Vol.2020-QS-1 No.18
2020/10/16

Study on Use Cases and Robustness of Quantum Random Access Coding in Quantum Machine Learning

Napat Thumwanit[1,a)] Chayaphol Lortararprasert[1] Rudy Raymond[2,3]

Abstract: Quantum classifiers provide sophisticated embeddings of input data in Hilbert space promising quantum advantage. The advantage stems from quantum feature maps encoding the inputs into quantum states with variational quantum circuits. A recent work shows how to map discrete features with fewer quantum bits using *Quantum Random Access Coding* (QRAC), an important primitive to encode binary strings into quantum states. We propose a new method to embed discrete features with trainable quantum circuits by combining QRAC and a recently proposed strategy for training quantum feature map called *quantum metric learning*. We show that the proposed trainable embedding requires not only as few qubits as QRAC but also overcomes the limitations of QRAC to classify inputs whose classes are based on hard Boolean functions. We numerically demonstrate its use in variational quantum classifiers to achieve better performances in classifying real-world datasets, and thus its possibility to leverage near-term quantum computers for quantum machine learning.

1. Introduction

Quantum computers can execute algorithms based on the principles of quantum mechanics and can provide quantum speedups over classical computers in wide-range applications, including machine learning [1]. There are many proposed quantum algorithms for machine learning that are proven superior to their classical counterparts, such as, solving linear equations [2] and principal component analysis [3], all of which require perfect quantum computers. Recent noisy quantum devices [4] are built towards the goal of realizing such perfect quantum computers with the progress comparable to that of the Moore's Law [5]. Nevertheless, although limited in the number and quality of their quantum bits (or, qubits), the recent hardware progress has sparked active studies in the search of quantum methods that are applicable to near-term quantum devices [6], [7], [8].

Variational Quantum Classifiers (VQC) is a potential machine learning method on near-term quantum devices for classifying classical data. VQC is built from variational quantum circuits whose parameters are tuned by coordinating the noisy quantum devices and perfect classical computers. It finds a quantum feature map embedding classical data into a high-dimensional Hilbert space, and a measurement set to perform the classification. The most popular type of VQCs fixes the quantum feature map for each classical data, and trains a measurement for the classification. This is akin to quantization of the classical kernel methods, such as the polynomial support vector machine (SVM) [9]. There have been various proposals of quantum feature maps embedding real-valued data with near-term quantum devices [8], [10], [11], [12]. A recent work [13] shows how to use QRAC in VQCs for mapping binary strings into the Hilbert space. The resulting VQCs are shown to require fewer qubits while retaining high classification accuracy.

QRAC encodes m classical bits with n qubits so that any 1-out-of-m bits can be recovered with probability $p > 1/2$ [14]. It is often denoted as (m, n, p)-QRAC). When $p = 1$, the Holevo bound [15] implies that $n \geq m$ meaning no quantum advantage. But when $1/2 < p < 1$, QRAC can achieve provable quantum advantage, e.g., in quantum communication [16], [17]. In particular, when p is enough to be bigger than $1/2$, the number of qubits of QRAC is shown exactly half of the bits for classical encoding [18].

The advantage of QRACs stems from the facts that (*i*) the n-qubit quantum states are spanned by the 2^n-dimensional Hilbert space whose real dimension is $2^{2n} - 1$, and (*ii*) bits of length up to $2^{2n} - 1$ can be mapped into quantum states distributed in the Hilbert space. The QRACs for mapping discrete features in VQCs in [13] are mainly with the well-known construction of (m, n, p)-QRACs for $n = 1, 2$. Main challenges in using QRACs for quantum feature map are their limited known construction, namely, no optimal QRAC is known for three or more qubits, and the difficulties to map bitstrings to quantum states that guarantees maximally separating embeddings. The latter one is serious because the fixed mapping by QRACs can result in an arrangement of quantum states which is impossible to find its boundary decision without adding more resources such as latent qubits or copies

1 University of Tokyo
2 IBM Quantum, IBM Research — Tokyo
3 Quantum Computing Center, Keio University
a) thumwanit-napat@g.ecc.u-tokyo.ac.jp

ⓒ 2020 Information Processing Society of Japan

of encoding states. This can mitigate the benefits of QRACs to map discrete features with fewer qubits.

To solve the aforementioned challenges, we extend the use of QRACs to embed discrete features with the new strategy of *quantum metric learning* [19] to obtain trainable embeddings. The trainable discrete feature embeddings can find well-separated quantum states in the Hilbert space by taking into account the property of QRACs to use a set of diverse quantum states for encoding discrete features. By numerical experiments, we show the limitations of QRACs on classifying inputs whose decision depends on hard Boolean functions called *parity*, and show how trainable embeddings can overcome them. We also provide evidences the trainable embeddings can maintain the advantage of QRACs to achieve high classification accuracy with fewer qubits on various real-world datasets such as the breast cancer dataset [20], the Titanic survival datasets [21], and the simplified version of MNIST [22]. We believe the trainable embeddings provide new insights on how to levarage near-term quantum devices for machine learning.

2. Related Work

Dealing with discrete features, representing qualitative or categorical properties, is important because many real-world datasets contain such features, such as, gender, race, age group, etc. We propose a new framework to embed the discrete features in VQC for quantum machine learning. Classical machine learning models such as decision trees [23] can deal directly with discrete features, but models such as neural-network often rely on transforming them into continuous features [24]. Standard transformations include entity embeddings [25] and distributed representations [26]. Most successful transformation often rely on combination of transformation methods, e.g., *word2vec* [27] obtains continuous representation of categorical features from the use of one-hot encoding and neural networks.

