Approximate complex amplitude encoding algorithm and its application to classification problem in financial operations

Naoki Mitsuda,1,2 Kohei Nakaji,3,4,2 Yohichi Suzuki,2 Tomoki Tanaka,5,2 Rudy Raymond,6,2,7 Hiroyuki Tezuka,8,2,9 Tamiya Onodera,6,2 and Naoki Yamamoto2,10
1Sumitomo Mitsui Trust Bank, Ltd., 1-4-1, Marunouchi, Chiyoda-ku, Tokyo, 100-8233, Japan
2Quantum Computing Center, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, 223-8522, Japan
3Department of Chemistry, University of Toronto, Toronto, Ontario M5G 1Z8, Canada
4Research Center for Emerging Computing Technologies, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan.
5Mitsubishi UFJ Financial Group Inc. and MUFG Bank Ltd., 2-7-1 Marunouchi, Chiyoda-ku, Tokyo 100-8388
6IBM Quantum, IBM Japan, 19-21 Nihonbashi Hakozaki-cho, Chuo-ku, Tokyo, 103-8510, Japan
7Dept. of Computer Science, the Univ. of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo, Japan 113-0033
8Sony Group Corporation, 1-7-1 Konan, Minato-ku, Tokyo, 108-0075, Japan
9Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, 223-8522, Japan
10Department of Applied Physics and Physico-Informatics, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522, Japan.

Quantum computing has a potential to accelerate the data processing efficiency, especially in machine learning, by exploiting special features such as the quantum interference. The major challenge in this application is that, in general, the task of loading a classical data vector into a quantum state requires an exponential number of quantum gates. The approximate amplitude encoding (AAE) method, which uses a variational means to approximately load a given real-valued data vector into the amplitude of quantum state, was recently proposed as a general approach to this problem mainly for near-term devices. However, AAE cannot load a complex-valued data vector, which narrows its application range. In this work, we extend AAE so that it can handle a complex-valued data vector. The key idea is to employ the fidelity distance as the cost function for optimizing a parameterized quantum circuit, where the classical shadow technique is used to efficiently estimate the fidelity. We apply this algorithm to realize the complex-valued-kernel binary classifier called the compact Hadamard classifier, and then give a numerical experiment showing that it enables classification of Iris dataset and credit card fraud detection.

I. INTRODUCTION

Quantum computing is expected to execute information processing tasks that classical computers cannot perform efficiently. One of the most promising domains in which quantum computing has the potential to boost its performance is machine learning [1–6]. The advantage of quantum computing in machine learning is the capability to represent and manipulate exponential amount of classical data since quantum computers can exploit an exponentially large Hilbert space and quantum interference [7–9]. Furthermore, quantum computing can exponentially speed up basic linear algebra subroutines [10, 11], which are at the core of machine learning, such as Support Vector Machine (SVM) [12], the solution of systems of linear equations [11] and principal component analysis [13]. Nevertheless, applying quantum machine learning to practical problems is still hard due to constraints on the size and depth of quantum circuits, especially for noisy intermediate-scale quantum computers [14].

In recent years, some attempts to implement machine learning algorithms with small size and shallow quantum circuits have been made, one of which is a kernel-based classifier such as quantum SVM [15], swap-test classifier [16], Hadamard test classifier (HTC) [17], and compact Hadamard classifier (CHC) [18]. The central idea underlying these classifiers is that inner products in an exponentially-large Hilbert space [6] can be directly accessed by measurement without expensive subroutines [12, 16, 17, 19–21]. It should be noted, however, that these classifiers assume that the classical data (i.e., the training data and the test data) has been loaded into the amplitudes of a quantum state, i.e., \textit{amplitude encoding} [17].

In general, the number of gates grows exponentially with the number of qubits for realizing the amplitude encoding [22–28], which might be a major bottleneck of the practical application of quantum computation. In order to address this issue, Ref. [29] proposed an algorithm called the \textit{approximate amplitude encoding (AAE)} that generates approximated n-qubit quantum states with using only $O(\text{poly}(n))$ gates. However, AAE is only applicable to loading real-valued data and cannot load complex-valued data. This limitation narrows the scope of AAE application. For example, AAE cannot be applied for preparing an initial state of CHC [18], because CHC utilizes not only real part but also imaginary part of the probability amplitudes of the quantum state. Aside
from CHC, complex-valued data encoding is also required for state preparation of wavepacket dynamics simulations [30, 31] in quantum chemistry. Since the wavepackets are described with complex functions, complex-valued amplitude encoding is demanded to prepare arbitrary initial states. Hence, improvements to enable complex-valued data encoding with AAE are required.

In this paper, we extend the applicability of AAE and propose a method that can load complex-valued data with shallow quantum circuits. We refer to this algorithm as approximate complex amplitude encoding (ACAE). The key idea is to change the cost function from the maximum mean discrepancy (MMD) [32, 33] used in AAE, to the fidelity, which can capture the difference in complex amplitude between two quantum states unlike the MMD-based cost function. A notable point is that the classical shadow [34] is used to efficiently estimate the fidelity or its gradient. As a result, ACAE allows embedding the real and imaginary parts of any complex-valued data into the probability amplitudes of the quantum state. In addition, we provide an algorithm composed of ACAE and CHC; this algorithm realizes a quantum circuit for binary data classification, using fewer gates than the original CHC which requires exponentially many gates to prepare the exact quantum state. We then give a proof-of-principle demonstration of this classification algorithm for the benchmark Iris data classification problem; further, we apply the algorithm to the credit card fraud detection problem, that is considered as a key challenge in financial institutions.

The rest of the paper is organized as follows. In Section II, we describe ACAE. Section III presents the ACAE’s application to CHC. Section IV gives a demonstration of HCH implemented with ACAE, to carry out the Iris data application to CHC. Section V gives a demonstration of this classification algorithm for the benchmark Iris data classification problem; further, we apply the algorithm to the credit card fraud detection problem, that is considered as a key challenge in financial institutions.

