PAPER • OPEN ACCESS

A Grover-search based quantum learning scheme for classification

To cite this article: Yuxuan Du *et al* 2021 *New J. Phys.* **23** 023020

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
21 October 2020

REVISED
21 January 2021

ACCEPTED FOR PUBLICATION
22 January 2021

PUBLISHED
17 February 2021

PAPER

A Grover-search based quantum learning scheme for classification

Yuxuan Du[1], **Min-Hsiu Hsieh**[2], **Tongliang Liu**[1] and **Dacheng Tao**[1,*]

[1] UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering, The University of Sydney, Darlington NSW 2008, Australia

[2] Centre for Quantum Software and Information, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia

[*] Author to whom any correspondence should be addressed.

E-mail: dacheng.tao@sydney.edu.au

Keywords: quantum machine learning, quantum computation, quantum classification

Abstract

The hybrid quantum–classical learning scheme provides a prominent way to achieve quantum advantages on near-term quantum devices. A concrete example toward this goal is the quantum neural network (QNN), which has been developed to accomplish various supervised learning tasks such as classification and regression. However, there are two central issues that remain obscure when QNN is exploited to accomplish classification tasks. First, a quantum classifier that can well balance the computational cost such as the number of measurements and the learning performance is unexplored. Second, it is unclear whether quantum classifiers can be applied to solve certain problems that outperform their classical counterparts. Here we devise a Grover-search based quantum learning scheme (GBLS) to address the above two issues. Notably, most existing QNN-based quantum classifiers can be seamlessly embedded into the proposed scheme. The key insight behind our proposal is reformulating the classification tasks as the search problem. Numerical simulations exhibit that GBLS can achieve comparable performance with other quantum classifiers under various noise settings, while the required number of measurements is dramatically reduced. We further demonstrate a potential quantum advantage of GBLS over classical classifiers in the measure of query complexity. Our work provides guidance to develop advanced quantum classifiers on near-term quantum devices and opens up an avenue to explore potential quantum advantages in various classification tasks.

1. Introduction

The field of machine learning has achieved remarkable success in computer vision, natural language processing, and data mining [1]. Recently, an increasing interest from the physics community to use machine learning methods to solve complicated physics problems, e.g. classifying phases of matter and simulating quantum systems [2–4], has emerged. Besides the revolutionary influence of machine learning to the physics world, another uprising field that tightly binds machine learning with physics is quantum machine learning whose goal is to solve specific tasks beyond the reach of classical computers [5].

To better understand how quantum computing facilitates the machine learning tasks, devising quantum algorithms that have the ability to solve fundamental machine learning problems with quantum advantages is desirable [5]. For example, the proposed quantum linear systems algorithm (a.k.a., HHL algorithm) enables the linear equations to be solved with the exponential speedup over its classical counterparts [6]. By employing HHL algorithm as the subroutine, many quantum machine learning algorithms with exponential quantum speedup have been proposed, e.g. the quantum principal component analysis [7], quantum singular value decomposition [8], quantum non-negative matrix factorization [9], and the quantum regression [10]. However, those proposed quantum algorithms that possess fabulous quantum advantages

can only be executed on a fault-tolerant quantum computer by using the quantum random access memory [6], which is still a rather distant dream.

When approaching the noisy intermediate-scale quantum (NISQ) era, it is intrigued to explore whether there exists any quantum algorithm that can not only solve fundamental learning problems with promised quantum advantages but can also be efficiently implemented on near-term quantum devices [11]. To achieve this goal, one of the most likely solutions is the quantum neural network (QNN), which is also called as *variational quantum algorithms* [12–14]. Concretely, QNN is composed of a variational quantum circuit to prepare quantum states and a classical controller to perform optimization tasks [13, 15]. Partial evidence to support this claim is the theoretical result that the probability distribution generated by the variational quantum circuit used in QNN can not be efficiently simulated by classical computers [16–18]. Driven by the strong expressive power of quantum circuits and the similar work philosophy between QNN and the classical deep neural network (DNN), its natural to exploit whether QNN can be realized on near-term quantum computers to accomplish certain machine learning tasks with better performance over classical learning algorithms.

A central application of QNN, analogous to DNN, is tackling classification tasks [1]. Many real-world problems can be categorized into the classifying scenario, e.g. the recognization of hand-written digits, the characterization of different creatures, and the discrimination of quantum states. For binary classification, given a dataset

$$\hat{\mathcal{D}} = \left\{ \left(\boldsymbol{x}_i, y_i \right) \right\}_{i=0}^{N-1} \in (\mathbb{R}^{N\times M}, \{0,1\}^N), \tag{1}$$

with N examples and M features in each example, QNN aims to learn a decision rule $f_{\boldsymbol{\theta}}(\cdot)$ that correctly predicts the label of the given dataset $\hat{\mathcal{D}}$, i.e.

$$\min_{\boldsymbol{\theta}} \sum_{i=0}^{N-1} \mathbb{1}_{y_i \neq f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}, \tag{2}$$

where $\boldsymbol{\theta}$ refers to the trainable parameters and $\mathbb{1}_z$ is the indicator function that takes the value 1 if the condition z is satisfied and zero otherwise. Recently, QNNs with varied quantum circuit architectures and optimization methods have been proposed to accomplish the aforementioned classification tasks. In particular, the references [19–21] have devised the amplitude encoding based QNN to classify the Iris dataset and the hand-written digits image dataset; the references [22–24] have developed the kernel-based QNN to accomplish the synthetic datasets; and the references [25] have proposed the convolution based QNN to tackle quantum state discrimination tasks. When no confusion can arise, we use the *quantum classifier* in the rest of the study to specify QNNs that are used to accomplish classification tasks defined in equation (2).

Despite the promising heuristic results mentioned above, very few studies have theoretically explored the power of quantum classifiers. A noticeable theoretical result about quantum classifiers is the trade-off between the computational cost (i.e. the number of measurements) and the training performance indicated by [13]. Denote $\mathcal{L}(\boldsymbol{\theta}^{(t)}, \boldsymbol{z})$ as the loss function employed in quantum classifiers, where $\boldsymbol{\theta}^{(t)}$ refers to the trainable parameters at the tth iteration and $\boldsymbol{z} = \{\boldsymbol{z}_j\}_{j=1}^N$ is the given dataset with in total N samples. As shown in figure 1, when the *batch gradient descent* method is employed to optimize the loss function $\mathcal{L}$, the updating rule of the trainable parameters follows

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \frac{\eta}{B} \sum_{i=1}^{B} \nabla \mathcal{L}(\boldsymbol{\theta}^{(t)}, \mathcal{B}_i), \tag{3}$$

where η is the learning rate, $\mathcal{B}_i$ refers to the ith batch with $\cup_{i=1}^{B} \mathcal{B}_i = \boldsymbol{z}$ and $\mathcal{B}_i \cap \mathcal{B}_j = \emptyset$, and B denotes the number of batches. Define

$$R_1 = \mathbb{E}[\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2]. \tag{4}$$

as the utility measure that evaluates the distance between the optimized result and the stationary point in the optimization landscape. The following theorem summarizes the utility bound R_1 of quantum classifiers.

Theorem 1 (Modified from theorem 1 of [13]). *Quantum classifiers under the depolarization noise setting output $\boldsymbol{\theta}^{(T)} \in \mathbb{R}^d$ after T iterations with the utility bound*

$$R_1 \leqslant \tilde{O}\left(\text{poly}\left(\frac{d}{T(1-p)^{L_Q}}, \frac{d}{BM(1-p)^{L_Q}}, \frac{d}{(1-p)^{L_Q}} \right) \right),$$

where M is the number of measurements to estimate the quantum expectation value, L_Q is the circuit depth of variational quantum circuits, p is the rate of the depolarization noise, and B is the number of batches.