Quantum-enhanced machine learning techniques such as VQCs [8], [10] are quite similar to neural networks and their framework are often developed by assuming appropriate representation of datasets with real-valued and continuous features. VQCs consists of a series of quantum operations acting on qubits represented as a quantum circuit. The circuit can be divided into two: the part for mapping classical data into quantum states, and the part for measuring the quantum states to obtain the classification classes. The mapping part is usually fixed and there are many potential methods that maybe difficult to realize with classical computers. Recently, [19] proposes trainable quantum feature mapping (or, embedding) called quantum metric learning, akin to the classical metric learning [28], so that the measurement part can be simplified. The measurement part is usually a parameterized circuits whose optimal parameters are computed with variational methods similar to those of neural networks. A recent work [13] shows how to directly deal with discrete features in VQCs by using QRACs that can reduce computational resources of VQCs.

Originally formulated in the communication setting [29], QRACs have been extensively used in the theory of quantum computations, such as, quantum state learning [30]. Some QRACs also offer cryptographic properties known as *parity obliviousness* [31], which can be useful for private information retrieval. QRACs can encode n-bit binary strings using $\log n/2$ qubits, which is half of the bits used in classical random access codes (RACs) [18]. QRACs with one and two qubits were shown in [14], [32], [33]. One qubit can encode at most three bits, and two qubits at most fifteen bits, and so on. [13] shows how to use them for healthcare datasets, but there are still many obstacles to deal with complicated decision functions. Such QRACs may be used as a layer in a quantum neural network (QNN) [34], [35].

There are variants of QRACs using shared entanglement and classical randomness [36], that enable encoding any number of bits into a single qubit. Like the QRACs and our proposed embedding, those variants rely on finding a set of quantum states that are as diverse as possible within the Hilbert space of the underlying qubits.

3. Methods

Our method deals with embedding discrete features of inputs into the Hilbert space and finding the boundary decision using VQCs. We briefly describe the quantum circuits, VQCs and their embeddings before showing the proposed method of Trainable Embeddings (TEs).

3.1 Preliminaries

3.1.1 Quantum Circuit (QC)

QC is popular to model quantum computation. It consists of sequences of quantum gates that transform qubits from initial quantum states (usually the all-zero state) into final quantum states that can be measured to obtain the computational result (such as, the classification). An n-qubit (pure) quantum state $|\psi\rangle$ is a linear superposition of all possible n-bit classical states $|i\rangle$ for $i = 0, \ldots, 2^n - 1$. Namely, $|\psi\rangle = \sum_i \alpha_i |i\rangle$, where $\alpha_i \in \mathbb{C}$ is the probability amplitude whose squared magnitude is the probability of measuring $|i\rangle$, and thus $\sum_i \backslash\texttt{alpha_i}\backslash^2 = 1$. The n-qubit quantum state is thus a complex vector lies in a 2^n-dimensional Hilbert space. Quantum gates are unitary operations that can be represented by a series of $2^n \times 2^n$ unitary matrices $\{\mathbf{U}_i\}$ rotating 2^n-dimensional complex vectors. All unitary operators can be realized with a small set of universal gates that consists of arbitrary one-qubit rotation and two-qubit entangling gates, such as the controlled-NOT and the ZZ-interaction gate. See [37] for more details on QC. The near-term quantum computers are those with dozens to hundreds of qubits, and can only run few quantum gates. Therefore, all quantum machine learning leveraging ner-term devices must be designed with shallow QCs in mind.

3.1.2 Variational Quantum Classifiers (VQC)

VQC is a classifier using variational quantum circuit that can be trained from labelled inputs, $S = \{(\boldsymbol{x}_i, y_i)\}_i$ for $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. As in Fig. 1, it consists of two parts: the first implementing feature map of input data, namely, $\boldsymbol{x} : \boldsymbol{\Phi}(\boldsymbol{x}) \in \mathbb{R}^k$ for $k \geq d$, and the second implementing measurement to determine the classification. Its principle is similar to support vector machine (SVM) [38] for finding the best hyperplane $(\boldsymbol{w}, b)$ that classifies the embedded data by the linear boundary function $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x}) + b$.

ⓒ 2020 Information Processing Society of Japan

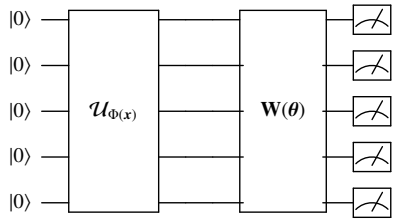

Fig. 1 A VQC that consists two subcircuits: $\mathcal{U}_{\Phi(\boldsymbol{x})}$ for mapping features and $\mathbf{W}(\boldsymbol{\theta})$ for boundary decision.

At VQC, the data $\boldsymbol{x} \in \mathbb{R}^d$ is mapped to (pure) quantum state by the feature map circuit $\mathbf{U}_{\Phi(\boldsymbol{x})}$ that realizes $\Phi(\boldsymbol{x})$. That is, conditioned on the data $\boldsymbol{x}$ we apply the circuit $\mathbf{U}_{\Phi(\boldsymbol{x})}$ to the n-qubit all-zero state $|0_n\rangle$ to obtain the quantum state $|\Phi(\boldsymbol{x})\rangle$. A short-depth quantum circuit $\mathbf{W}(\boldsymbol{\theta})$ is then applied to the quantum state, where $\boldsymbol{\theta}$ is the hyperparameter set of the quantum circuit that can be learned from the training data. Finding the circuit $\mathbf{W}(\boldsymbol{\theta})$ is akin to finding the separating hyperplane $(\boldsymbol{w}, b)$ in the Soft-SVM, with the promise of quantum advantage due to the difficulty for classical procedures to realize the feature map $\Phi(\boldsymbol{x})$.

Learning the best $\boldsymbol{\theta}$ can be obtained by minimizing the empirical risk $R(\boldsymbol{\theta})$ with regards to the training data $S = \{(\boldsymbol{x}_i, y_i)\}$. The risk is then approximated with a continuous function, as detailed in [8], to enable variational methods for tuning $\boldsymbol{\theta}$ to minimize the approximated cost function.