The rest of the paper is organized as follows. In Section II, we describe ACAE. Section III presents the ACAE’s application to CHC. Section IV gives a demonstration of HCH implemented with ACAE, to carry out the Iris data classification problem and the credit card fraud detection problem. We conclude the paper with some remarks in Section V.

II. APPROXIMATE COMPLEX AMPLITUDE ENCODING ALGORITHM

A. Goal of the approximate complex amplitude encoding algorithm (ACAE)

Quantum state preparation is an important subroutine in quantum algorithms that process a classical data, such as linear solvers [10, 11], quantum machine learning [35], and quantum recommendation systems [36]. In many cases, these algorithms assume the ability to access to a state preparation oracle $U$ that encodes a $N$-dimensional complex vector $c = \{c_0, \ldots, c_{N-1}\}, c_k \in \mathbb{C}$ to the amplitudes of an $n$-qubit state $|\text{Data}\rangle$:

$$|\text{Data}\rangle = U|0\rangle^\otimes n = \sum_{k=0}^{N-1} c_k |k\rangle,$$  

(1)

where the input vector is normalized as $\|e\| = 1$. Note that $n = \lceil \log(N) \rceil$. Hereafter, the state (1) is referred to as the target state. Recall that, in general, a quantum circuit for generating the target state requires $O(2^n)$ controlled-NOT (CNOT) gates, which might destroy any possible quantum advantage [22–28].

The objective of ACAE is to generate a quantum state that approximates the target state (1), using a parameterized quantum circuit (PQC) that is represented by a unitary matrix $U(\theta)$ with $\theta$ the vector of parameters. Hereafter, we refer to the state generated from the PQC as a model state, i.e., $|\psi(\theta)\rangle = U(\theta)|0\rangle^\otimes n$. We train the model state $|\psi(\theta)\rangle$ to approximate the target state except for the global phase; hence, ideally, the model state is trained to satisfy

$$|\psi(\theta)\rangle = U(\theta)|0\rangle^\otimes n = e^{i\alpha}|\text{Data}\rangle,$$  

(2)

where $e^{i\alpha}$ is the global phase.

Note that, if the elements of $c = \{c_0, \ldots, c_{N-1}\}$ are all real-numbers, we can use AAE [29], which also trains a PQC to generate an approximating state. The training is accomplished by minimizing the cost function given by the MMD [32, 33] between two probability distributions corresponding to the target and the model states.

B. The proposed algorithm

1. Cost function

In order to execute a complex-valued data loading, it is necessary to introduce a measure that reflects the difference between two quantum states with complex-valued amplitude, which cannot be captured by the MMD-based cost function. Here we employ the fidelity between the model state $|\psi(\theta)\rangle$ and a target state $|\text{Data}\rangle$:

$$f(\theta) = \text{Tr} (\rho_{\text{model}}(\theta) \rho_{\text{target}}) = |\langle \text{Data}|\psi(\theta)\rangle|^2,$$  

(3)

where $\rho_{\text{model}}(\theta) = |\psi(\theta)\rangle\langle \psi(\theta)|$ and $\rho_{\text{target}} = |\text{Data}\rangle\langle \text{Data}|$. Although in general the fidelity can be estimated using the quantum state tomography [37], it is highly resource-intensive because this procedure requires accurate expectation values for a set of observables whose size grows exponentially with respect to the number of qubits. For this reason, we employ the classical shadow technique to estimate the fidelity.

2. Fidelity estimation by classical shadow

Classical shadow [34] is a method for constructing a classical description approximating a quantum state us-
ing much fewer measurements than the case of state tomography. The general goal is to predict the expectation values \( o_j \) for a set of \( L \) observables \( O_j \):

\[
o_j(\rho) = \text{Tr} (O_j \rho), \quad 1 \leq j \leq L,
\]

(4)

where \( \rho \) is the underlying density matrix. The procedure for constructing the predictor is described below.

First, \( \rho \) is transformed by a unitary operator \( U \) taken from the set of random unitaries \( \mathcal{U} \) as \( \rho \rightarrow U \rho U^\dagger \), and then each qubit is measured in the computational basis. For a measurement outcome \( |\hat{b}⟩ \), the reverse operation \( U^\dagger |\hat{b}⟩ ⟨\hat{b}| U \) is calculated and stored in a classical memory. The averaging operation on \( U^\dagger |\hat{b}⟩ ⟨\hat{b}| U \) with respect to \( U \in \mathcal{U} \) is regarded as a quantum channel on \( \rho \):

\[
\mathbb{E}[U^\dagger |\hat{b}⟩ ⟨\hat{b}| U] = \mathcal{M}(\rho),
\]

(5)

which implies

\[
\rho = \mathbb{E}[\mathcal{M}^{-1}(U^\dagger |\hat{b}⟩ ⟨\hat{b}| U)].
\]

(6)

The quantum channel \( \mathcal{M} \) depends on the ensemble of unitary transformations \( \mathcal{U} \). Equation (6) gives us a procedure for constructing an approximator for \( \rho \). That is, if the above measurement-reverse operation is performed \( N_{\text{shot}} \) times for different \( U \in \mathcal{U} \), then we obtain an array of \( N_{\text{shot}} \) independent classical snapshots of \( \rho \):

\[
S(\rho; N_{\text{shot}}) = \{ \hat{\rho}_1 = \mathcal{M}^{-1}(U_1^\dagger |\hat{b}_1⟩ ⟨\hat{b}_1| U_1), \ldots, \hat{\rho}_{N_{\text{shot}}} = \mathcal{M}^{-1}(U_{N_{\text{shot}}}^\dagger |\hat{b}_{N_{\text{shot}}}⟩ ⟨\hat{b}_{N_{\text{shot}}}| U_{N_{\text{shot}}}) \}.
\]

(7)