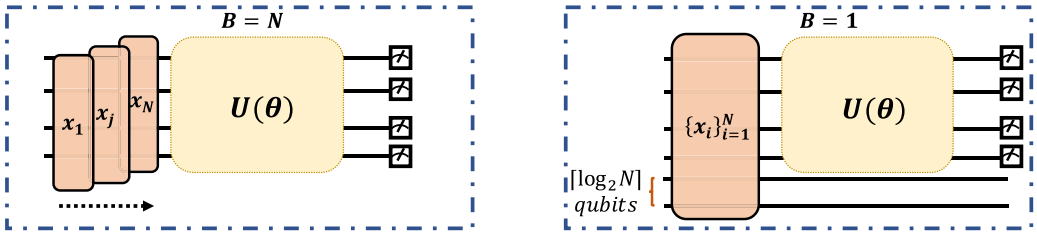

Figure 1. The protocol of the batch gradient descent method. The left panel corresponds to the setting as $B = N$, where the N training examples $\{\boldsymbol{x}_i\}_{i=1}^N$ are iteratively fed into the variational quantum circuits to acquire the gradients that estimate $\{\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)}, \boldsymbol{x}_i)\}_{i=1}^N$. The right panel exhibits the implementation of the quantum classifier when $B = 1$. Specifically, a superposition state $|\phi(\boldsymbol{x})\rangle = \frac{1}{\sqrt{N}}\sum_{i=1}^N |h(\boldsymbol{x}_i)\rangle_F|i\rangle_I$ is prepared, where $h(\cdot)$ corresponds to the employed encoding method and the subscripts 'I' and 'F' refer to the index and feature registers, respectively. Given access to $|\phi(\boldsymbol{x})\rangle$, the trainable quantum circuit $U_L(\boldsymbol{\theta})$ is employed to interact with its feature register subscripted with F.

Table 1. The basic information of different quantum classifiers. The notations T, K, M, N, and d refer to the number of epochs, the batch size (i.e. in our simulation $K = 4$), the number of measurements used to estimate quantum expectation value, the total number of training examples, and the total number of trainable parameters.

Methods	MSE_batch	MSE	BCE	GBLS
Number of batches B	$\frac{N}{K}$	N	N	$\frac{N}{K}$
Number of measurements	$O(\frac{TMNd}{K})$	$O(TMNd)$	$O(TMNd)$	$O(\frac{TMNd}{K})$

The result of theorem 1 indicates that a larger number of batches B ensures a better utility bound R_1, while the price to pay is increasing the total number of measurements. For example, when $B = N$, we have $\mathcal{B}_i = \boldsymbol{z}_i$ for $\forall i \in [N]$ and each sample $\boldsymbol{z}_j$ is sequentially fed into variational quantum circuits to acquire $\nabla\bar{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{z}_i)$ that estimates $\nabla\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}_i)$. Once the set $\{\nabla\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}_i)\}_{i=1}^N$ is collected, the gradients $\nabla\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z})$ can be estimated by $\frac{1}{N}\sum_{i=1}^N \nabla\bar{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{z}_i)$. Suppose that the required number of measurements to estimate the derivative of the jth parameter $\boldsymbol{\theta}_j$, i.e. $\nabla_j\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}_i) = \frac{\partial\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}_i)}{\partial\boldsymbol{\theta}_j}$, is M, then the total number of measurements to acquire $\frac{1}{N}\sum_{i=1}^N \nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{z}_i)$ is NM. Therefore, the estimation of $\nabla\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z})$, which includes d parameters, requires NMd measurements. Such a cost becomes unaffordable for large N. However, the trade-off between the utility R_1 and the computational efficiency caused by the varied number of batches B is not considered in previous quantum classifiers, where most of them only focused on the setting $B = N$. How to design a quantum classifier that can attain a good utility R_1 with a low computational cost is unknown.

Another theoretical issue toward quantum classifiers is that none of the previous results have explored their potential advantages compared with classical counterparts. This questions the necessity of employing quantum classifiers because no benefit can be offered. Under the above observations, it is highly desirable to develop a quantum classifier that can not only achieve a good utility R_1 using a low computational cost, but can also possess certain quantum advantages compared with classical classifiers.

Here we devise a Grover-search based learning scheme (GBLS) to address the above two issues under the NISQ setting. Our proposal has the following advantages. First, GBLS is a flexible and effective learning scheme, which enables the optimization of different quantum classifiers with a varied number of batches B. Note that the choice of the encoding methods and the variational ansatz used in GBLS is very flexible, which covers a wide range of the proposed quantum classifiers [20–24]. Moreover, the Grover-search based machinery is only required in the training process, and the prediction of the new input is completed by only using the optimized variational quantum circuits, which ensures its efficacy. Second, we prove that the query complexity can be quadratically reduced over its classical counterparts in the optimal setting (see theorem 2) when it is applied to accomplish specific binary classification tasks. Last, numerical simulation results demonstrate that GBLS can well accomplish binary classification tasks even when the system noise and the finite number of quantum measurements are considered (see section 3). Notably, the required number of measurements of GBLS is dramatically less than other advanced quantum classifiers [22–24] with competitive performance (see table 1). In other words, GBLS is a powerful protocol that allows quantum classifiers to achieve a good utility bound R_1 with a low computational cost.

The central concept in GBLS is reformulating the classification tasks as the search problem. Note that although the advantage held by the quantum Grover-search algorithm is evident, how to transform the classification task into the search problem is *inconclusive*. Such a reformulation is the main technical contribution in this study. Recall that Grover-search [26] identifies the target element i^* in a database of size

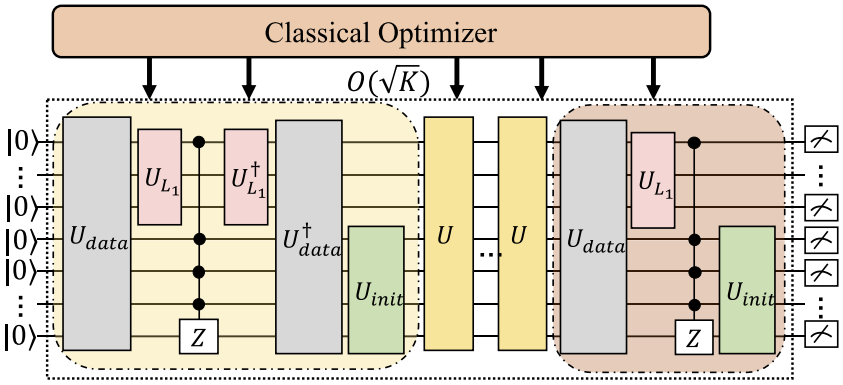

Figure 2. The paradigm of GBLS. U defined in equation (9) is composed of unitary operators (i.e. U_{data}, U_{L_1}, MCZ, and U_{init}) highlighted by the shadowed yellow region. The last cycle employs the unitary operation U_E defined in equation (10), highlighted by the brown region. The qubits interacted with U_{L_1} (or U_{init}) form the feature (or data) register $\mathcal{R}_F$ (or $\mathcal{R}_I$).

K by iteratively applying a predefined oracle $U_f = \mathbb{I} - 2\,|i^*\rangle\langle i^*|$ and a diffusion operator $U_{\text{init}} = 2\,|\varphi\rangle\langle\varphi| - \mathbb{I}$ with $|\varphi\rangle = \frac{1}{\sqrt{K}}\sum_i |i\rangle$ to the input state. GBLS, as shown in figure 2, employs a specified variational quantum circuit U_{L_1} and a multiple controlled qubits gate along the Z axis (MCZ) to replace the oracle U_f. In particular, the variational quantum circuit conditionally flips a flag qubit (i.e. the black dot behind U_{L_1} highlighted by the pink region) depending on the training data. The flag qubit is then employed as a part of MCZ gate to guide a Grover-like search algorithm to identify the index of the specified example, i.e. the status of the flag qubit such as '0' or '1' determines the successful probability to identify the target index. Through optimizing the trainable parameters of the variational quantum circuits U_{L_1}, GBLS aims to maximize the successful probability to sample the target index when the corresponding training example is positive; otherwise, GBLS minimizes the successful probability of sampling the target index. The inherited property from the Grover-search algorithm allows our proposal to achieve an advantage in terms of query complexity when the binary classification task involves the searching constraint (see section 2.3 for details). Besides the computational merit, GBLS is insensitive to noise, guaranteed by the fact that combining a variational learning approach with Grover-search can preserve a high probability of success in finding the solution under the NISQ setting [27].