The binary classification with VQC now follows from first fixing the feature maps on inputs and then training the VQC to learn the best $\boldsymbol{\theta}^*$ to obtain $(\boldsymbol{w}(\boldsymbol{\theta}^*), b^*)$. The classification against unseen data $\boldsymbol{x}$ is then performed according to the classifier function $f(\boldsymbol{x})$ with $(\boldsymbol{w}(\boldsymbol{\theta}^*), b^*)$. Both training and classification need to be repeated for multiple times (or, shots) due to the probabilistic nature of quantum computation.

3.1.3 Quantum Neural Networks (QNN)

We use the QNN proposed by [34] for binary classification with binary inputs $\boldsymbol{x} \in \{0,1\}^n$. It is a special case of VQC in Fig. 1. At QNN two quantum registers are used; the first to represent the n-bit input and the second for its label, and thus requires at least $n+1$ qubits (omitting the working qubits used in the middle of the computation). QNN is VQC with simple n-qubit feature map $\Phi(\boldsymbol{x})$ to create the quantum state representing computational basis $|\boldsymbol{x}\rangle|0\rangle = \mathbf{U}_{\Phi(\boldsymbol{x})}|0\rangle^{\otimes n}|0\rangle$. This can realized with the $\mathbf{X}$ gate at the i-th qubit for $x_i = 1$.

At QNN $\mathbf{W}(\boldsymbol{\theta})$ is a sequence of two-qubit unitaries $\mathbf{U}_1(\theta_1), \mathbf{U}_2(\theta_2) \ldots, \mathbf{U}_l(\theta_l)$. The optimization target is to find the set of parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_l\}$ that minimizes the training loss of dataset $\mathcal{D}$ such as binary cross-entropy or hinge loss. The evolution of the state vector due to $\mathbf{W}(\boldsymbol{\theta})$ is $|\psi\rangle = \mathbf{W}(\boldsymbol{\theta})|\boldsymbol{x}, 0\rangle = \sum_i (\alpha_{i0}|i,0\rangle + \alpha_{i1}|i,1\rangle)$, where the classification l is obtained by measuring the second register, i.e., $l = 0$ iff. $\sum_{i0}|\alpha_{i0}|^2 \geq \sum_{i1}|\alpha_{i1}|^2$. [34] showed how to perform supervised learning on QNN for any n-bit Boolean functions, and thus for binary classification of discrete inputs. In particular, numerical experiments showed that QNN could learn the parity function of a subset of the input bits with samples much less than 2^n even under some noisy labels. However, [34] pointed out that the learning became impossible as the subsets became larger. This motivates our choice of the parity functions in Experiments.

3.1.4 Discrete Feature Encoding with QRAC

Discrete features can be trivially mapped into binary strings and be used as inputs to VQCs. Here, we consider binary representation. For example, a categorical feature with at most c different categories can be represented with $\lceil \log c \rceil$ bits and thus requires the same amount of qubits at VQCs. In practice, if there are k types of discrete features, and each of the features has c_i categories, then the number of bits representing all the features is at most $S = \sum_{i=1}^{k} \lceil \log_2(c_i) \rceil$.

[13] showed how to use QRAC to reduce the number of qubits representing the categorical features. In particular, it is known how to encode bitstrings of length 2 and 3 by $(2,1)$-QRAC and $(3,1)$-QRAC (we omit the success probabilities p) by mapping each bitstring into a one-qubit quantum state. The left figure in Fig. 2 depicts the states of $(3,1)$-QRACs that are placed at the corners of a cube in the Bloch sphere. Due to this property of evenly placing quantum states, QRAC ensures that each quantum encoding of a string can be distinguished from other encodings. This can be advantageous for classification with few qubits. By using (m,n)-QRAC the number of qubits required for encoding S bits is $n \times \lceil S/m \rceil$.

The discrete feature encoding by QRAC is easily combined with continuous features. For discrete feature $\boldsymbol{x}_d$ and continuous feature $\boldsymbol{x}_c$, $\boldsymbol{x}_d$ can be embedded using (m,n)-QRAC and $\boldsymbol{x}_c$ using some continuous feature map such as the ZZ feature map [*1]. We can also apply QRAC in the convolutional manner that we call *C-QRAC* (for *Convolutional*-QRAC). For example, with regard to binary strings $\boldsymbol{b} = b_1 b_2 \ldots b_t$, instead of encoding it sequentially, the convolutional manner can encode its overlapped substrings. The convolutional preprocessing is effective for image datasets, and therefore we believe so is the C-QRAC for MNIST datasets.

3.2 Trainable Embedding (TE)

There are several drawbacks of mapping the above features with QRAC that fixes mapping of binary strings to quantum states. First, there are only few concrete constructions of QRACs [14], [32], [33]. Second, for some Boolean functions such as the parity, the fixed mapping of QRAC may result in mapping inputs to quantum states that are not "well-separated" in the Hilbert space. This can easily be seen from mapping 8 binary strings $\boldsymbol{x} \in \{0,1\}^3$ into one-qubit quantum states that lie in 3-dimensional Bloch sphere with QRAC as in the left subfigure of Fig. 2; because the VC-dimension of the Boolean functions with 3 variables is at least 4, VQCs with QRAC can fail to find the boundary decision in the Bloch sphere. Previous work [13], [19] suggest adding more qubits or copies of QRACs, to avoid this problem but it reduces the advantage of QRACs. We propose trainable embedding to avoid the drawbacks.