This array is called the classical shadow of \( \rho \). Once a classical shadow (7) is obtained, an estimator of \( \hat{o}_j \) can be calculated as

\[
\hat{o}_j(\rho) = \frac{1}{N_{\text{shot}}} \sum_{i=1}^{N_{\text{shot}}} \text{Tr} (O_j \hat{\rho}_i),
\]

(8)

where each \( \hat{\rho}_i \) is the classical snapshot in \( S(\rho; N_{\text{shot}}) \). Although Ref. [34] proposed to use the median-of-means estimator, we employ the empirical mean for simplicity of the implementation. Ref. [34] proved that this protocol has the following sampling complexity:

**Theorem** [34]: Classical shadows of size \( N_{\text{shot}} \) suffice to predict \( L \) arbitrary linear target functions \( \text{Tr}(O_1 \rho), \ldots, \text{Tr}(O_L \rho) \) up to additive error \( \epsilon \) given that

\[
N_{\text{shot}} \geq \mathcal{O}\left(\frac{\log(L)}{\epsilon^2} \max_j \|O_j\|_{\text{shadow}}^2\right).
\]

(9)

The definition of the shadow norm \( \|O_j\|_{\text{shadow}} \) depends on the ensemble \( \mathcal{U} \). Two different ensembles can be considered for selecting the random unitaries \( \mathcal{U} \):

**Random Clifford measurements:**

\( U \) belongs to the \( n \)-qubit Clifford group.

**Random Pauli measurements:**

\( U \) is a tensor product of single-qubit operations.

For the random Clifford measurements, \( \|O\|^2_{\text{shadow}} \) is closely related to the Hilbert-Schmidt norm \( \text{Tr}(O^2) \). As a result, a large collection of (global) observables with bounded Hilbert-Schmidt norm can be predicted efficiently. For the random Pauli measurements, on the other hand, the shadow norm scales exponentially in the locality of observable.

The above classical shadow technique can be directly applied to the problem of estimating the fidelity \( f(\theta) \) given in Eq. (3). Actually this corresponds to \( L = 1 \), \( O = O_1 = \rho_{\text{target}} \), and \( \rho = \rho_{\text{model}}(\theta) \) in Eq. (4); then, from Eq. (8), we have the estimate of \( f(\theta) \) as

\[
\hat{f}(\theta) = \frac{1}{N_{\text{shot}}} \sum_{i=1}^{N_{\text{shot}}} \text{Tr} (\rho_{\hat{\rho}_i}(\theta)),
\]

(10)

where \( \hat{\rho}_i(\theta) \) is the classical snapshot of \( \rho_{\text{model}}(\theta) \). In this case, the random Clifford measurements should be selected, because the shadow norm is given by \( \text{Tr}(\rho_{\text{target}}^2) = 1 \); also, \( N_{\text{shot}} \) becomes independent of system size, because the max \( j \|O_j\|^2_{\text{shadow}} \) term now becomes constant. For the random Clifford measurements, Ref. [34] shows that the inverted quantum channel \( \mathcal{M}^{-1} \) is given by

\[
\mathcal{M}^{-1}(\rho) = (2^n + 1) \rho - I.
\]

(11)

Lastly note that \( \mathcal{O}\left(\sqrt{n^2/\log(n)}\right) \) entangling gates are needed to do sampling from the \( n \)-qubit Clifford unitaries [39, 40], which is a practical drawback.

3. Optimization of \( U(\theta) \)

Here we describe the training method for optimizing \( U(\theta) \). First, we prepare a quantum circuit consisting of a \( n \)-qubits and \( l \) layers PQC \( U(\theta) \), which thus contains \( \mathcal{O}(ln) \) gates. The number of layers \( l \) is set to be \( \mathcal{O}(1) \sim \mathcal{O}(\text{poly}(n)) \). In particular, in this paper we take the PQC \( U(\theta) \) composed of single-qubit parameterized rotational gates \( R_x(\theta_x) = \exp(-i\theta_x \sigma_x/2) \), \( R_y(\theta_y) = \exp(-i\theta_y \sigma_y/2) \), and \( R_z(\theta_z) = \exp(-i\theta_z \sigma_z/2) \) together with CNOT gates; here \( \theta = \{ \theta_x, \theta_y, \theta_z \} \) are the \( l \)-th element of \( \theta \) and \( \sigma_x, \sigma_y, \sigma_z \) are the Pauli X, Y, Z operators, respectively. We take the so-called hardware efficient ansatz [41]; an example of the structure is shown in Fig. 1. This PQC is followed by a random Clifford unitary \( U_i \) as shown in Fig. 2. The output of the circuit is measured \( N_{\text{shot}} \) times in the computational basis, with changing the random Clifford unitary \( U_i \) and obtaining outcome

---

1 For certain cases of the random Pauli measurements, we can use Decision-Diagram based Classical Shadow to have an efficient measurement scheme [38].
Let us give a thought to the run-time for evaluating \( \rho \) the string of length \( n \) the estimate \( \hat{f} \) the fidelity \( \theta \), or \( \sigma_y/2 \) or \( \sigma_z/2 \). We randomly initialize all axes of each rotating gate (i.e., \( X, Y, \) or \( Z \)) and \( \theta \) at the beginning of each training.

\[
\langle \hat{b}_i | U_i O U_i^\dagger | \hat{b}_i \rangle \text{ is calculated as follows;}
\]
\[
\langle \hat{b}_i | U_i O U_i^\dagger | \hat{b}_i \rangle = \langle \hat{b}_i | U_i | Data \rangle \langle Data | U_i^\dagger | \hat{b}_i \rangle
\]
\[
= \left| \langle \hat{b}_i | U_i | Data \rangle \right|^2
\]
\[
= \left| \sum_{k=0}^{N-1} c_k \langle \hat{b}_i | U_i | k \rangle \right|^2.
\]

(12)

Gottesman-Knill theorem[42] allows for evaluation of \( \langle \hat{b}_i | U_i | k \rangle \) in \( O(n^2) \)-time, because \( \langle \hat{b}_i \rangle, |k\rangle \) and \( U_i \) are stabilizer states and a Clifford operator, respectively. Note that the summation in Eq. (12) requires \( O(N) = O(2^n) \) computations, which means that the required run-time in the training process of the PQC scales exponentially with the number of qubit. However, this is classical computation, which should become exponential as long as we would like to process a general exponential-size classical data. Rather, the advantage of ACAE is in the depth of the PQC, which operates only \( O(n \text{poly}(n)) \) gates instead of \( O(2^n) \), to achieve \( O(2^n) \) data encoding.

