This provides motivation to determine whether one approach is more robust than the other in terms of learning capacity. Another motivation is to investigate how testing accuracy varies

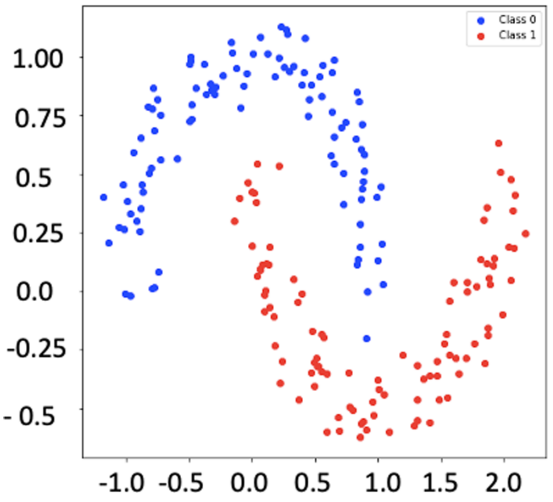

FIG. 13. The `make_moons` data set.

with training size. We choose the `make_moons` data set with $L = 2$ labels to perform the numerical experiment (Fig. 13).

The procedure is given as follows.

(i) For each class, we generate a fixed set of 50 points (hence there are 100 points in total), serving as testing instances.

(ii) For each class, we then choose randomly 5, 10, 20, and 25 points, serving as training instances. For each number of training instances, we average over multiple training sessions to obtain the testing accuracy.

(iii) We train the quantum circuit using implicit and explicit approaches separately and compare the testing accuracy after 100 epochs of training.

Table III summarizes the results. From these, we observe that both approaches perform reasonably well even with a small training size. It is interesting to note that even with five data points in each class for training, the implicit approach has about 10% higher accuracy in the testing than the explicit approach. However, this gap becomes smaller as the training size increases.

TABLE III. Summary of testing results from the `make_moons` data set on the simulator, with varying training size and fixed testing points. There are 100 points in each class for a total of 200. For each class, we choose 50 points to serve as testing instances. The remaining 50 are used as the pool to randomly select a certain size of training instances as described previously.

Training size	Implicit approach (%)	Explicit approach (%)
5	84	75.8
7	84	80.4
10	83.4	80.4
15	87	86
20	87.5	86.5
25	89.5	92

IV. PROSPECTS IN THE NISQ ERA

To maximally enhance the performance of any quantum algorithm or, generally, a quantum procedure, we also need to take into account the hardware structure, e.g., the connectivity of qubits in the system and specific qubits chosen. Figure 14 shows the topology of the machine `ibmq_bogota` used in this paper.

Table II shows the result of testing the Iris data set on `ibmq_bogota` with the inversion test done using qubits labeled "3" and "4." We also carried out the same "testification" using qubits "1" and "2." The testification result 65.83% is dramatically lower than that using qubits 3 and 4 (91.67%). Such a deviation can be reasonably argued from the noise rates of the qubit pair involved and their CNOT gates. As depicted in Fig. 14, qubits 3 and 4, as well as the connection between them, have much lower error rates compared with those of qubits 1 and 2. Hence, in practice, any quantum procedure needs to be hardware-aware to reach its maximum efficiency. Of course, our experiment requires very small numbers of qubits and simple gates. As such, we can simply choose specific qubits to obtain better accuracy. More complicated quantum circuits generally require more careful qubit specification. The quantum hardware topology also varies, e.g., `ibmq_16_melbourne` and `ibmq5_yorktown` versus `ibmq_bogota` (Fig. 15). Such differences can affect the decomposition of a multiqubit gate into available one- and two-qubit (in particular) gates. A hardware-aware compiler that optimizes the selection also is a necessity for future large-scale tasks. This hardware-specification optimization is important practically and requires additional development. Given an arbitrary quantum backend's topology and description of some quantum procedures, such as a circuit's length, width, and number of one-qubit or two-qubit gates, it may be worth determining whether a systematic procedure exists that can decide which qubits—and in what orders—should be used in order to maximize the performance.