2. Grover-search based learning scheme

The outline of this section is as follows. In subsection 2.1, we first elaborate on the implementation details of the proposed GLBS as depicted in figure 2. We then explain how to use the trained GLBS to predict the given new input with $O(1)$ query complexity in subsection 2.2. We last explain how GBLS can solve certain learning problems with potential advantages in subsection 2.3.

2.1. Implementation

In the preprocessing stage, GBLS employs the dataset $\hat{\mathcal{D}}$ defined in equation (1) to construct an *extended* dataset $\mathcal{D}$. Compared with the original dataset $\hat{\mathcal{D}}$, the cardinality of each training example in $\mathcal{D}$ is enlarged to K. For the purpose of applying the Grover-search algorithm to locate the target index $i^* = K-1$, the construction rule for the kth extended training example $\mathcal{D}_k$ for all $k \in [N]$ is as follows. The mathematical representation of $\mathcal{D}_k$ is

$$\mathcal{D}_k = [(\boldsymbol{x}_k^{(0)}, y_k^{(0)}), (\boldsymbol{x}_k^{(1)}, y_k^{(1)}, \ldots, (\boldsymbol{x}_k^{(K-1)}, y_k^{(K-1)})]. \tag{5}$$

The last pair in $\mathcal{D}_k$ corresponds to the kth example of $\hat{\mathcal{D}}$, i.e. $(\boldsymbol{x}_k^{(K-1)}, y_k^{(K-1)}) = (\boldsymbol{x}_k, y_k)$. The first $K-1$ pairs $\{(\boldsymbol{x}_k^{(i)}, y_k^{(i)})\}_{i=0}^{K-2}$ in $\mathcal{D}_k$ are uniformly sampled from a subset of $\hat{\mathcal{D}}$, where all labels of this subset, i.e. $\{y_k^{(i)}\}_{i=1}^{K-2}$, are opposite to y_k. Note that the construction of the subset is efficient. Since $y_k \in \{0,1\}$, we can construct two subsets $\hat{\mathcal{D}}^{(0)}$ and $\hat{\mathcal{D}}^{(1)}$ that only contains examples of $\hat{D}$ with label '0' and label '1', respectively, where $\hat{\mathcal{D}}^{(0)} \cup \hat{\mathcal{D}}^{(1)} = \hat{\mathcal{D}}$. When $y_k = 0$, the first $K-1$ pairs are sampled from $\hat{\mathcal{D}}^{(1)}$; otherwise, when $y_k = 1$, the first $K-1$ pairs are sampled from $\hat{\mathcal{D}}^{(0)}$.

As aforementioned, different quantum classifiers exploit different methods to encode $\mathcal{D}_k$ into the quantum states [12]. For ease of notation, we denote the quantum state corresponding to the kth example

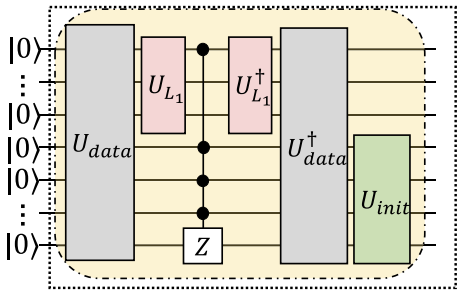

Figure 3. The circuit implementation of the oracle U in equation (9).

$\mathcal{D}_k$ as

$$U_{\text{data}}|\mathbf{0}\rangle := \left|\Phi^k\right\rangle_{F,I} = \frac{1}{\sqrt{K}}\sum_{i=0}^{K-1}|h(\boldsymbol{x}_i)\rangle_F|i\rangle_I, \tag{6}$$

where $h(\cdot)$ is an encoding operation (a possible encoding method is discussed in section 3), and the subscripts 'F' and 'I' refer to the feature register $\mathcal{R}_F$ with N_F qubits and the index register $\mathcal{R}_I$ with N_I qubits, respectively.

We now move on to explain the training procedure of GBLS. Recall that the reference [27] points out that combining a variational learning approach with Grover-search algorithm produces an additional quantum advantage than conventional Grover's algorithm such that the target solution can be located with a higher success probability. A similar idea is used in GBLS. Namely, the employed variational quantum circuits U_{L_1} aim to learn a hyperplane that separates the last pair in $\mathcal{D}_k$ with its first $K-1$ pairs. Denote $U_{L_1} = \prod_{l=1}^{L} U(\boldsymbol{\theta}^l)$, where each layer $U(\boldsymbol{\theta}^l)$ contains $O(\text{poly}(N_F))$ parameterized single qubit gates and at most $O(\text{poly}(N_F))$ fixed two-qubit gates with the identical layouts. In the *optimal* situation, given the initial state $\left|\Phi^k\right\rangle_{F,I}$ in equation (6), applying $U_{L_1} = \prod_{l=1}^{L} U(\boldsymbol{\theta}^l)$ to the feature register $\mathcal{R}_F$ yields the following target state:

(a) If the *last pair* of the input example $\mathcal{D}_k$ refers to the label $y_k = 0$, the target state is

$$(U_{L_1}\otimes\mathbb{I})\left|\Phi^k(y_k=0)\right\rangle_{F,I} = \frac{1}{\sqrt{K}}\sum_{i=0}^{K-1}\left|\psi_i^{(0)}\right\rangle_F|i\rangle_I; \tag{7}$$

(b) Otherwise, when the *last pair* of the input example $\mathcal{D}_k$ refers to $y_k = 1$, the target state is

$$(U_{L_1}\otimes\mathbb{I})\left|\Phi^k(y_k=1)\right\rangle_{F,I} = \frac{1}{\sqrt{K}}\sum_{i=0}^{K-1}\left|\psi_i^{(1)}\right\rangle_F|i\rangle_I. \tag{8}$$

We denote $\left|\psi_i^{(0)}\right\rangle_F$ (resp. $\left|\psi_i^{(1)}\right\rangle_F$) as the first qubit of the quantum state in the feature register $\mathcal{R}_{\mathcal{F}}$ being $|0\rangle$ (resp. $|1\rangle$). As shown in figure 3, once the state $(U_{L_1}\otimes\mathbb{I}_I)\left|\Phi^k\right\rangle_{F,I}$ is prepared, GBLS iteratively applies MCZ gate to the index register controlled by the first qubit of the feature register and the index register, uses U_{data} and U_{L_1} to uncompute the feature register, and applies the diffusion operator U_{init} to the index register to complete the first cycle. Denote all quantum operations belong to one cycle as U, i.e.

$$U := U_{\text{init}}\circ U_{\text{data}}^{\dagger}\circ(U_{L_1}\otimes\mathbb{I})^{\dagger}\circ\text{MCZ}\circ(U_{L_1}\otimes\mathbb{I})\circ U_{\text{data}}. \tag{9}$$

With a slight abuse of notation, we define $U_{\text{init}} = \mathbb{I}_F\otimes(2|\varphi\rangle\langle\varphi| - \mathbb{I}_I)$ with $|\varphi\rangle = \frac{1}{\sqrt{K}}\sum_i|i\rangle$ in the rest of the paper. GBLS repeatedly applies U to the initial state $|\mathbf{0}\rangle$ except for the last cycle, where the applied unitary operations are replaced by

$$U_E := U_{\text{init}}\circ\text{MCZ}\circ(U_{L_1}\otimes\mathbb{I})\circ U_{\text{data}}, \tag{10}$$

as highlighted by the brown shadow in figure 4. Following the conventional Grover-search, GBLS queries U and U_E with in total $O(\sqrt{K})$ times before taking quantum measurements. This completes the quantum part of GBLS.