To maximize the fidelity estimate \( \hat{f}(\theta) \) (i.e., to minimize the \( -\hat{f}(\theta) \)), we take the standard gradient descent algorithm. The gradients of \( \hat{f}(\theta) \) with respect to \( \theta_r \) can be computed by using the parameter shift rule [43] as

\[
\frac{\partial \hat{f}(\theta)}{\partial \theta_r} = \hat{f}_{\theta_r^+} - \hat{f}_{\theta_r^-}, \tag{13}
\]
where
\[ \hat{f}_{\theta_r} = \hat{f}(\theta_1, \ldots, \theta_{r-1}, \theta_r \pm \pi/2, \theta_{r+1}, \ldots, \theta_R). \] (14)

\( R \) denotes the number of the parameters, which can be written as \( R = ln \) (recall that \( l \) is the number of layers of PQC). That is, the gradient can also be effectively estimated using the classical shadow. This maximization procedure will ideally bring us the optimal parameter set \( \theta^* \) and unitary \( U(\theta^*) \) that generates a state approximating the target state (2).

III. APPLICATION TO COMPACT HADAMARD CLASSIFIER

This section first reviews the method of compact amplitude encoding and the compact Hadamard classifier (CHC). Then we describe how to apply ACAE to implement CHC.

A. Compact amplitude encoding

This method encodes two real-valued data vectors into real and imaginary parts of the amplitude of a single quantum state. More specifically, given two \( N \)-dimensional real-valued vectors \( \mathbf{x}^+ = (x^+_0, \ldots, x^+_{N-1})^T \) and \( \mathbf{x}^- = (x^-_0, \ldots, x^-_{N-1})^T \), the compact amplitude encoder prepares the following quantum state:
\[ \mathbf{x}_j := \sum_{l=0}^{N-1} (x^+_l + ix^-_l) |l\rangle, \tag{15} \]
where
\[ \|\mathbf{x}_j^+\|^2 + \|\mathbf{x}_j^-\|^2 = 1, \tag{16} \]
is assumed to satisfy the normalization condition. For simplicity, here we assume \( \|\mathbf{x}_j^+\| = 1/\sqrt{2} \) without loss of generality. In addition, we define \( \mathbf{x}_j^\pm \) as
\[ \mathbf{x}_j^\pm := \frac{1}{\|\mathbf{x}_j\|} \sum_{l=0}^{N-1} x^\pm_l |l\rangle = \sqrt{2} \sum_{l=0}^{N-1} x^\pm_l |l\rangle. \tag{17} \]

B. Compact Hadamard classifier (CHC)

Here we give a quick review about CHC [18], a kernel-based quantum binary classifiers. Suppose the following training data set \( \mathcal{D} \) is given:
\[ \mathcal{D} = \{(\mathbf{x}_0, y_0), \ldots, (\mathbf{x}_{M-1}, y_{M-1})\}. \]
All inputs \( \{\mathbf{x}_j\} \) are \( N \)-dimensional real-valued vectors, and each \( y_j \) takes either +1 or −1. The goal of CHC is to predict the label \( \hat{y} \) for a test data \( \mathbf{x} \), which is also a \( N \)-dimensional real-valued vector. For simplicity, we assume that the number of training data with label +1, denoted by \( M^+ \), is equal to the number of training data with label −1, denoted by \( M^- \); i.e., \( M^+ = M^- = M/2 \) where \( M \) is an even number. In particular, we sort the training data set so that
\[ x_j = \begin{cases} x_j^+ & (0 \leq j \leq M/2 - 1) \\ x_j^- & (M/2 \leq j \leq M - 1) \end{cases} \]
and
\[ y_j = \begin{cases} +1 & (0 \leq j \leq M/2 - 1) \\ -1 & (M/2 \leq j \leq M - 1). \end{cases} \]
Note that CHC can also be applied to imbalanced training data sets as we will see later.

Assuming the existence of the compact amplitude encoder \( U_{\text{CAE}}(\mathbf{x}_j) \) that encodes the two training data \( \mathbf{x}_j^\pm \) into a single quantum state (15) and the encoder \( U_{\text{AE}}(\mathbf{x}) \) that encodes a test data \( \mathbf{x} \) into a quantum state \( |\mathbf{x}\rangle \), the following quantum state can be generated:
\[ |\psi_{\text{init}}\rangle = \frac{1}{\sqrt{2}} \sum_{j=0}^{M-1} \sqrt{b_j} \left( |0\rangle |x_j^+\rangle + e^{-i\phi} |1\rangle |\mathbf{x}\rangle \right) |j\rangle, \tag{18} \]
where \( \phi \) is the relative phase and \( b_j \) is the set of weights satisfying \( \sum_{j=0}^{M/2-1} b_j = 1 \). For example, the uniform weights can be chosen as \( b_j = 2/M \). The number of qubits required to prepare this state is \( n + m + 1 \), where \( n = \lceil \log(N) \rceil \) and \( m = \lceil \log(M/2) \rceil \). Then, we operate the single-qubit Hadamard gate on the first qubit (which we call the ancilla qubit in what follows) and have
\[ |\psi_f\rangle := H^{(a)} |\psi_{\text{init}}\rangle = \frac{1}{\sqrt{2}} \sum_{j=0}^{M-1} \sqrt{b_j} \left( |0\rangle (|x_j^+\rangle + e^{-i\phi} |\mathbf{x}\rangle) \\
+ |1\rangle (|x_j^-\rangle - e^{-i\phi} |\mathbf{x}\rangle) \right) |j\rangle. \tag{19} \]

Finally, we measure the ancilla qubit in the computational basis. The entire quantum circuit is illustrated in Fig. 3.