Other works also have demonstrated the ability and feasibility of using actual quantum computers to classify real-world data [28,56]. In addition to providing a unified framework for QSL, we have performed simulations and cloud-based real-device experiments. These experiments on real quantum backends have extended the prospects of applying quantum computers for ML one step further, demonstrated explicitly in this paper that current noisy quantum computers can achieve high accuracy on classifying data. The low-accuracy results obtained via the SWAP test routine may be improved by using the inversion test routine, and we have emphasized that the inversion test is more appropriate in the NISQ era. Real quantum systems undoubtedly are more complicated, and their noise on long circuits (especially those with many CNOTs) may result in worse accuracy than noisy simulations. In our experiments, the error rate of one-qubit gates is $\sim 10^{-3}$ or smaller, and that of the two-qubit CNOT is $\sim 10^{-2}$ for current hardware [37]. Full implementation of quantum error correction remains elusive. Along with development of precise, high-fidelity gates, efforts have been made in error mitigation methods [57–60] to obtain useful outcomes. Some of these mitigation methods require repetition of the same circuit but with different overall error rates by

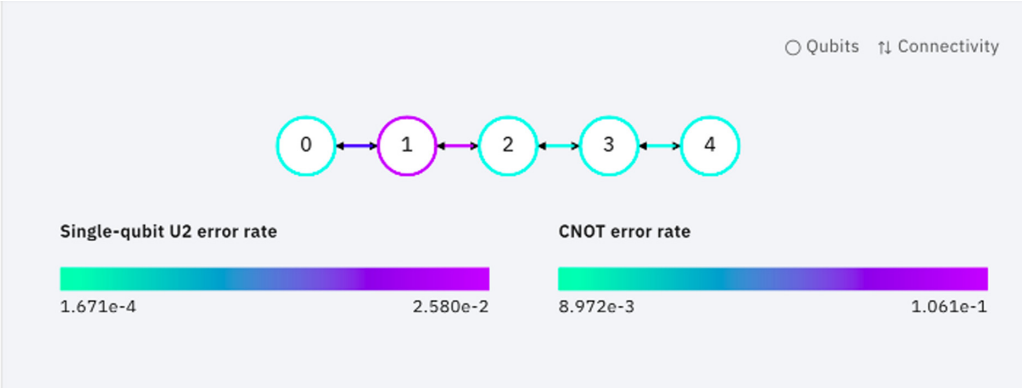

FIG. 14. Topology of `ibmq_bogota`. The picture was acquired at the time of our experiment. Topologies of other machines used in our experiment can be found in Fig. 15.

FIG. 15. Topology and the coupling map of other IBM Q devices used in this paper: `ibmq_5_yorktown`, `ibmq_16_melbourne`, and `ibmq_rome`.

possibly stretching the gate pulses. This allows observables to be extrapolated to the gate noiseless limit [57,58]. Error mitigation in measurement also is necessary to infer correct readout outcomes [59,60]. The experiments done as part of our work do not employ any mitigation techniques. As such, our real-device results may be further improved with these techniques, especially results from the SWAP test using gate mitigation. Additionally, our classification model has been shown to do well with a small training pool (compared with the testing set), achieving very high accuracy. Hence one can reasonably expect that the model can be trained on real machines to achieve comparable performance.

V. CONCLUSION

In our framework for quantum supervised learning, the main conceptual tool of our method is the idea that the input data x are "forwarded" to the classifying vector $\vec{f}$, and their classification can be done accordingly. A hybrid optimization step then proceeds to train the circuit. After being trained, the embedding circuit can map the data from the input space $\mathcal{X}$ to the proper subspaces in $\mathcal{H}$.

Our work emphasizes that the quantum feature map, equipped with a learning procedure, is an especially powerful tool for supervised learning. With the implicit approach, the number of separated classes (labels) in a supervised learning problem ideally can be arbitrarily high. Thus it provides a means to construct a universal quantum classifier. Compared with the explicit approach, the learning capacity of this approach has been demonstrated with a small training pool, which is also encouraging. Moreover, we show that the explicit approach can intrinsically unify other *traditional* QSL models (detailed in Appendix B). The fact that our framework can be divided into explicit and implicit approaches demonstrates its flexibility, affording the option to choose how data are embedded and analyzed on the Hilbert space $\mathcal{H}$. These two approaches constitute a unified framework for supervised learning methods using a quantum computer.

Along with classification, we note that the trained quantum circuit possibly can be employed as a subroutine of other quantum ML algorithms as we know with high confidence that embedded data from different classes are well separated

from each other (trained with the implicit approach) or approximately well contained in some subspaces in $\mathcal{H}$ (trained with the explicit approach).