Instead of fixing the encoded quantum spaces of the input binary strings using (m,n)-QRAC, we propose a method by applying a trainable circuit which is inspired by the quantum metric learning [19]. The circuit is learned with an objective to obtain maximally separating data classes in the Hilbert space. By adap-

[*1] The details of the ZZ feature map can be found at `https://qiskit.org/documentation/stubs/qiskit.circuit.library.ZZFeatureMap.html`

ⓒ 2020 Information Processing Society of Japan

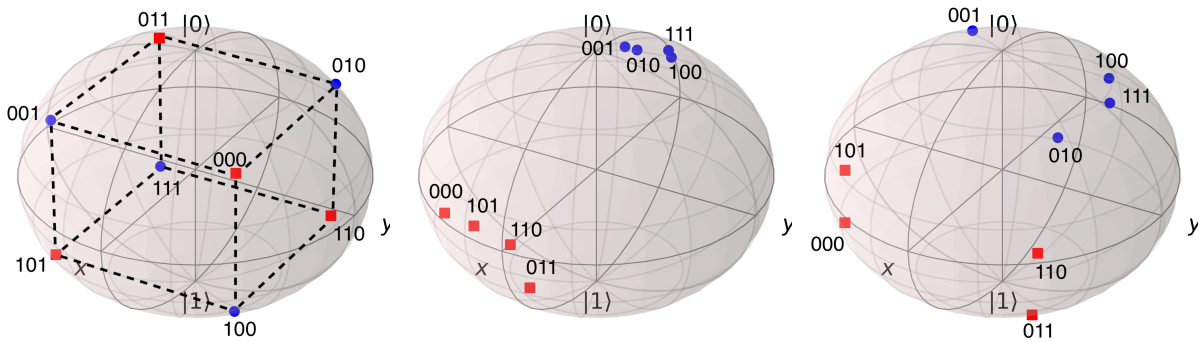

Fig. 2 The embeddings of 3-bit discrete features into one-qubit quantum states drawn as colored dots on Bloch spheres whose north and south poles correspond to the classical states "0" and "1", respectively. The embedding on the left Bloch sphere is by the $(3, 1)$-QRAC, while those on the middle and the right Bloch spheres are by TEs without and with regularization (as in Eq. 1 with $\lambda = 0.02$), respectively. The blue dots denote those with parity 1, and the red dots parity 0. It is clear that a $(3, 1)$-QRAC is insufficient to solve the parity as there is no linear plane can separate the red from the blue dots.

tively training feature embeddings to separate the data classes, [19] showed that the best measurements to obtain the classification can be realized with shallow quantum circuits and hence suitable for near-term quantum devices.

The trainable circuits are represented as a set of unitary $\mathbf{U}(\theta_{\boldsymbol{b}})$ matrices where $\boldsymbol{b}$ is the binary string input and $\theta_{\boldsymbol{b}}$ is the set of parameters corresponding to $\boldsymbol{b}$. Namely, (m, n)-QRAC is replaced with a trainable quantum circuit realizing $\mathbf{U}(\theta_{\boldsymbol{b}})$. For example, the $(3, 1)$-QRAC is replaced with trainable embedding using $u3(\theta, \varphi, 0)$ gates for each 3-bit binary string, where θ and φ are trainable parameters[*2]. Therefore, there are 16 parameters to be trained for each qubit encoding 3 bits. For convenience, we denote $(3, 1)$-TE as the trainable embedding of 3 bits into 1 qubit.

However, the increasing number of parameters might possibly lead to overfitting. The center subfigure of Fig. 2 shows the result of $(3, 1)$-TE for the parity of the bits. As can be shown from the left of the same figure, with QRAC the distance between any two embedded states is guaranteed to be above some threshold. On the other hand, the embedded states of the $(3, 1)$-TE are clustered as shown in the middle of the same figure.

To guide the embedded states to become as diverse as those with QRACs, and so that each of them is far for the rest, we introduce a cost to penalize the entropy of the overall embedding vectors. Let us assume $\{(\theta_i, \varphi_i)\}$ be the parameter for encoding 3-bits binary string i. The embedding for each i is therefore $u3(\theta_i, \varphi_i, 0)\,|0\rangle = \cos(\theta_i/2)\,|0\rangle + e^{\varphi_i}\sin(\theta_i/2)\,|1\rangle$ which corresponds to the point $\boldsymbol{r}_i = (\sin\theta_i\cos\varphi_i, \sin\theta_i\sin\varphi_i, \cos\theta_i)$ on the 3-dimensional Bloch sphere[*3]. From these embedding vectors $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_8$, we can compute their averages and covariance matrix, as

$$\boldsymbol{\mu} = \frac{1}{n}\sum_i \boldsymbol{r}_i$$

$$\boldsymbol{\Sigma} = \frac{1}{n^2}\sum_{i,j}(\boldsymbol{r}_i - \boldsymbol{\mu})(\boldsymbol{r}_j - \boldsymbol{\mu})^\top,$$

where n is the number of the embedding. In $(3, 1)$-TE case, we have $n = 8$ number of embedding. The determinant of the covariance matrix can be used to scale the spread factor of the embedding. By adding the negative of the determinant of the covariance matrix, i.e., $\mathcal{L}_{\text{spread}} = -\det(\boldsymbol{\Sigma})$ to the training loss to find optimal embedding parameters, we minimize the following regularized loss function:

$$\mathcal{L}_{\text{all}} = \mathcal{L} + \lambda\mathcal{L}_{\text{spread}}. \tag{1}$$

The effect of regularization is shown in the right subfigure of Fig. 2. We can see a wider spread of distribution of embedded quantum state by the regularization. A similar regularization by determinant has also been used to induce diversity of state-action vectors in reinforcement learning [40].

As a comparison to classical machine learning method, TE would be similar to embedding feature that prevalently used with discrete feature such as word embedding [27] to map a discrete word token into a continuous vector space.

4. Experiments

We perform numerical experiments demonstrating the limitation of discrete feature embedding with QRAC and the possibility of trainable embedding to overcome this limitation. To this end, the embeddings are tested on parity functions, which are known as hard Boolean functions, and on real-world datasets: the Breast Cancer dataset (BC), the Titanic Survival dataset (TS), and the hand-written digits MNIST dataset. On parity functions, the experiments showed that the TE achieved higher accuracies with fewer qubits than straightforward and QRAC embeddings. We also confirmed the advantage of TE for classifying the real-world datasets. The experiments were run by Qiskit using its RyRz-variational VQCs [*4]. We also confirmed the efficacy of the em-

[*2] $u3$ gate is a standard one-qubit rotational gate in Qiskit[39]. The last parameter of the $u3$ gate is fixed to zero because it does not affect the transformation starting from the zero state.