Now, the probabilities that the measurement outcome of ancilla qubit is |0\rangle and |1\rangle are given by
\[ \Pr(0) = \frac{1}{2} \sum_{j=0}^{M-1} b_j \left( 1 + \cos(\phi) \text{Re}(\kappa_j) - \sin(\phi) \text{Im}(\kappa_j) \right), \]
and \( \Pr(1) = 1 - \Pr(0) \), where \( \kappa_j = \langle \mathbf{x} | x_j^- \rangle \). Therefore, the expectation value of the Pauli Z operator measured on the ancilla qubit, denoted as \( \sigma_z^{(a)} \), is
\[ \langle \sigma_z^{(a)} \rangle = \sum_{j=0}^{M/2-1} b_j \left( \cos(\phi) \text{Re}(\kappa_j) - \sin(\phi) \text{Im}(\kappa_j) \right). \]
If we set $\phi = \pi/4$, this becomes

$$\langle \sigma_z^{(a)} \rangle = \frac{1}{2} \sum_{j=0}^{M-1} b_j \left( \langle \tilde{x} | x_j^+ \rangle - \langle \tilde{x} | x_j^- \rangle \right).$$  

(20)

Note now that, if the number of training data vectors in two classes are not equal (i.e., $M^+ \neq M^-$), this difference can be compensated by tuning $\phi$ to satisfy $\tan(\phi) = M_-/M_+$. We now end up with the final form of Eq. (20) as

$$\langle \sigma_z^{(a)} \rangle = \frac{1}{2} \sum_{j=0}^{M-1} b'_j y_j \langle \tilde{x} | x_j \rangle,$$  

(21)

where

$$b'_m = b'_{m+M/2} = b_m, \quad \sum_{j=0}^{M-1} b'_j = 2.$$

Clearly Eq. (21) has the form of standard kernel-based classifier where $\langle \tilde{x} | x_j \rangle$ represents the similarity between the test data $\tilde{x}$ and the training data $x_j$. Since the right-hand side of Eq. (21) represents the sum of the kernel weighted by $b'_j y_j$, the sign of $\langle \sigma_z^{(a)} \rangle$ tells us the class, or equivalently the sign of $\tilde{y}$, for the test data $\tilde{x}$, i.e.,

$$\tilde{y} = \text{sgn} \left[ \langle \sigma_z^{(a)} \rangle \right].$$  

(22)

Note that the weights $\{b'_j\}$ can be optimized, like the standard kernel-based classifier such as the support vector machine which indeed optimizes those weights depending on the training dataset; this clearly improves the classification performance of ACAE, which will be examined in a future work.

The advantage of quantum kernel-based classifiers over classical classifiers is the accessibility to kernel functions. Actually, the kernel (or similarity) between test data and training data is calculated as the inner product in the feature Hilbert space, which is however computationally expensive to evaluate via classical means when the feature space is large. On the other hand, quantum kernel-based classifiers efficiently evaluate kernel functions. In particular, CHC can evaluate the sum of all the inner products in the $N$-dimensional feature space appearing in the right-hand side of Eq. (21), just by measuring the expected value of $\sigma_z^{(a)}$. We also emphasize that CHC can be realized with compact quantum circuits compared with other quantum kernel-based classifiers. Actually, thanks to the compact amplitude encoding, two qubits can be removed in the CHC formulation compared to the others; moreover, the number of operations for encoding the training data set $x_j$ is reduced by a factor of four compared with HTC [17]. Hence, CHC can be implemented in a compact quantum circuit in both depth and width compared to the other quantum classifier, meaning that a smaller and thus easier-trainable variational circuit may function for CHC.

C. Implementation of CHC using ACAE

Although CHC efficiently realizes a compact quantum classifier, it relies on the critical assumption; that is, the quantum state (15) is necessarily prepared. Recall that, in general, the quantum circuit for generating the state (15) requires an exponential number of gates. Moreover, to generate the quantum state $|\psi_{\text{init}}\rangle$ in Eq. (18), the uniformly controlled gate $[44, 45]$ shown in Fig. 3 also requires an exponential number of gates. These requirements may destroy the quantum advantage of CHC.

The ACAE provides a method to implement CHC without using exponentially many gates. We can approximately generate the quantum state $|\psi_f\rangle$ in Eq. (19) by
replacing the component indicated by the red rectangle in Fig. 3 with $U(\theta)$:

$$H^{(a)}U(\theta)|0\rangle \approx e^{i\alpha}H^{(a)}|\psi_{\text{init}}\rangle$$

(23)

$$= e^{i\alpha} |\psi_f\rangle,$$

(24)

where $H^{(a)}$ is the Hadamard gate on the ancilla qubit and $e^{i\alpha}$ is the global phase. $U(\theta)$ is trained by the algorithm described in Section II. Although the global phase $e^{i\alpha}$ is added to $|\psi_f\rangle$, this does not affect the probability $\Pr(0)$ and the expectation value $\langle \sigma_z^{(a)} \rangle$. Recall that, this parameterized quantum circuit consists of $(n + m + 1)$ qubits with $O(1) \sim O(\text{poly}(n + m + 1))$ layers. Therefore we can realize CHC without using exponentially many gates.

IV. DEMONSTRATION

This section gives two numerical demonstrations to show the performance of our algorithm composed of ACAE and CHC. First we present an example application to a classification problem for the Iris dataset [46]. Next we show application to the fraud detection problem using the credit card fraud dataset [47] provided by Kaggle.