ACKNOWLEDGMENTS

We acknowledge use of the IBM Q for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Q team. This research used resources of the Oak Ridge Leadership Computing Facility, which is a U.S. Department of Energy, Office of Science (DOE-SC), User Facility supported under Contract No. DE-AC05-00OR22725. This work was partially supported by the DOE-SC Office of High Energy Physics program under Award No. DE-SC-0012704 (S.Y.-C.C.), Brookhaven National Laboratory LDRD No. 20-024 (S.Y.-C.C.), and Subcontract No. 384153 from Brookhaven Science Associates LLC (T.-C.W.).

APPENDIX A: SWAP AND INVERSION TESTS

Figure 16 provides a description of two alternative methods to evaluate the overlaps between two data points. The first is the controlled-SWAP gate, acting on five qubits, and the second is the inversion test.

APPENDIX B: MORE ON THE EXPLICIT APPROACH

In Sec. II C 2, we have illustrated the intrinsic relation between the explicit approach and traditional quantum classification models via the measurement outcomes. Here, we offer an alternative explanation. We consider a binary classification problem (two classes, A and B) and a single-qubit quantum circuit (Fig. 17).

In traditional approaches, the first block $U(x)$ embeds classical information x into a quantum state in the Hilbert space

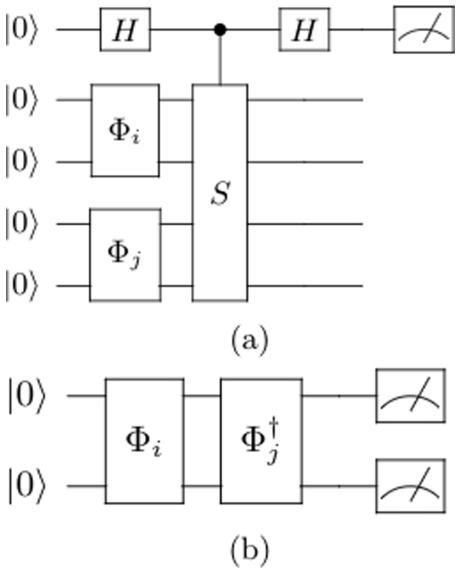

FIG. 16. Circuit representation for the (a) SWAP test and (b) inversion test. There is an abuse of the notation Φ: In both the SWAP and inversion test circuits, the actual embedding circuit $\Phi_{i,j}$ already includes the repetition of the unit embedding in Fig. 6.

FIG. 17. A common circuit model for machine learning tasks.

$\mathcal{H}$. Then, the variational layer $W(\theta)$ is trained to distinguish those embedded states. For instance, x can be assigned to class A if the probability of measuring 0 is greater than the probability of measuring 1 ($P_0 > P_1$). The training step should focus on maximizing the probability of measuring 0 for data in class A and measuring 1 otherwise.

Recall that our embedding-based framework exploits the ability of a quantum circuit to represent data in a complex space. An alternative, simple perspective is evident if we interpret the whole circuit as an encoding of the classical data x.

Without loss of generality, we assume that classical data x are mapped to a quantum state $|\psi\rangle$ (refer to Fig. 18). When we measure this state, the probability of measuring 0 is $\cos^2(\theta)$, which means that if such a probability is high, the state $|\psi\rangle$ will be close to the "north pole" $|0\rangle$. If we follow the explicit approach and decompose $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$, where $\mathcal{H}_0, \mathcal{H}_1$ are spanned by $|0\rangle, |1\rangle$, respectively, we end up maximizing the probability of measuring 0 for data from class A and measuring 1 for data from class B. Pictorially, those data from class A will form a cluster around $|0\rangle$, and data from class B will cluster around $|1\rangle$. After optimization, these clusters are hopefully well separated. To classify unseen data, we can use the optimized circuit to map them to some states and measure. The measurement probability may be understood as a "closeness" to either one of the two data cluster centers from classes A and B. This example also illustrates that our embedding-based framework, or, more specifically, the explicit approach, conceptually unifies other *traditional* quantum classification models, as they share the same underlying mathematical structure. To relate the explicit approach to the kernel method, one can think that the above decomposition $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$ sets a fixed decision boundary (the dashed line in Fig. 18), and training of the embedding circuit would maximally distribute the data points on two sides, far from the boundary.