We next analyze how the quantum state evolves for the case $y_k = 0$ and $y_k = 1$, respectively. For the case of $y_k = 0$, applying $U_{L_1}\otimes\mathbb{I}_I$ to the input state $\left|\Phi^k(y_k=0)\right\rangle_{F,I}$ in equation (6) will transform this state to $1/\sqrt{K}\sum_{i=0}\left|\psi_i^{(0)}\right\rangle_F|i\rangle_I$ as described in equation (7). Since the control qubit in the feature register is 0,

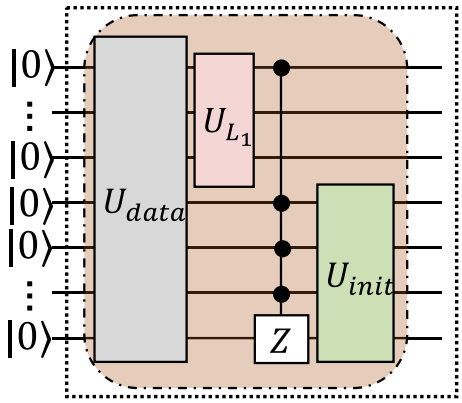

Figure 4. The circuit implementation of the oracle U_E in equation (10).

applying MCZ gate does not flip the phase of the state. After uncomputing, the result state yields $1/\sqrt{K}\sum_{i=0}|\mathbf{0}\rangle_F|i\rangle_I$. The positive phase for all computational basis $i \in [K-1]$ implies that applying the quantum operation $U_{\text{init}} \circ U_{\text{data}}^{\dagger} \circ (U_{L_1} \otimes \mathbb{I}_I)^{\dagger}$ does not change the state as well, i.e.

$$\left(\mathbb{I}_F \otimes \left(2\left|\varphi\right\rangle\left\langle\varphi\right| - \mathbb{I}_I\right)\right) \frac{1}{\sqrt{K}} \sum_{i=0}^{K-1} |\mathbf{0}\rangle_F |i\rangle_I = \frac{1}{\sqrt{K}} \sum_{i=0}^{K-1} |\mathbf{0}\rangle_F |i\rangle_I. \tag{11}$$

In other words, when we measure the index register of the output state, the probability to sample the computation basis i with $i \in [K-1]$ is uniformly distributed.

For the case of $y_k = 1$, the input state $\left|\Phi^k(y_k = 1)\right\rangle_{F,I}$ in equation (6) will be transformed to $1/\sqrt{K}\sum_{i=0}\left|\psi_i^{(1)}\right\rangle_F |i\rangle_I$ after interacting with unitary $U_{L_1} \otimes \mathbb{I}_I$, as described in equation (8). With the control qubit in the feature register being 1, such a generated quantum state will evolve as Grover-search algorithm does by iteratively applying MCZ, the uncomputation operation $U_{\text{data}}^{\dagger} \circ (U_{L_1} \otimes \mathbb{I})^{\dagger}$, and U_{init}. Mathematically, the result state after interacting with MCZ yields

$$\hat{U}_f \left|\Phi^k(y_k = 1)\right\rangle_{F,I} = \cos\,\gamma \left|\psi_B^{(0)}\right\rangle_F |B\rangle_I - \sin\,\gamma \left|\psi_{i^*}^{(1)}\right\rangle_F |i^*\rangle_I, \tag{12}$$

where $\hat{U}_f := \text{MCZ} \circ (U_{L_1} \otimes \mathbb{I})$, $\cos\,\gamma = \frac{\sqrt{K-1}}{\sqrt{K}}$, $|B\rangle_I = \frac{1}{\sqrt{K-1}}\sum_{i=0}^{K-2}|i\rangle_I$, and $|i^*\rangle_I$ refers to the computational basis $|K-1\rangle$. Analogous to the U_f in Grover-search, the trainable and data-driven $\hat{U}_f$ used above conditionally flips the phase of the state $|i^*\rangle$. Next, the uncomputing operation $U_{\text{data}}^{\dagger} \circ (U_{L_1} \otimes \mathbb{I})^{\dagger}$ and the diffusion operator U_{init} are employed to increase the probability of $|i^*\rangle_I$. Mathematically, the generated state after the first cycle yields

$$U\left|\Phi^k(y_k = 1)\right\rangle_{F,I} = \cos\,3\gamma |\mathbf{0}\rangle_F |B\rangle_I + \sin\,3\gamma |\mathbf{0}\rangle_F |i^*\rangle_I, \tag{13}$$

where U is defined in equation (9). The probability of sampling i^* is increased to $\sin^2 3\gamma$, which is in accordance to Grover-search algorithm. This observation leads to the following theorem, whose proof is given in appendix A.

Theorem 2. *For GBLS, under the optimal setting, the probability of sampling the outcome $i^* = K-1$ approaches 1 asymptotically iff the label of the last entry of $\mathcal{D}_k$ is $y_k = 1$.*

We leverage the particular property of GBLS, in which the output distribution is varied for different label of input $\mathcal{D}_k$ as shown in theorem 2, to accomplish the binary classification task. Concisely, the output state of GBLS, i.e. $U_E U^{O(\sqrt{K})}|\mathbf{0}\rangle_{F,I}$, corresponding to $y_k = 1$ will contain the computational basis $i = K-1$ with probability near to 1. By contrast, the output state corresponding to $y_k = 0$ will contain all computational bases $i \in [K-1]$ with the equal probability. Driven by this observation and the mechanism of the Grover-search algorithm, the loss function of GBLS is

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) := \text{sign}(1/2 - y_k)\,\text{Tr}(\Pi\rho(\boldsymbol{\theta})), \tag{14}$$

where $\text{sign}(\cdot)$ is the sign function, $\Pi = (|1\rangle\langle 1|) \otimes \mathbb{I} \otimes (|i^*\rangle\langle i^*|)$ refers to the measurement operator, $\rho(\boldsymbol{\theta}) = U_E U(\boldsymbol{\theta})^{O(\sqrt{K})} |\mathbf{0}\rangle\langle\mathbf{0}| (U_E U(\boldsymbol{\theta})^{O(\sqrt{K})})^{\dagger}$ is the generated quantum state, and $U(\boldsymbol{\theta})$ is defined in equation (9) (for clearness, we use the explicit form $U(\boldsymbol{\theta})$ instead of U). Intuitively, the minimized $\mathcal{L}(\boldsymbol{\theta})$

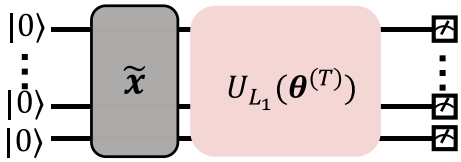

Figure 5. The circuit implementation of GBLS for prediction. The same encoding method used in the training process is adopted to prepare the state $|h(\tilde{\boldsymbol{x}})\rangle$. The trained variational quantum circuit $U(\boldsymbol{\theta}^{(T)})$ is applied to $|h(\tilde{\boldsymbol{x}})\rangle$ before the measurement.

corresponds to the facts that when $y_k = 1$ ($y_k = 0$), the success probability to sample i^* as well as attain the first feature qubit to be '1' ('0') is maximized (minimized). GBLS employs a gradient-based method, i.e. the parameter shift rule [22], to optimize $\boldsymbol{\theta}$. Confer appendix B for the detail.

We would like to address that, GBLS can be used to conduct both the linear and nonlinear classification tasks depending on the specified quantum classifiers. For example, when GBLS adopts the proposal [23, 24] to implement U_{data} and U_{L_1}, it has capability of classifying nonlinear data.

2.2. Prediction

Once the training of GBLS has finished, the trained U_{L_1} can be directly employed to predict the label of the future instances with $O(1)$ query complexity, where the corresponding circuit implementation is shown in figure 5. To achieve this, we devise the following prediction method. Denote the new input as $(\tilde{\boldsymbol{x}}, \tilde{y})$. We first encode $\tilde{\boldsymbol{x}}$ into the quantum state with the identical encoding method used in the training procedure, i.e. $\left|\tilde{\psi}\right\rangle_F = |h(\tilde{\boldsymbol{x}})\rangle$. Applying the trained U_{L_1} to $\left|\tilde{\psi}\right\rangle_F$ yields

$$U_{L_1}|\psi\rangle_F = \tilde{\alpha}\left|\tilde{\psi}^{(0)}\right\rangle_F + \tilde{\beta}\left|\tilde{\psi}^{(1)}\right\rangle_F, \tag{15}$$

where $|\tilde{\alpha}|^2 + |\tilde{\beta}|^2 = 1$.

Denote the probability of the outcome '1' after measuring the first feature qubit of the state in equation (15) as $p_1 = |\tilde{\beta}|^2$ and let the threshold be $1/2$. The new input data $\tilde{\boldsymbol{x}}$ will be identified as label '0', if $p_1 < 1/2$; otherwise, it will be given label '1'.