[*3] See, e.g., [37] for the correspondence of one-qubit (pure) states with points on the Bloch sphere

[*4] The details of the RyRz variational form can be found at

ⓒ 2020 Information Processing Society of Japan

beddings on TensorFlow Quantum [41] by experimenting with QNN on the MNIST dataset.

4.1 VQC on Parity

The parity function is determined by the XOR of its individual input bits (or equivalently, the sum of its input bits modulo 2). For example, for 3-bit inputs the parity function is 1 if and only if the input is in $\{001, 010, 100, 111\}$, as colored in blue in Fig. 2. We tested the embeddings on parity of bits whose lengths are $3, 6$ and 9. As can be seen from the left subfigure in Fig. 2, with the QRAC embedding the 3-bit parity is not linearly separable. This is the main motivation to use the parity function as a test case for the embeddings.

We compared the efficacy of VQC with the embeddings by calculating the correctly classified ratio (or precision). All VQCs were optimized using SPSA [42] for 200 epochs. The embeddings were as below:

- **Naive**: a qubit is used to represent a bit.
- n×**QRAC**: n copies of $(3, 1)$-QRAC encoding is used to represent the binary string. $n > 1$ is suggested to increase the chance finding a decision boundary in the Hilbert space [13].
- **TE**: The $(3, 1)$-TE is used to encode the binary string.

Table 1 shows the classified ratios of the VQCs with the embeddings. We can observe that TE achieved better precisions on the 6-bit and 9-bit parity despite using the fewest number of qubits. On the 3-bit parity, its precision was only slightly less than Naive and 3 copies of QRACs, but notice that both Naive and QRACs could not reduce the qubits.

The parity of 6-bit and 9-bit showed interesting cases because TE with only 2 and 3 qubits, respectively, could beat Naive and QRACs that essentially store all information of the bits (by the same number of qubits). In particular, the copies of QRACs seem to only slightly increase the precision. We can further confirm the limitation of QRACs and the advantage of TE from Fig. 3 that shows the training losses of the VQCs. TEs clearly outperformed other embeddings, while the losses of 3 copies of QRACs were only as good as Naive, which is expected. This is because in that case one can extract all values of bits from the qubits.

4.2 VQC on Breast Cancer dataset (BC)

The BC dataset contains 286 samples, each of which consists of 9 discrete features, along with a cancer recurrence as a target variable (201 samples of no-recurrence, 85 samples of recurrence). We selected 4 most important features determined by a random forest algorithm [43], which are *tumor-size*, *breast-quad*, *deg-malig*, and *age*. The features have 11, 6, 3, and 6 different categorical values respectively.

Due to imbalance in the labels of the dataset, we doubled the positively-labeled data in the training set to encourage true positive predictions, and evaluated F1 score in addition to accuracy (precision). We tested the VQCs with the RyRz variational form, all of which were optmized using SPSA for 200 epochs, using the four different embeddings as follows.

- **ZZ**: the baseline mapping for real-valued features as used in [8]. Each feature is simply encoded into an ordinal value, and then embedded using the ZZ feature map (ZZ).
- **QRAC**: the mapping used in [13]. Each feature is encoded into an ordinal value, and then converted into a bit string. The concatenated bit string is encoded with $(3, 1)$-QRACs.
- **TE**: similar to QRAC up to the concatenated bit string. $(3, 1)$-TE is used to embed the concatenated string. Reg.TE denotes the a regularized model as in Eq. 1 with $\lambda = 0.02$ to prevent overfitting.

The experimental results shown in Table 2 suggest that feature maps with QRAC and TEs significantly outperformed ZZ feature map (ZZ), with TEs had better accuracies than QRAC. Without regularization, however, TE showed slight overfitting to the training data. TE with regularaization (Reg.TE) avoided the overfitting.

4.3 VQC on Titanic Survival dataset (TS)

The TS dataset consists of 11 features, including both discrete and continuous ones. The target variable is the survival of the passengers. This dataset is split into training nd test sets whose size are 891 and 418 samples respectively.

We selected 4 most important features as ranked again by the random forest algorithm. The features are *sex*, *age*, *pclass*, *fare* which are mixtures of continuous and discrete ones. We performed two types of experiments: VQCs with embedding both continuous (with rescaling) and discrete features, as well as VQCs with discrete features (after by ordinal representation of the continuous ones).

As for the mapping of the discrete features in both experiments, we implemented four distinct feature mappings as in the BC dataset, where the two rescaled continuous features in the first experiment were embedded with the ZZ feature map. The RyRz-

Table 1 Classified ratios of VQCs with various discrete-feature embeddings on parity functions.

3-bit parity (8 strings)		
Method	#qubits	Classified ratio
Naive	3	**1.000**
1×QRAC	1	0.500
2×QRAC	2	0.750
3×QRAC	3	**1.000**
TE	1	0.913
6-bit parity (64 strings)		
Naive	6	0.671
1×QRAC	2	0.563
2×QRAC	4	0.531
3×QRAC	6	0.547
TE	2	**0.934**
9-bit parity (512 strings)		
Naive	9	0.570
1×QRAC	3	0.506
2×QRAC	6	0.520
3×QRAC	9	0.529
TE	3	**0.674**

Table 2 Accuracies and F1 scores of 4-fold cross validated VQCs on the BC dataset, with four different discrete feature mappings. All mappings require four qubits.