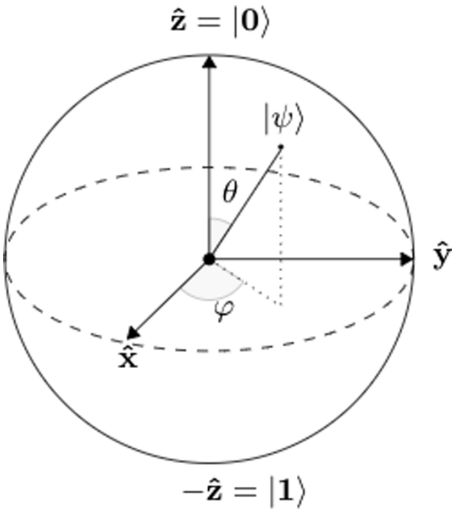

FIG. 18. Visualization of a qubit on a Bloch sphere. Note that the angle θ in this figure differs from the circuit parameters θ in Fig. 17.

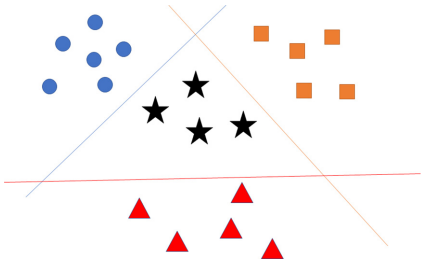

FIG. 19. One-vs-all strategy. There are three classes (blue, orange, and red) and the corresponding decision boundary (blue, orange, and red lines). All classes are linearly separable for simplicity.

APPENDIX C: ONE-VERSUS-ALL STRATEGY

The one-versus-all strategy (Fig. 19) often has been used to transform a binary classifier to a multiclass classifier, especially for those models that, in essence, can deal with only binary classification. Here, we review this strategy and discuss its drawbacks.

The underlying mechanism of the one-versus-all strategy is that it assumes there are only two classes (or labels) in the supervised learning problem that learn the corresponding decision boundary. For example, in Fig. 19, all blue circles and red triangles can be treated as one class that learns the decision boundary to distinguish them from the orange rectangles, as well as the decision boundary between the blue circles and the other symbols and the decision boundary between the red triangles and the other symbols.

The caveat of the one-versus-all strategy is clear from Fig. 19: The black stars (unseen data) struggle to find a class (label). A similar issue appears in QSL, where the data are embedded by a fixed circuit in the Hilbert space $\mathcal{H}$ and the subsequent variational circuit is trained to draw the decision boundary. Notably, our framework can naturally surpass this issue as the representation of the data in $\mathcal{H}$ is learned. Then, a measure is employed to compare data directly as in the implicit approach or indirectly as in the explicit approach. Hence a label for unseen data is always guaranteed.

[1] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2002).

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

[3] 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, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[4] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, Measurement-based quantum computation, Nat. Phys. **5**, 19 (2009).

[5] R. Raussendorf and T.-C. Wei, Quantum computation by local measurement, Annu. Rev. Condens. Matter Phys. **3**, 239 (2012).

[6] P. W. Shor, Polynomial-time algorithms for prime factorization, and discrete logarithms on a quantum computer, SIAM Rev. **41**, 303 (1999).

[7] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing* (Association for Computing Machinery, New York, 1996), pp. 212–219.

[8] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*, Conference Track Proceedings (2015).

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision, and Pattern Recognition* (IEEE, New York, 2015), pp. 1–9.

[10] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, Deep learning for computer vision: A brief review, Comput. Intell. Neurosci. **2018**, 7068349 (2018).

[11] I. Sutskever, O. Vinyals, and Q. V. Le, Sequence to sequence learning with neural networks, in *Advances in Neural Information Processing Systems (NIPS 2014)* (Curran Associates, Red Hook, NY, 2014), Vol. 27, pp. 3104–3112.

[12] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, and S. Zhao, Applications of machine learning in drug discovery, and development, Nat. Rev. Drug Discovery **18**, 463 (2019).

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

[14] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic, New York, 2014).

[15] M. Schuld, *Machine Learning in Quantum Spaces* (Springer, Cham, Switzerland, 2019).

[16] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress, Rep. Prog. Phys. **81**, 074001 (2018).

[17] E. Aïmeur, G. Brassard, and S. Gambs, Quantum speed-up for unsupervised learning, Mach. Learn. **90**, 261 (2013).

[18] J. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, C. A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete *et al.*, Unsupervised machine learning on a hybrid quantum computer, arXiv:1712.05771.