2.3. Potential advantage of GBLS

Here we design a binary classification task to explore the potential advantage of GBLS in terms of query complexity. Consider the classification task that requires not only to find a decision rule in equation (2) but also to output the index j satisfying a pre-determined black-box function. Note that the identification of a target index is a common functionality in the context of database searching in the medical system, economy, and online shopping. For example, given a medical database, it is natural to expect that the trained classifier can predict whether a patient is ill or healthy based on her/his symptoms, and can identify a healthy patient with additional properties, e.g. the gender of the patient is female, which can be modeled by a black box function.

The mathematical formulation of this classification task is as follows. Given the data $\mathcal{D}_k$ in equation (5), denoted the black box as $q(\cdot)$, the task yields

$$\left(\min_{\boldsymbol{\theta}} \sum_{i=0}^{K-1} \mathbb{1}_{y_i \neq f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}\right) \wedge \left(\{j | q(j) = 1,\ y_j = 1\}\right), \tag{16}$$

where the function $q(\cdot)$ is a Boolean function with the input set $\{j : \forall\ y_j \in \mathcal{D}_k,\ y_j = 1\}$. Taking GBLS implemented in the previous subsections as an example, $q(\cdot)$ has the following form, $\forall j = \{0, \ldots, K-1\}$

$$q(j) = \begin{cases} 1, & \text{if } j = K-1; \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

Furthermore, $q(\cdot)$ could be implemented by the MCZ gate, which conditionally flips the phase of the computational basis corresponding to $j^* := K-1$ if the state is $\left|\psi_j^{(1)}\right\rangle_F |j^*\rangle_I$ given in equation (8). In this way, the Grover-like search structure used in GBLS promises that the probability to sample j^* will be maximized. We remark that GBLS can be effectively generalize to implement other forms of $q(\cdot)$ via modifying the MCZ gate. When the size of the dataset loaded by GBLS is K, a well-trained GBLS can locate the target index with $O(\sqrt{K})$ query complexity, guaranteed by the result of theorem 2. However, given

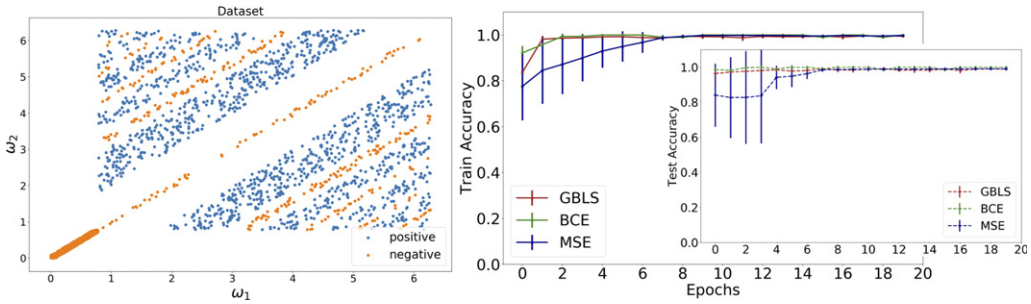

Figure 6. The synthetic dataset and performance of different quantum classifiers under the ideal setting. The left panel illustrates the synthetic dataset used in the numerical simulations. The legend 'positive' (or 'negative') refers that the label of the data is 1 (or 0). The right panel demonstrates the training and test accuracy of different quantum classifiers. The labels 'GBLS', 'BCE', 'MSE' refer to the proposed GBLS, the quantum kernel classifier with BCE loss, and the quantum kernel classifier with the mean square error loss ($B = N$) [23, 24], respectively. The vertical sticks reflect the variance of the test accuracy at each iteration.

access to the well-trained classifier $f_{\boldsymbol{\theta}}(\cdot)$, both classical algorithms and previous quantum classifiers need at least $O(K)$ query complexity to find j^*. The reduced query complexity of GBLS implies a potential quantum advantage to accomplish classification tasks.

3. Numerical experiments

We now apply GBLS to classify a nonlinear synthetic dataset $\hat{\mathcal{D}}$ to evaluate its performance. The construction of $\hat{\mathcal{D}}$ follows the proposal [23]. Consider a synthetic dataset $\hat{\mathcal{D}} = \{\boldsymbol{x}_i, y_i\}_{i=0}^{N-1}$ with $N = 200$, where $\boldsymbol{x}_i = (\omega_1^{(i)}, \omega_2^{(i)}) \in \mathbb{R}^2$, $\omega_1^{(i)}, \omega_2^{(i)} \in (0, 2\pi)$. Let $g(\cdot)$ be a specific embedding function with $\left|g(\omega_1^{(i)}, \omega_2^{(i)})\right\rangle \in \mathbb{C}^4$ for all $i \in \{0, \ldots, N-1\}$. The label of $\boldsymbol{x}_i$ is assigned as $y_i = 1$ if

$$\left\langle g(\omega_1^{(i)}, \omega_2^{(i)}) \right| V^\dagger \Pi V \left| g(\omega_1^{(i)}, \omega_2^{(i)}) \right\rangle \geqslant 0.5 + \Delta,$$

where $V \in SU(4)$ is a unitary operator, $\Pi = \mathbb{I} \otimes |0\rangle\langle 0|$ is the measurement operator, and the gap Δ is set as 0.2. The label of $\boldsymbol{x}_i$ is assigned as $y_i = 0$ if

$$\left\langle g(\omega_1^{(i)}, \omega_2^{(i)}) \right| V^\dagger \Pi V \left| g(\omega_1^{(i)}, \omega_2^{(i)}) \right\rangle \leqslant 0.5 - \Delta.$$

We illustrate the synthetic dataset $\hat{\mathcal{D}}$ in the left panel of figure 6.

At the data preprocessing stage, we split the dataset $\hat{\mathcal{D}}$ into the training datasets $\hat{\mathcal{D}}_{\text{train}}$ with size $N_{\text{train}} = 100$ and the test dataset $\hat{\mathcal{D}}_{\text{test}}$ with $N_{\text{test}} = 100$. In the training process, we follow the construction rule of GBLS to build the extended training dataset $\mathcal{D}_{\text{train}}$ by using $\hat{\mathcal{D}}_{\text{train}}$. We set $K = 4$ in the following analysis, where the training example $\mathcal{D}_k \subset \mathcal{D}_{\text{train}}$ can be encoded into a quantum state by using four qubits with $N_I = N_F = 2$ (see appendix C for the detailed implementation of GBLS). Note that, at each epoch, we shuffle $\mathcal{D}_{\text{train}}$ and rebuild the extended dataset $\hat{\mathcal{D}}_{\text{train}}$. An epoch means that an entire dataset is passed forward through the quantum learning model, e.g. when the dataset contains 1000 training examples, and only two examples are fed into the quantum learning model each time, then it will take 500 iterations to complete 1 epoch.

The numerical simulations are implemented on Python in conjunction with the PennyLane, Qiskit, and pyQuil libraries [28–30]. The hyper-parameters setting used in our experiment is as follows. The block of U_E in figure 4 is employed once for the case $K = 4$, according to the Grover's theorem $O(\sqrt{K})$. The layer number of variational quantum circuits, i.e. $U_{L_1} = \prod_{l=1}^{L} U(\boldsymbol{\theta}^l)$, is set as $L = 2$. The number of epochs used in classical optimization is 20. For comparison, we also apply the quantum kernel classifier proposed by [23, 24] with two different loss functions, i.e. the mean squared error (MES) loss, and the binary cross entropy (BCE) loss, to learn the synthetic dataset $\hat{\mathcal{D}}$. The selection of the quantum kernel classifiers as the reference is based on the fact that this method has achieved state-of-the-art performance to classify nonlinear data [23].