A. Iris dataset classification

In this subsection, we perform Iris flower classification, using CHC via encoding Iris dataset into amplitudes of a quantum state by ACAE, as a proof-of-concept experiment.

Iris dataset consists of three iris species (Iris setosa, Iris virginica, and Iris versicolor) with 50 samples each as well as four features (sepal length, sepal width, petal length, and petal width) for each flower. Each sample data includes ID number, four features, and species. IDs for 1 to 50, 51 to 100, and 101 to 150 represent data for Iris setosa, Iris versicolor, and Iris virginica, respectively. In this paper, we consider the Iris setosa and Iris versicolor classification problem. The goal is to create a binary classifier that outputs the correct label $\tilde{y} (0 : \text{Iris setosa}, 1 : \text{Iris versicolor})$ for given test data $\tilde{x} = (\text{sepal length, sepal width, petal length, petal width})$. In this demonstration, we employ the first four data of each species as training data. That is, we use data with IDs 1 to 4 and 51 to 54 as training data for Iris setosa and Iris versicolor, respectively. On the other hand, we use data with IDs 5 to 8 and 55 to 58 as test data.

First, we need to prepare a quantum state $|\psi_{\text{init}}\rangle$ given in Eq. (18) using ACAE. Since the dimension of the feature vector and the number of training data are $N = 4$ and $M = 8$, the number of required qubits is $n = \lceil \log(N) \rceil = 2$ and $m = \lceil \log(M/2) \rceil = 2$, respectively. The number of total qubits required for composing quantum circuit is $n + m + 1 = 5$, which means that $|\psi_{\text{init}}\rangle$ has $2^5 = 32$ amplitudes. We encoded the training data of the Iris setosa and Iris versicolor into the real and imaginary parts of the complex amplitude of the ancilla qubit state $|0\rangle$ and encoded the test data into the complex amplitude of the ancilla qubit state $|1\rangle$. Figure 4 shows the data contents embedded in the quantum amplitude of each basis. We train the PQC $U(\theta)$ so that it approximately generates the quantum state (18). The specific data content to be embedded in each complex amplitude is shown in Fig. 4. We use the 12-layers ansatz $U(\theta)$ illustrated in Fig. 1. We randomly initialize all the direction of each rotating gate (i.e., $X, Y$, or $Z$) and $\theta$, at the beginning of each training. As the optimizer, Adam [48] is used. The number of iterations (i.e., the number of the updates of the parameters) is set to 400 for training $U(\theta)$. For each iteration, 1000 classical snapshots are used to estimate the fidelity; that is, we set $N_{\text{shot}} = 1000$ in Eq. (8). The learning rate is 0.1 for the first 100 iterations, 0.01 for the next 100 iterations, 0.005 for the next 100 iterations, and 0.001 for the last 100 iterations. In Fig. 5, we show an example of the fidelity transition in the training process of $U(\theta)$.

As an example of the data encoding results, a set of complex amplitudes generated by ACAE is shown in Fig. 6. In the figure, the value of each complex amplitude is plotted on the complex plane. The black dots and red triangles represent the exact data and the approximate complex amplitudes embedded by ACAE, respectively. Note that the complex amplitude embedded by ACAE contains a global phase $e^{i\alpha}$ as in Eq. (23). In order to compare the exact data with the ACAE result visually, the result in which the global phase is hypothetically eliminated is also shown. Recall that the global phase does not affect the measurement in the remaining procedure.

After the state preparation, we operate the Hadamard gate on the ancilla qubit and obtain $|\psi_f\rangle$ in Eq. (19). By measuring the ancilla qubit of this quantum state and obtaining the sign of $\langle \sigma_z^{(a)} \rangle$, we can predict the label $\tilde{y}$ of the test data $\tilde{x}$, i.e., $\tilde{y} = \text{sgn} \left[ \langle \sigma_z^{(a)} \rangle \right]$. Classification results are shown in TABLE I(a) for 8 cases in which test data are IDs 5 to 8 and IDs 55 to 58. In addition to the Iris setosa and Iris versicolor classification problem, we carry out Iris versicolor and Iris virginica classification problem in the same way and show the result in TABLE I(b). All classification results are correct in TABLE I(a). On the other hand, three out of eight classification results are incorrect in TABLE I(b).

Let us discuss the results. Pairwise relationships in Iris dataset shows that Iris setosa can be clearly separated from the other two varieties, whereas the features of Iris versicolor and Iris virginica slightly overlap with each other. Therefore, classification of Iris versicolor and Iris virginica (TABLE I(b)) is considered more difficult than that of Iris setosa and Iris versicolor (TABLE I(a)). This could be the cause of the low accuracy rate of TABLE I(b). Note that the error between the exact data
In this subsection, we demonstrate credit card fraud detection as another practical application of CHC with ACAE. The goal is to encode credit card transaction data into quantum states using the PQC to detect fraud. The accuracies achieved are comparable to those of a model trained using conventional machine learning (ML) algorithms. When we compare the accuracy of the quantum model with the accuracy of a classical model, the difference increases as the number of data encoding procedures grows. Furthermore, compared to a test dataset constructed by extracting real data from the credit card transaction database, the quantum model's accuracy is higher. We also present an example of the transition of the fidelity between the target state and the model state, which is encoded by the PQC.

Nowadays, credit card fraud is a social problem in terms of customer protection, financial crime prevention, and avoiding negative impacts on corporate finances. The losses that arise from credit card fraud are a serious problem for financial institutions; according to the Nilson Report [49], credit card fraud losses are expected to reach $49.3 billion by 2030. Banks and credit card companies that pay for fraud will be hit hard by these rising costs. With digital crime and online fraud on the rise, it is more important than ever for financial institutions to prevent credit card fraud through advanced technology and strong security measures. To detect fraudulent use, financial institutions use human judgment to determine fraud based on information such as cardholder attributes, past transactions, and product delivery address information, however this method requires human resources and costs. Although attempts to detect fraud by machine learning based on features extracted from credit card transaction data also have been attracting attention in recent years, there are disadvantages such as more time and epochs to converge for a stable prediction, excessive training and so on. Quantum machine learning has the potential to solve these challenges, and its application to credit card fraud detection deserve exploring.