Method	Train acc./F1	Test acc./F1
ZZ	0.606/0.475	0.549/0.419
QRAC	0.707/**0.541**	0.682/0.483
TE	**0.722**/0.534	0.681/0.452
Reg.TE	0.714/0.534	**0.702/0.511**

https://qiskit.org/documentation/stubs/qiskit.aqua.components.variational_forms.RYRZ.html

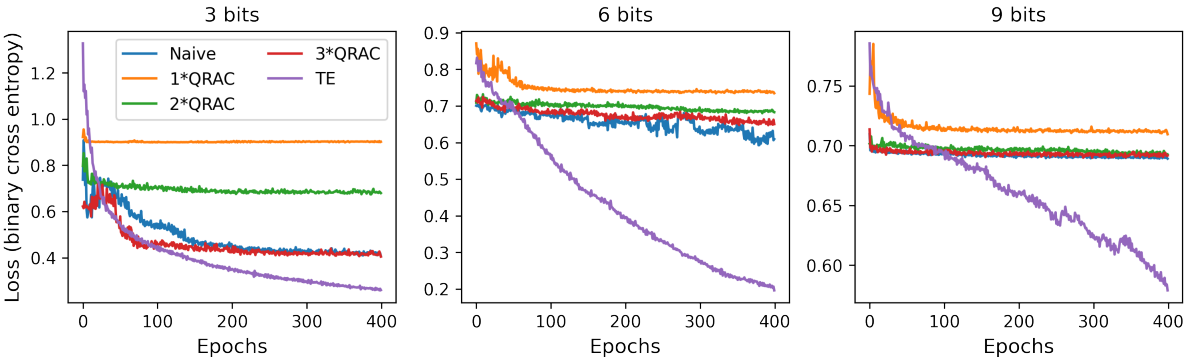

Fig. 3 Plots of training losses (binary cross entropy) for learning parity of 3, 6, and 9 bits for each embedding. Note that the number of epoch in the x-axis is doubled because each epoch of SPSA consists of two stages.

Table 3 Accuracies and F1 scores of 4-fold cross-validated VQCs on the TS dataset embedding two ordinal features and two rescaled continuous features (top four rows), and those embedding ordinal-encoded features (bottom four rows).For both experiments, the top ZZ methods used 4 qubits, while QRACs and TEs used 3 qubits.

Method	Train acc./F1	Test acc./F1
ZZ(dis.+cont.)	0.729/0.636	0.706/0.610
QRAC(dis.)+ZZ(cont.)	**0.777**/0.699	**0.767**/0.687
TE(dis.)+ZZ(cont.)	0.749/0.711	0.739/0.706
Reg.TE(dis.)+ZZ(cont.)	0.764/**0.723**	0.756/**0.713**
ZZ(dis.)	0.740/0.643	0.711/0.601
QRAC(dis.)	0.785/**0.721**	0.772/**0.707**
TE(dis.)	0.771/0.704	0.761/0.692
Reg.TE(dis.)	**0.788**/0.714	**0.776**/0.702

variational form of the VQCs were of depth 4 and were optimized using SPSA for 300 epochs in all experiments.

Table 3 shows the results of the two experiments. The results indicate that encoding continuous features as ordinal values slightly improves the overall performances of the models. However, F1 scores for the mixed TE(dis.) + ZZ(cont.) models were higher than the discrete-only TE(dis.) counterparts. The results also confirm that QRAC and TE were better than the baseline ZZ feature maps, and TE with regularization fairly to be on a par with QRAC.

4.4 QNN on hand-written digit MNIST

MNIST is one of the most well-known datasets that consists of black-white hand-written digit images of size 28×28. Such instances of images can be fed easily into classical machine learning. However, due to the limited number of qubits of near-term quantum devices, the size of those images must be reduced by some preprocessing in order to be usable in quantum machine learning. We followed the preprocessing shown at the tutorial of TensorFlow Quantum[*5], which is similar to the preprocessing in [34] . Only images of digits 3 and 6 were selected while those labelled both 3 and 6 were omitted. The selected images were resized down to 4×4, and their pixels were rounded to 0s and 1s to obtain binary string representations. The classification task is binary decision (i.e., either 3 or 6). We compared various embeddings of the images as the following.

- **Naive 16px**: a qubit for a pixel/bit of the 4×4 pixels. This was also used in [34].
- **Naive 8px**: a qubit for a pixel of those at the second and third rows of the 4×4 pixels. The purpose of this method is to observe the performance loss/gain if we reduce the number of qubits by ignoring parts of the inputs.
- **QRAC/TE**: the flatten 4×4 pixels, and thus 16 bits were embedded with $(3, 1)$-QRAC or $(3, 1)$-TE. Thus, in total 6 qubits were used.
- **Conv QRAC/TE**: convolutional embedding of the 4×4 pixels. C-QRAC and C-TE were used for every 3 consecutive pixels of each row and column with stride 1. Formally, suppose that $\boldsymbol{b}$ is the 4×4 input image. Every 3 pixels at i-th row: $b_{ij}b_{i(j+1)}b_{i(j+2)}$ for $1 \leq i \leq 4$ and $1 \leq j \leq 2$, and 3 pixels at j-th column: $b_{ij}b_{(i+1)j}b_{(i+2)j}$ for $1 \leq i \leq 2$ and $1 \leq j \leq 4$ were encoded with $(3, 1)$-QRAC or $(3, 1)$-TE, resp. Therefore, as in *Naive 16px*, in total 16 qubits were used.
- **Conv (4,1)-TE**: convolutional embedding on the 4×4 image with kernel size 2×2 and stride 1. Each 2×2 block was embedded using $(4, 1)$-TE. Formally, suppose that $\boldsymbol{b}$ is the 4×4 input image, we embedded every 2×2 pixels $b_{ij}b_{i(j+1)}b_{(i+1)(j)}b_{(i+1)(j+1)}$ for $1 \leq i, j \leq 3$ with $(4, 1)$-TE. In total, 9 qubits were used for the embedding. Note that $(4, 1)$-QRAC is impossible [44].

For every QNN with a particular embedding, we employed the same structure of circuit $\mathbf{W}(\boldsymbol{\theta})$ that consisted of mainly two layers, the first layer consisted of Ising XX gates controlling from every input qubits toward the output qubit. The second layer consisted of Ising ZZ gates controlling from every input qubits as the first layer[*6]. The binary cross entropy was used for training

Table 4 Experimental results of QNNs on MNIST dataset. Training and testing accuracies of various discrete feature embeddings with fewer qubits than Naive embeddings of [34] are shown.