Ideal setting. We first evaluate performance of different quantum classifiers under the ideal setting, where the quantum system is noiseless and the number of measurements is infinite. The right panel of figure 6 illustrates the averaged training and testing accuracies versus the number of epochs. In particular, our proposal achieves comparable performance with the quantum kernel classifier with the BCE loss, where both the train and test accuracies converge to 99% within 2 epochs. Moreover, these two methods

Table 2. Performance of different quantum classifiers under the depolarization noise at the 20-th epoch. The labels 'MSE_batch', 'MSE', 'BCE', and 'GBLS' follow the same meanings as explained in table 1. The value '$a \pm b$' refers that the averaged accuracy is a and its variance is b.

Methods	MSE_batch	MSE	BCE	GBLS
$p = 0.05$ (train)	0.929 ± 0.037	0.978 ± 0.013	0.956 ± 0.024	0.935 ± 0.024
$p = 0.25$ (train)	0.846 ± 0.072	0.936 ± 0.032	0.918 ± 0.031	0.881 ± 0.025
$p = 0.05$ (test)	0.943 ± 0.032	0.975 ± 0.006	0.860 ± 0.089	0.945 ± 0.021
$p = 0.25$ (test)	0.862 ± 0.095	0.934 ± 0.009	0.791 ± 0.056	0.879 ± 0.040

outperform the quantum kernel classifier with the MSE loss ($B = N$), whose test accuracy can only reach 95% after 10 epochs. The variance of these three quantum classifiers after 10 epochs becomes small, which implies that all of them hold stable performance under the ideal setting.

Depolarization noise setting. We next investigate performance of GBLS and the referenced quantum kernel classifiers under the realistic setting, where the quantum system noise is considered and the number of measurements is finite. Specifically, we employ the depolarization channel to model the system noise, i.e. given a quantum state $\rho \in \mathbb{C}^{d\times d}$, the quantum depolarization channel $\mathcal{E}_p$ that acts on this state is defined as

$$\mathcal{E}_p(\rho) = (1-p)\rho + p\pi_d,$$

where p is the depolarization rate, and π_d is the maximally mixed state with $\pi_d = \mathbb{I}_d/d$. Meanwhile, to explore the trade-off between the computational cost (i.e. the total number of measurements) and the utility R_1 indicated by theorem 1, we also compare performance between GBLS and a modified quantum kernel classifier with the MSE loss, which supports to use the batch gradient descent method with $B = N/4$ to optimize parameters (please refer to appendix C for implementation details). Table 1 summarizes the basic information about GBLS and the referenced quantum classifiers. See appendix D about the derivation of the required number of measurements for GBLS and the quantum kernel classifier with the BCE loss.

The hyper-parameters settings applied to GBLS and other quantum classifiers are as follows. The depolarization rate is set as $p = 0.05$ and $p = 0.25$, respectively. The number of measurements is set as 10 to approximate the quantum expectation result. The parameter shift rule is used to estimate the analytic gradients [22, 31]. For each classifier, we repeat the numerical simulations with five times to collect the statistical information. Confer appendix C for other settings such as learning rates and random seeds.

The simulation results of GBLS and the referenced quantum classifiers are illustrated in figure 7. Specifically, when $p = 0.05$, GBLS and the other three referenced quantum classifiers achieve comparable performance after 10 epochs. Moreover, the quantum kernel classifier with the MSE loss ($B = N/4$ possesses a lower the convergence rate and a larger variance than the rest three classifiers. When $p = 0.25$, there exists a relatively large gap between the quantum kernel classifiers with the MSE_bactch method and the rest three quantum classifiers in the measure of the convergence rate. Such a difference reflects the importance to use GBLS to investigate classification tasks under the varied number of batches. We summarize the averaged training and test accuracies of GBLS and other quantum classifiers at the last epoch in table 2. Even though the measurement error and quantum gate noise are considered, GBLS can still attain stable performance, since its variance is very small (i.e. at most 0.04). This observation suggests the applicability of our proposal on NISQ machines.

We would like to emphasize the main issue considered in this study: whether there exists a quantum classifier that can attain a good utility bound R_1 by using a few number of measurements. The numerical simulation results of GBLS provide a positive response toward this issue. Recall the setting given in table 1 and the results in figure 7. Although the required number of measurements for GBLS is reduced by $K = 4$ times compared with quantum classifiers with the BCE loss and the MSE loss ($B = N$), they achieve comparable performance. This result implies a huge separation of the computational efficacy between GBLS and previous quantum classifiers with $B = N$ when N is large.

Noise model from real quantum hardware. We further compare performance of GBLS and the referenced quantum classifiers under a noise model extracted from real quantum hardware, i.e. IBMQ_ourense, provided by the Qiskit and PennyLane python libraries [28, 29]. Notably, for all classifiers, the gate noise is only imposed on the trainable quantum circuits U_L instead of the whole circuits, since the implementation of multi-controlled gates (e.g. CCZ) used in GBLS will introduce a huge amount of noise and destroy the optimization of GBLS (see appendix C for details). Meanwhile, the measurement noise is applied to all quantum classifiers. Due to the relatively poor performance of the quantum kernel classifier

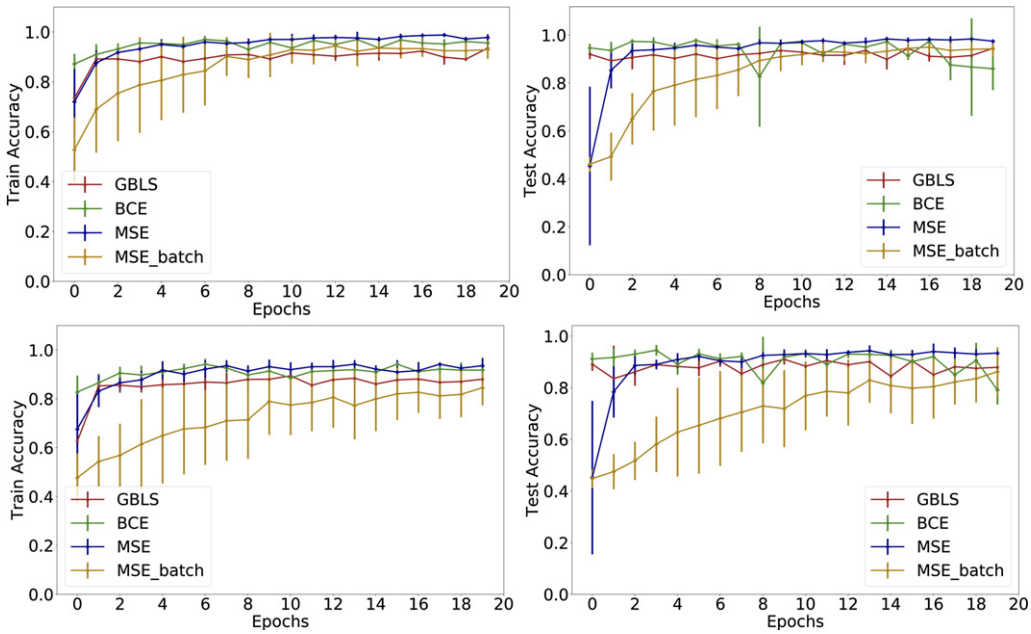

Figure 7. The performance of different quantum classifiers with finite measurements under the varied depolarization rates. The labels 'GBLS', 'BCE', 'MSE', and 'MSE_batch' refer to the proposed Grover-based quantum classifier, the quantum kernel classifier with the BCE loss, the quantum kernel classifier with the mean square error loss and the number of batches being $B = N$, and the quantum kernel classifier with the mean square error loss and the number of batches being $B = N/4$. The upper panel and the lower panel demonstrate the train and test accuracies of GLBS and the quantum kernel classifier with BCE loss when the depolarization rate is set as $p = 0.05$ and $p = 0.25$, respectively. The vertical sticks reflect the variance of the train and test accuracy at each iteration.