[19] N. Wiebe, A. Kapoor, and K. Svore, Quantum algorithms for nearest-neighbor methods for supervised, and unsupervised learning, arXiv:1401.2142.

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

[21] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, q-means: A quantum algorithm for unsupervised machine learning, in *Advances in Neural Information Processing Systems (NIPS 2019)* (Curran Associates, Red Hook, NY, 2019), Vol. 32, pp. 4134–4144.

[22] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers* (Springer, New York, 2018).

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

[24] G. Sergioli, R. Giuntini, and H. Freytes, A new quantum approach to binary classification, PloS One **14**, e0216224 (2019).

[25] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum Support Vector Machine for Big Data Classification, Phys. Rev. Lett. **113**, 130503 (2014).

[26] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, arXiv:1802.06002.

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

[28] 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).

[29] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[30] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems (NIPS 2012)* (Curran Associates, Red Hook, NY, 2012), Vol. 25, pp. 1097–1105.

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

[33] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, Hybrid quantum-classical convolutional neural networks, arXiv:1911.02998.

[34] S. Lu, L.-M. Duan, and D.-L. Deng, Quantum adversarial machine learning, Phys. Rev. Research **2**, 033212 (2020).

[35] Y. Du, M.-H. Hsieh, T. Liu, S. You, and D. Tao, On the learnability of quantum neural networks, arXiv:2007.12369.

[36] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu *et al.*, Experimental quantum generative adversarial networks for image generation, arXiv:2010.06201.

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

[38] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[39] S. Chopra, R. Hadsell, and Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in *2005 IEEE Computer Society Conference on Computer Vision, and Pattern Recognition (CVPR'05)* (IEEE, New York, 2005), Vol. 1, pp. 539–546.

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

[41] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, Quantum Inf. Process. **13**, 2567 (2014).

[42] P. Rebentrost, T. R. Bromley, C. Weedbrook, and S. Lloyd, Quantum Hopfield neural network, Phys. Rev. A **98**, 042308 (2018).

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

[44] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, Training deep quantum neural networks, Nat. Commun. **11**, 808 (2020).

[45] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, Phys. Rev. Lett. **100**, 160501 (2008).

[46] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, Data re-uploading for a universal quantum classifier, Quantum **4**, 226 (2020).

[47] S. Adhikary, An entanglement enhanced training algorithm for supervised quantum classifiers, arXiv:2006.13302.

[48] S. Cao, L. Wossnig, B. Vlastakis, P. Leek, and E. Grant, Cost-function embedding, and dataset encoding for machine learning with parametrized quantum circuits, Phys. Rev. A **101**, 052309 (2020).

[49] R. A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. **7**, 179 (1936).

[50] E. Anderson, The species problem in iris, Ann. Mo. Bot. Gard. **23**, 457 (1936).

[51] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, C. Blank, K. McKiernan, and N. Killoran, PennyLane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968.

[52] T. Tieleman and G. Hinton, Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude, Coursera: Neural Networks for Machine Learning **4**, 26 (2012).

[53] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient Z gates for quantum computing, Phys. Rev. A **96**, 022330 (2017).

[54] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, Learning the quantum algorithm for state overlap, New J. Phys. **20**, 113022 (2018).

[55] A similar discussion regarding the inversion test and the method in Ref. [54] also is presented in Ref. [28]. Our experimental work elaborates on this issue further as we have explicitly examined the SWAP and inversion tests' performance in the presence of noise.

[56] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, Quantum classifier with tailored quantum kernel, npj Quantum Inf. **6**, 41 (2020).

[57] K. Temme, S. Bravyi, and J. M. Gambetta, Error Mitigation for Short-Depth Quantum Circuits, Phys. Rev. Lett. **119**, 180509 (2017).

[58] S. Endo, S. C. Benjamin, and Y. Li, Practical Quantum Error Mitigation for Near-Future Applications, Phys. Rev. X **8**, 031027 (2018).

[59] Y. Chen, M. Farahzad, S. Yoo, and T.-C. Wei, Detector tomography on IBM quantum computers and mitigation of an imperfect measurement, Phys. Rev. A **100**, 052315 (2019).

[60] S. Bravyi, S. Sheldon, A. Kandala, D. C. Mckay, and J. M. Gambetta, Mitigating measurement errors in multiqubit experiments, Phys. Rev. A **103**, 042605 (2021).