Method	#qubits	Train acc.	Test acc.
Naive 8px	8	0.872	0.891
QRAC	6	0.883	0.898
TE	6	**0.903**	**0.912**
Naive 16px	16	0.902	0.908
Conv QRAC	16	0.896	0.898
Conv TE	16	0.911	0.914
Conv (4, 1)-TE	9	**0.911**	**0.917**

*5 As detailed in `https://www.tensorflow.org/quantum/tutorials/mnist`

*6 The Ising XX and ZZ gates are also used in `https://www.tensorflow.org/quantum/tutorials/mnist`

loss. We performed 5 different runs for each embedding method using Adam [45] as an optimizer for 10 epochs and reported their average accuracy. The experimental results are as in the Table 4.

As can be seen from the table, by using QRAC or TE we can embed the images with fewer qubits. Moreover, the accuracies of the embedding were better compared to the Naive method with 8 qubits, and were comparable to that with 16 qubits. In particular, we can see that Convolutional TEs performed better than others, even for Convolutonal $(4, 1)$-TE with only 9 qubits. The drawback of the Convolutional TE is the significant increase of the parameters for the embeddings, but not all combination of 4-bit binary strings appear; hence, some parameters can be ignored.

5. Conclusion

We propose trainable discrete feature embeddings by combining QRAC with the quantum metric learning. Our trainable embeddings show a remarkable advantage on the parity over embedding with QRACs [13] which needs additional qubits to overcome the linear separability limitation due to their fixed encoding. Furthermore, trainable embeddings also provide good classification performance with fewer qubits on various real-world datasets as demonstrated numerically through the framework of VQCs and QNNs. Another advantage of trainable embeddings is that it is not limited by the existence of QRACs, i.e., we show that $(4, 1)$-TE is possible to train and be used for classification achieving better or comparable accuracies with fewer qubits despite the non-existence of $(4, 1)$-QRAC [44]. For this reason, we believe our trainable embedding is flexible and can provide satisfying results in various tasks of quantum machine learning.

Lastly, there are clearly many directions for future work, including to develop trainable embedding with more qubits, to find good heuristics for the initialization as it heavily affects the final result of embeddings, and to design generalized (m, n)-TEs with regularization that provides theoretical guarantees.

Acknowledgments We would like to thank Prof. Hiroshi Imai of the University of Tokyo for his support in the lectures of Introduction to Quantum Computing from April to July 2020 during which parts of this paper were conceived. RR would also like to thank the TAs of the lectures, Hyungseok Chang and Atsuya Hasegawa, for their help, and Yutaka Shikano of Keio University for valuable feedback and suggestion. We are grateful to Hiroshi Yano of Keio University for providing assistance on the geometry of QRACs. Finally, we greatly appreciate the group members of the final project of the lectures, Junfeng Jiang, Yushan Zhang, Xujie Wang, and Zhihong Wu, who helped preliminary experiments in this paper.

References

[1] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. and Lloyd, S.: Quantum machine learning, *Nature*, Vol. 549, No. 7671, p. 195–202 (online), DOI: 10.1038/nature23474 (2017).

[2] Harrow, A. W., Hassidim, A. and Lloyd, S.: Quantum Algorithm for Linear Systems of Equations, *Phys. Rev. Lett.*, Vol. 103, p. 150502 (online), DOI: 10.1103/PhysRevLett.103.150502 (2009).

[3] Lloyd, S., Mohseni, M. and Rebentrost, P.: Quantum principal component analysis, *Nature Physics*, Vol. 10, No. 9, p. 631–633 (online), DOI: 10.1038/nphys3029 (2014).

[4] Preskill, J.: Quantum Computing in the NISQ era and beyond, *Quantum*, Vol. 2, p. 79 (online), DOI: 10.22331/q-2018-08-06-79 (2018).

[5] Jurcevic, P., Javadi-Abhari, A., Bishop, L. S., Lauer, I., Bogorin, D. F., Brink, M., Capelluto, L., Günlük, O., Itoko, T., Kanazawa, N., Kandala, A., Keefe, G. A., Krsulich, K., Landers, W., Lewandowski, E. P., McClure, D. T., Nannicini, G., Narasgond, A., Nayfeh, H. M., Pritchett, E., Rothwell, M. B., Srinivasan, S., Sundaresan, N., Wang, C., Wei, K. X., Wood, C. J., Yau, J.-B., Zhang, E. J., Dial, O. E., Chow, J. M. and Gambetta, J. M.: Demonstration of quantum volume 64 on a superconducting quantum computing system (2020).

[6] McClean, J. R., Romero, J., Babbush, R. and Aspuru-Guzik, A.: The theory of variational hybrid quantum-classical algorithms, *New Journal of Physics*, Vol. 18, No. 2, p. 023023 (2016).

[7] Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J. M. and Gambetta, J. M.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature*, Vol. 549, No. 7671, p. 242 (2017).

[8] Havlícek, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M. and Gambetta, J. M.: Supervised learning with quantum-enhanced feature spaces, *Nature*, Vol. 567, pp. 209–212 (2019).

[9] Goldberg, Y. and Elhadad, M.: splitSVM: Fast, Space-Efficient, non-Heuristic, Polynomial Kernel Computation for NLP Applications, *Proceedings of ACL-08: HLT, Short Papers*, Columbus, Ohio, Association for Computational Linguistics, pp. 237–240 (online), available from ⟨https://www.aclweb.org/anthology/P08-2060⟩ (2008).

[10] Schuld, M., Bocharov, A., Svore, K. and Wiebe, N.: Circuit-centric quantum classifiers (2018).

[11] Schuld, M. and Killoran, N.: Quantum Machine Learning in Feature Hilbert Spaces, *Physical Review Letters*, Vol. 122, No. 4 (online), DOI: 10.1103/physrevlett.122.040504 (2019).

[12] Mitarai, K., Negoro, M., Kitagawa, M. and Fujii, K.: Quantum circuit learning, *Physical Review A*, Vol. 98, No. 3 (online), DOI: 10.1103/physreva.98.032309 (2018).

[13] Yano, H., Suzuki, Y., Raymond, R. and Yamamoto, N.: Efficient Discrete Feature Encoding for Variational Quantum Classifier, *IEEE International Conference on Quantum Computing and Engineering (QCE)*, p. to appear (2020).