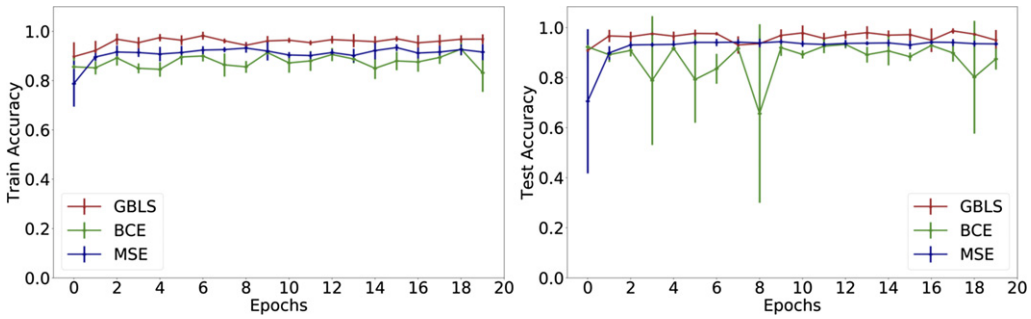

Figure 8. Simulation results of different quantum classifiers under the realistic noise setting. The labels 'GBLS', 'BCE', and 'MSE' have the same meaning as explained in figure 7. The noise model, which is extracted from a real quantum hardware, is applied to the trainable unitary $U_L(\boldsymbol{\theta})$ of these three classifiers.

with the MSE loss and $B = N/4$, here we only focus the comparison among GBLS and quantum kernel classifiers with the BCE loss and the MSE loss ($B = N$). Note that all hyper-parameters settings are identical to those used in the above numerical simulations.

The simulation results are exhibited in figure 8. Specifically, the three classifiers achieve comparable performance. Such results indicate that the efficacy of GBLS, since the required number of measurements for GBLS is reduced by four times compared with the rest two quantum classifiers.

4. Discussion and conclusion

In this study, we have proposed a GBLS for classification. Different from previous proposals, GBLS supports the optimization of a wide range of quantum classifiers with a varied number of batches. This property allows us to explore the trade-off between the computational efficiency and the utility bound R_1. Moreover, we demonstrate that GBLS possesses a potential advantage to tackle certain classification tasks in the measure of query complexity. Numerical experiments showed that GBLS can achieve comparable performance with other advanced quantum classifiers by using a fewer number of measurements. We believe that our work will provide immediate and practical applications for near-term quantum devices.

Acknowledgments

This work received support from Australian Research Council (Project FL-170100117), and the Faculty of Engineering and Information Technologies at the University of Sydney (the Engineering and Information Technologies Research Scholarship).

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/yuxuan-du/.

Appendix A. Proof of theorem 1

Proof of theorem 1.

To achieve theorem 1, we separately discuss the situations in which the label of the last entry in $\mathcal{D}_k$ is $y_k = 1$ and $y_k = 0$, respectively.

For the case $y_k = 1$. Suppose that the label of the last entry in $\mathcal{D}_k$ is $y_k = 1$. Followed from equation (13), after the first cycle, the generated state of GBLS is

$$U|\mathbf{0}\rangle_{F,I} \equiv U_{c_1}\left|\Phi^k(y_k = 1)\right\rangle_{F,I} = |\mathbf{0}\rangle_F \otimes \left(\cos 3\gamma|B\rangle_I + \sin 3\gamma|i^*\rangle_I\right),$$

where $\sin\gamma = \frac{1}{\sqrt{K}}$. This result indicates that the probability to sample the target index i^* is increased from $\sin^2\gamma$ to $\sin^2 3\gamma$, which is same with Grover-search.

Then, by induction as the proof of Grover-search does [32], the generated state of GBLS after applying U to $|\mathbf{0}\rangle_{F,I}$ with ℓ times yields

$$\prod_{i=1}^{\ell} U^i|\mathbf{0}\rangle_{F,I} = |\mathbf{0}\rangle_F \otimes (\cos((2\ell+1)\gamma)|B\rangle_I + \sin((2\ell+1)\gamma)|i^*\rangle_I). \tag{A.1}$$

Note that, GBLS requires that the employed quantum operation at the last cycle is U_E as defined in equation (10) instead of U. Mathematically, the generated state is

$$\begin{aligned} U_E\prod_{i=1}^{\ell} U^i|\mathbf{0}\rangle_{F,I} &= U_{\text{init}} \circ \text{MCZ} \circ (U_{L_1} \otimes \mathbb{I}) \circ U_{\text{data}}|\mathbf{0}\rangle_F \otimes (\cos((2\ell+1)\gamma)|B\rangle_I + \sin((2\ell+1)\gamma)|i^*\rangle_I) \\ &= U_{\text{init}} \circ \text{MCZ}\left(\cos((2\ell+1)\gamma)\left|\psi_B^{(0)}\right\rangle_F|B\rangle_I + \sin((2\ell+1)\gamma)\left|\psi_B^{(1)}\right\rangle_F|i^*\rangle_I\right) \\ &= U_{\text{init}}\left(\cos((2\ell+1)\gamma)\left|\psi_B^{(0)}\right\rangle_F|B\rangle_I - \sin((2\ell+1)\gamma)\left|\psi_B^{(1)}\right\rangle_F|i^*\rangle_I\right) \\ &= \left(\cos((2\ell+3)\gamma)\left|\psi_B^{(0)}\right\rangle_F|B\rangle_I + \sin((2\ell+3)\gamma)\left|\psi_B^{(1)}\right\rangle_F|i^*\rangle_I\right), \end{aligned} \tag{A.2}$$

where the first equality uses equation (A.1), the second equality exploits equation (13) to engineer the feature register, the third equality employs MCZ to flip the phase the state $|i^*\rangle$ whose first qubit in the feature register is $|1\rangle$, and last equality comes from the application of the diffusion operator $U_{\text{init}} = \mathbb{I}_F \otimes (2|\varphi\rangle\langle\varphi| - \mathbb{I}_I)$ with $|\varphi\rangle = \frac{1}{\sqrt{K}}\sum_i |i\rangle$ to the index register.

The result of equation (A.2) indicates that, under the optimal setting, the probability to sample i^* is close to 1 when $\ell \sim O(\sqrt{K})$, since $\sin\gamma \approx \gamma = 1/\sqrt{K}$ and then $\sin((2\ell+3)\gamma) \approx 1$.

For the case $y_k = 0$. We then demonstrate that, when the label of the last entry in $\mathcal{D}_k$ is $y_k = 0$, even if applying $U = \prod_{i=1}^{\ell}$ and U_E to $|\mathbf{0}\rangle_{F,I}$ with $\ell \sim O(\sqrt{K})$, the probability to sample i^* is $1/K$. Followed from equation (11), after the first cycle, the generated state of GBLS is

$$U_{c_1}\left|\Phi^k(y_k = 0)\right\rangle_{F,I} = \frac{1}{\sqrt{K}}\sum_{i=0}^{K-1}|\mathbf{0}\rangle_F|i\rangle_I,$$

where $\sin\gamma = \frac{1}{\sqrt{K}}$. Due to $U_{c_1}\left|\Phi^k(y_k = 0)\right\rangle_{F,I} = U|\mathbf{0}\rangle_{F,I}$, after applying U to the state $|\mathbf{0}\rangle$, the probability to sample any index is identical. By induction, applying the corresponding U to the state $|\mathbf{0}\rangle_{F,I}$ with ℓ times

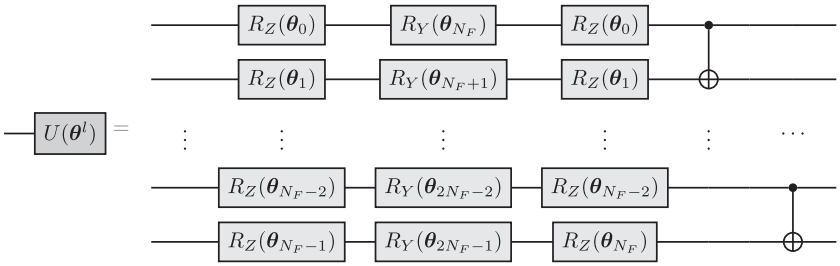

Figure B1. The implementation of the lth layer $U(\boldsymbol{\theta}^l)$. Suppose that the lth layer $U(\boldsymbol{\theta}^l)$ interacts with N_F qubits. Three trainable parameterized gates, R_Z, R_Y and R_Z, are firstly applied to each qubit, followed by $N_F - 1$ CNOT gates.

yields

$$\prod_{i=1}^{\ell} U^i|\mathbf{0}\rangle_{F,I} = \frac{1}{\sqrt{K}}\sum_{i=0}^{K-1}|\mathbf{0}\rangle_F|i\rangle_I, \tag{A.3}$$

where given any positive integer ℓ, the probability to sample $|i^*\rangle_I$ is $1/K$.