In this subsection, we demonstrate credit card fraud detection as another practical application of CHC with ACAE. The goal is to encode credit card transaction data...
The complex amplitudes of $|\psi_{\text{init}}\rangle$ generated by ACAE (red dots) and exact data (black dots). Note that red dots contain influence of the global phase $e^{i\alpha}$.

(b) The result in which the global phase is hypothetically eliminated is shown. The red dots are rotated clockwise about 37 degrees. The fidelity between model state (red dots) and target state (black dots) is 0.996.

FIG. 6. An example of data encoding results. Note that each data value is divided by a constant number to satisfy the normalization condition (16).

data into a quantum state as training data and classify whether a given transaction data $\tilde{x}$ is a normal ($\tilde{y} = +1$) or fraudulent transaction ($\tilde{y} = -1$).

In this demonstration we use the credit card fraud detection dataset [47] provided by Kaggle. The dataset contains credit card transactions made by European cardholders in September 2013. The dataset has 284,807 transactions which include 492 fraudulent transactions. Note that, since it is difficult to encode all transaction data into a quantum state, for proof-of-concept testing, we take 4 normal transaction data and 4 fraudulent transaction data from this dataset as training data, and perform classification tests to determine whether given test data is normal or fraudulent. Each data consists of $\text{Times}$, $\text{Amount}$ and 28 different features ($V_1, V_2, \ldots, V_{28}$) transformed by principal component analysis. We use 4 features ($V_1, V_2, V_3, V_4$) out of the 28 features for classification, which means that the dimension of the feature vector and the number of training data are $N = 4$ and $M = 8$ respectively, and the number of total qubits required for composing quantum circuit is $n + m + 1 = 5$. As in the previous subsection, we embed the training data of normal transaction and fraudulent transaction into the real and imaginary parts of the complex amplitude of the ancilla qubit state $|0\rangle$ and embed the test data into the complex amplitude of the ancilla qubit state $|1\rangle$.

After the state preparation, we operate the Hadamard gate on the ancilla qubit and make measurements to obtain $\langle \sigma_3^{(\alpha)} \rangle$ the sign of which provides classification result, i.e., $\tilde{y} = +1$ for normal data and $\tilde{y} = -1$ for fraudulent data. As test data, we take the top four normal data ID and fraudulent data ID, in ascending order, excluding the training data. Classification results are shown

### Table I. The results of the Iris dataset classification problem

#### (a) Iris setosa versus Iris versicolor

| test data ID | class      | $\langle \sigma_3^{(\alpha)} \rangle$ | $\tilde{y}$ | result  |
|--------------|------------|--------------------------------------|-------------|---------|
| #5           | Iris setosa| 0.0356                               | +           | Correct |
| #6           | Iris setosa| 0.0393                               | +           | Correct |
| #7           | Iris setosa| 0.0645                               | +           | Correct |
| #8           | Iris setosa| 0.0466                               | +           | Correct |
| #55          | Iris versicolor | -0.0345                          | -           | Correct |
| #56          | Iris versicolor | -0.0513                          | -           | Correct |
| #57          | Iris versicolor | -0.0367                          | -           | Correct |
| #58          | Iris versicolor | -0.0361                          | -           | Correct |

#### (b) Iris versicolor versus Iris virginica

| test data ID | class      | $\langle \sigma_3^{(\alpha)} \rangle$ | $\tilde{y}$ | result  |
|--------------|------------|--------------------------------------|-------------|---------|
| #55          | Iris versicolor | -0.0179                          | +           | Correct |
| #56          | Iris versicolor | -0.0186                          | -           | Incorrect |
| #57          | Iris versicolor | -0.0170                          | -           | Incorrect |
| #58          | Iris versicolor | 0.0158                           | +           | Correct |
| #105         | Iris virginica | -0.0080                          | -           | Correct |
| #106         | Iris virginica | -0.0145                          | -           | Correct |
| #107         | Iris virginica | -0.0381                          | -           | Correct |
| #108         | Iris virginica | 0.0287                           | +           | Incorrect |

#### (c) Iris versicolor versus Iris virginica (Exact data are used.)

| test data ID | class      | $\langle \sigma_3^{(\alpha)} \rangle$ | $\tilde{y}$ | result  |
|--------------|------------|--------------------------------------|-------------|---------|
| #55          | Iris versicolor | 0.0322                           | +           | Correct |
| #56          | Iris versicolor | 0.0013                           | +           | Correct |
| #57          | Iris versicolor | 0.0230                           | +           | Correct |
| #58          | Iris versicolor | 0.0534                           | +           | Correct |
| #105         | Iris virginica | -0.0392                          | -           | Correct |
| #106         | Iris virginica | -0.0269                          | -           | Correct |
| #107         | Iris virginica | -0.0430                          | -           | Correct |
| #108         | Iris virginica | -0.0194                          | -           | Correct |

---

2 We take top 4 normal data ID in ascending order, specifically #1, #2, #3 and #4.

3 We take top 4 fraudulent data ID in ascending order, specifically #524, #624, #4921 and #6109.
in TABLE II. Although the result of #7 is incorrect, it arises primarily from the error between the exact data and ACAE data as in the case of Iris dataset classification. We confirmed that the value of \( \langle \sigma_z^{(n)} \rangle \) turns 0.0199 and the correct result is obtained when the classification is performed using exact data.

| test data ID\(^1\) | class | \( \langle \sigma_z^{(n)} \rangle \) | \( \hat{y} \) | result |
|-------------------|-------|-----------------|-----|--------|
| #5                | normal| 0.6512          | +   | Correct|
| #6                | normal| 0.1684          | +   | Correct|
| #7                | (+)   | -0.0113         | -   | Incorrect|
| #8                |       | 0.1447          | +   | Correct|
| #6330             | fraud | -0.4012         | -   | Correct|
| #6332             |       | -0.4021         | -   | Correct|
| #6335             | (-)   | -0.3931         | -   | Correct|
| #6337             |       | -0.4246         | -   | Correct|

\(^1\) As test data, we take the top four normal data ID and fraudulent data ID, in ascending order, excluding the training data.