[14] Ambainis, A., Nayak, A., Ta-Shma, A. and Vazirani, U.: Dense Quantum Coding and Quantum Finite Automata, *J. ACM*, Vol. 49, No. 4, pp. 496–511 (online), DOI: 10.1145/581771.581773 (2002).

[15] Holevo, A. S.: Bounds for the quantity of information transmitted by a quantum communication channel, *Problems of Information Transmission*, Vol. 9, No. 3, pp. 177–183 (1973).

[16] Hayashi, M., Iwama, K., Nishimura, H., Raymond, R. and Yamashita, S.: Quantum Network Coding, *STACS 2007* (Thomas, W. and Weil, P., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 610–621 (2007).

[17] Quadeer, M., Banik, M., Ambainis, A. and Rai, A.: Strong supremacy of quantum systems as communication resource (2017).

[18] Iwama, K., Nishimura, H., Raymond, R. and Yamashita, S.: Unbounded-Error One-Way Classical and Quantum Communication Complexity, *Automata, Languages and Programming*, Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 110–121 (2007).

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

[20] Dua, D. and Graff, C.: UCI Machine Learning Repository (2017).

[21] Frank E. Harrell Jr., T. C.: Titanic dataset (2017).

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

[23] Breiman, L., Friedman, J., Stone, C. J. and Olshen, R. A.: *Classification and Regression Trees*, CRC press (1984).

[24] Hancock, J. T. and Khoshgoftaar, T. M.: Survey on categorical data for neural networks, *Journal of Big Data*, Vol. 7, No. 1, p. 28 (online), DOI: 10.1186/s40537-020-00305-w (2020).

[25] Guo, C. and Berkhahn, F.: Entity Embeddings of Categorical Variables, *CoRR*, Vol. abs/1604.06737 (online), available from ⟨http://arxiv.org/abs/1604.06737⟩ (2016).

[26] Hinton, G. E. et al.: Learning distributed representations of concepts, *Proceedings of the eighth annual conference of the cognitive science society*, Vol. 1, Amherst, MA, p. 12 (1986).

[27] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality, *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, Red Hook, NY, USA, Curran Associates Inc., p. 3111–3119 (2013).

[28] Chopra, S., Hadsell, R. and LeCun, Y.: Learning a Similarity Metric Discriminatively, with Application to Face Verification, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, USA, IEEE Computer Society, p. 539–546 (online), DOI: 10.1109/CVPR.2005.202 (2005).

ⓒ 2020 Information Processing Society of Japan

[29] Wiesner, S.: Conjugate Coding, *SIGACT News*, Vol. 15, No. 1, pp. 78–88 (online), DOI: 10.1145/1008908.1008920 (1983).

[30] Aaronson, S.: The learnability of quantum states, *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 463, No. 2088, pp. 3089–3114 (online), DOI: 10.1098/rspa.2007.0113 (2007).

[31] Chailloux, A., Kerenidis, I., Kundu, S. and Sikora, J.: Optimal bounds for parity-oblivious random access codes, *New Journal of Physics*, Vol. 18, No. 4, p. 045003 (online), available from ⟨http://stacks.iop.org/1367-2630/18/i=4/a=045003⟩ (2016).

[32] Liabøtrø, O.: Improved classical and quantum random access codes, *Phys. Rev. A*, Vol. 95, p. 052315 (online), DOI: 10.1103/PhysRevA.95.052315 (2017).

[33] Imamichi, T. and Raymond, R.: Constructions of Quantum Random Access Codes, *Asian Quantum Information Symposium (AQIS)* (2018).

[34] Farhi, E. and Neven, H.: Classification with Quantum Neural Networks on Near Term Processors (2018).

[35] Schuld, M., Sinayskiy, I. and Petruccione, F.: The quest for a Quantum Neural Network, *Quantum Information Processing*, Vol. 13, pp. 2567–2586 (2014).

[36] Ambainis, A., Leung, D., Mancinska, L. and Ozols, M.: Quantum random access codes with shared randomness, *arXiv preprint arXiv:0810.2937* (2008).

[37] Nielsen, M. A. and Chuang, I. L.: *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK (2000).

[38] Cortes, C. and Vapnik, V.: Support-Vector Networks, *Machine Learning*, pp. 273–297 (1995).

[39] Abraham, H. and et al.: Qiskit: An Open-source Framework for Quantum Computing (2019).

[40] Osogami, T. and Raymond, R.: Determinantal Reinforcement Learning, *AAAI*, pp. 4659–4666 (online), available from ⟨https://doi.org/10.1609/aaai.v33i01.33014659⟩ (2019).

[41] Broughton, M., Verdon, G., McCourt, T., Martínez, A. J., Yoo, J. H., Isakov, S. V., Massey, P., Niu, M. Y., Halavati, R., Peters, E., Leib, M., Skolik, A., Streif, M., Dollen, D. V., McClean, J., Boixo, S., Bacon, D., Ho, A., Neven, H. and Mohseni, M.: TensorFlow Quantum: A Software Framework for Quantum Machine Learning, *ArXiv*, Vol. abs/2003.02989 (2020).

[42] Spall, J. C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Transactions on Automatic Control*, Vol. 37, No. 3, pp. 332–341 (1992).

[43] Breiman, L.: Random Forests, *Mach. Learn.*, Vol. 45, No. 1, p. 5–32 (online), DOI: 10.1023/A:1010933404324 (2001).

[44] Hayashi, M., Iwama, K., Nishimura, H., Raymond, R. and Yamashita, S.: (4,1)-Quantum random access coding does not exist – one qubit is not enough to recover one of four bits, *New Journal of Physics*, Vol. 8, No. 8, p. 129 (online), available from ⟨http://stacks.iop.org/1367-2630/8/i=8/a=129⟩ (2006).

[45] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *CoRR*, Vol. abs/1412.6980 (2015).

ⓒ 2020 Information Processing Society of Japan