V. CONCLUSION

In this paper, we proposed the approximate complex amplitude encoding algorithm (ACAE) which allows for the efficient encoding of given complex-valued classical data into quantum states using shallow parameterized quantum circuits. The key idea of this algorithm is to use the fidelity as a cost function, which can reflect the difference in complex amplitude between the model state and the target state, unlike the MMD-based cost function in the original AAE. Also note that the classical shadow with random Clifford unitary is used for efficient fidelity estimation. In addition, we applied ACAE to realize CHC with fewer gates than the original CHC which requires an exponential number of gates to prepare the exact quantum state. Using this algorithm we demonstrated Iris data classification and the credit card fraud detection that is considered as a key challenge in financial institutions.

For practical applications of ACAE, we need to deal with enormously large dataset. For example, in the demonstration for credit card fraud detection, approximately 280,000 training data are provided and each data has 28 different features. This means \( n + m + 1 = [\log(28)] + [\log_2(280,000)/2] + 1 = 23 \) qubits are required to deal with the whole data. As the number of qubits is increased, the degree of freedom of the quantum state exponentially grows; in such case, there appear several practical problems to be resolved. For example, ACAE employs the fidelity as a cost function, which is however known as a global cost that leads to the so-called gradient vanishing problem or the barren plateau problem [50]; i.e., the gradient vector of the cost decays exponentially fast with respect to the number of qubits, and thus the learning process becomes completely stuck for large-size systems. To mitigate this problem, recently the localized fidelity measurement has been proposed in Ref. [51–53]. Moreover, applications of several existing methods such as circuit initialization [54, 55], special structured ansatz [56, 57], and parameter embedding [58] are worth investigating to address the gradient vanishing problem.

Another problem from a different perspective is that the random Clifford measurement for producing the classical shadow can be challenging to implement in practice, because \( O(n^2/\log(n)) \) entangling gates are needed to sample from \( n \)-qubit Clifford unitaries. Ref. [59, 60] presented that Clifford circuit depth over unrestricted architectures is upper bounded by \( 2n + O(\log^2(n)) \) for all practical purposes, which may improve the implementation of the fidelity estimation process. Overall, algorithm improvements to deal with these problems are all important and still remain as future works.

ACKNOWLEDGMENTS

This work was supported by Grant-in-Aid for JSPS Research Fellow 22J01501 and MEXT Quantum Leap Flagship Program Grants No. JPMXS0118067285 and No. JPMXS0120319794. We acknowledge the use of IBM Quantum services 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 Quantum team. We specially thank Kohei Oshio, Ryo Nagai and Yuki Sato for their comments on the early version of this draft.

[1] P. Wittek. Quantum machine learning: what quantum computing means to data mining. Academic Press, 2014.
[2] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe and S. Lloyd. Quantum machine learning. Nature, 549(7671):195–202, 2017.
[3] M. Schuld and F. Petruccione. Supervised learning with quantum computers, volume 17. Springer, 2018.
[4] V. Dunjko and H. J. Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Reports on Progress in Physics, 81(7):074001, 2018.
[5] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini and L. Wossnig. Quantum machine learning: a classical perspective. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 474(2209):20170551, 2018.
[6] M. Schuld. Supervised quantum machine learning models are kernel methods. arXiv preprint arXiv:2101.11020, 2021.
[7] V. Giovannetti, S. Lloyd and L. Maccone. Quantum random access memory. Physical review letters, 100(16):160501, 2008.
Credit card fraud detection. [https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud).

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.

Nilson report, 2021. [https://nilsonreport.com/upload/content_promo/NilsonReport_Issue1209.pdf](https://nilsonreport.com/upload/content_promo/NilsonReport_Issue1209.pdf).

J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush and H. Neven. Barren plateaus in quantum neural network training landscapes. Nature communications, 9(1):1–6, 2018.

S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger and P. J. Coles. Quantum-assisted quantum compiling. Quantum, 3:140, 2019.

K. Sharma, S. Khatri, M. Cerezo and P. J. Coles. Noise resilience of variational quantum compiling. New Journal of Physics, 22(4):043006, 2020.

H. Tezuka, S. Uno and N. Yamamoto. Generative model for learning quantum ensemble via optimal transport loss. arXiv preprint arXiv:2210.10743, 2022.

E. Grant, L. Wossnig, M. Ostaszewski and M. Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. Quantum, 3:214, 2019.

K. Zhang, M.-H. Hsieh, L. Liu and D. Tao. Gaussian initializations help deep variational quantum circuits escape from the barren plateau. arXiv preprint arXiv:2203.09376, 2022.

M. Cerezo, A. Sone, T. Volkoff, L. Cincio and P. J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nature communications, 12(1):1–12, 2021.

X. Liu, G. Liu, J. Huang and X. Wang. Mitigating barren plateaus of variational quantum eigensolvers. arXiv preprint arXiv:2205.13539, 2022.

T. Volkoff and P. J. Coles. Large gradients via correlation in random parameterized quantum circuits. Quantum Science and Technology, 6(2):025008, 2021.

D. Maslov and B. Zindorf. Depth optimization of CZ, CNOT, and Clifford circuits. IEEE Transactions on Quantum Engineering, 2022.

D. Maslov and W. Yang. CNOT circuits need little help to implement arbitrary hadamard-free clifford transformations they generate, 2022.