As with the case of $y_k = 1$, at the last cycle, we apply the unitary U_E to the state $\prod_{i=1}^{\ell} U^i|\mathbf{0}\rangle_{F,I}$, and the generated state is

$$\begin{aligned} U_E\prod_{i=1}^{\ell} U^i|\mathbf{0}\rangle_{F,I} &= U_{\text{init}}\circ \text{MCZ}\circ(U_{L_1}\otimes\mathbb{I})\circ U_{\text{data}}\frac{1}{\sqrt{K}}\sum_{i=0}^{K-1}|\mathbf{0}\rangle_F|i\rangle_I \\ &= U_{\text{init}}\left(\frac{1}{\sqrt{K}}\sum_{i=0}^{K-1}\left(\left|\psi_B^{(0)}\right\rangle_F|B\rangle_I + \left|\psi_{i^*}^{(0)}\right\rangle_F|i^*\rangle_I\right)\right) \\ &= \frac{1}{\sqrt{K}}\sum_{i=0}^{K-1}\left(\left|\psi_B^{(0)}\right\rangle_F|B\rangle_I + \left|\psi_{i^*}^{(0)}\right\rangle_F|i^*\rangle_I\right), \end{aligned} \tag{A.4}$$

where the first equality uses the explicit form of U_E and equation (A.3), and the second equality is guaranteed by equation (12) (note that the only difference is replacing $\left|\psi_{i^*}^{(1)}\right\rangle_F$ with $\left|\psi_{i^*}^{(0)}\right\rangle_F$ based on the setting $y_k = 0$), and the last equality exploits the explicit form of U_{init}.

The result of equation (A.4) reflects that, under the optimal setting, the probability to sample i^* can never be increased when $y_k = 0$. Therefore, we can conclude that, under the optimal setting, the probability to sampling the outcome i^* approaches 1 asymptotically if and only if the label of the last entry of $\mathcal{D}_k$ is $y_k = 1$. □

Appendix B. Variational quantum circuits and the optimizing method

In this section, we first introduce the variational quantum circuits $U_{L_1}(\boldsymbol{\theta})$ used in GBLS. We then elaborate the optimization method, i.e. the parameter shift rule, that is employed to train $U_{L_1}(\boldsymbol{\theta})$.

Variational quantum circuits, which is also called parameterized quantum circuit, are composed of trainable single qubit gates and two qubits gates (e.g. CNOT or CZ). As a promising scheme for NISQ devices, variational quantum circuits have been extensively investigated for accomplishing the generative and discriminative [15, 20, 33–35] tasks via variational hybrid quantum–classical algorithms [36]. One typical variational quantum circuits is called the multiple-layer parameterized quantum circuits (MPQC), where the arrangement of quantum gates in each layer is identical [33]. Denote the operation formed by the lth layer as $U(\boldsymbol{\theta}^l)$. The generated quantum state from MPQC yields

$$|\Psi\rangle = \prod_{l=1}^{L} U(\boldsymbol{\theta}^l)|0\rangle^{\otimes N_F},$$

where L is the total number of layers. GBLS employs MPQC to construct U_{L_1}, i.e.

$$U_{L_1}(\boldsymbol{\theta}) = \prod_{l=1}^{L} U(\boldsymbol{\theta}^l), \tag{B.1}$$

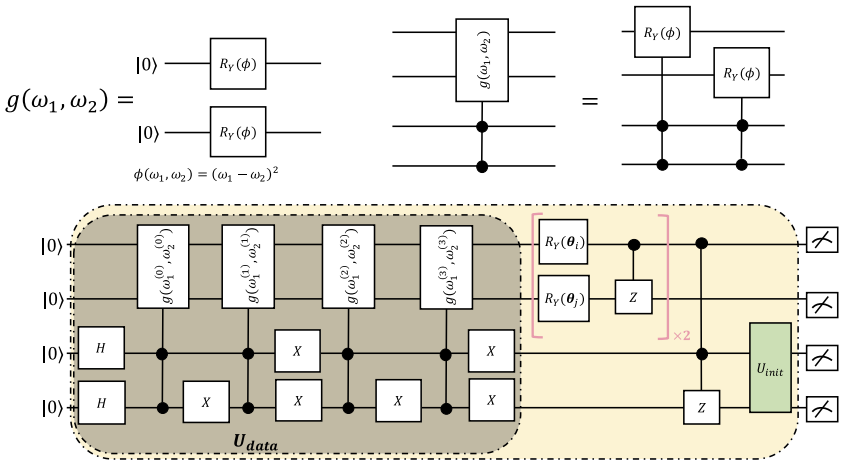

Figure B2. The implementation of GBLS used in numerical simulations. The upper left panel illustrates the circuit implementation of the encoding unitary U_{data} corresponding to the feature map $g(\boldsymbol{x}_i)$. The lower panel demonstrates the implementation of GBLS given the input $\mathcal{D}_k = \{\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k, \boldsymbol{x}_l\}$, where the implementation of the controlled-$g(\boldsymbol{x}_i)$ quantum operation is shown in the upper right panel.

and the circuit arrangement for the lth layer $U(\boldsymbol{\theta}^l)$ is shown in figure B1. When the number of layers is L, the total number of trainable parameters for GBLS is $2N_F L$.

The updating rule of GBLS at the kth iteration follows

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \eta \frac{\mathcal{L}(\boldsymbol{\theta}^{(k)}, \mathcal{D}_k)}{\partial \boldsymbol{\theta}}, \tag{B.2}$$

where η is the learning rate and $\mathcal{D}_k$ is the kth training example. By expanding the explicit form of $\mathcal{L}(\boldsymbol{\theta}^{(k)}, \mathcal{D}_k)$ given in equation (14), the gradients of $\mathcal{L}(\boldsymbol{\theta}^{(k)}, \mathcal{D}_k)$ can be rewritten as

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}^{(k)}, \mathcal{D}_k)}{\partial \boldsymbol{\theta}} = \text{sign}(1/2 - y_k) \frac{\partial \text{ Tr}(\Pi \rho(\boldsymbol{\theta}^{(k)}))}{\partial \boldsymbol{\theta}}, \tag{B.3}$$

where y_k refers to the label of the last entry in $\mathcal{D}_k$, sign$(\cdot)$ is the sign function, Π is the measurement operator, and

$$\rho(\boldsymbol{\theta}^{(k)}) = U_E U(\boldsymbol{\theta}^{(k)})^{O(\sqrt{K})} |\mathbf{0}\rangle \langle \mathbf{0}| (U_E U(\boldsymbol{\theta}^{(k)})^{O(\sqrt{K})})^\dagger.$$

GBLS adopts the parameter shift rule proposed by [22] to attain the gradient $\frac{\partial \text{ Tr}(\Pi \rho(\boldsymbol{\theta}^{(k)}))}{\partial \boldsymbol{\theta}}$. Concisely, the parameter shift rule iteratively computes each entry of the gradient. Without loss of generality, here we explain how to compute $\frac{\partial \text{ Tr}(\Pi \rho(\boldsymbol{\theta}^{(k)}))}{\partial \boldsymbol{\theta}_j}$ for $j \in [2N_F L]$. Define $\boldsymbol{\theta}_{\pm}^{(k)}$ as

$$\boldsymbol{\theta}_{\pm}^{(k)} = \left[\boldsymbol{\theta}_0^{(k)}, \ldots, \boldsymbol{\theta}_{j-1}^{(k)}, \boldsymbol{\theta}_j^{(k)} \pm \frac{\pi}{2}, \boldsymbol{\theta}_{j+1}^{(k)}, \ldots, \boldsymbol{\theta}_{2N_F L-1}^{(k)}\right], \tag{B.4}$$

where only the jth parameter is rotated by $\pm\frac{\pi}{2}$. Then the mathematical representation of the gradient for the jth entry is

$$\frac{\partial \text{ Tr}(\Pi \rho(\boldsymbol{\theta}^{(k)}))}{\partial \boldsymbol{\theta}_j} = \frac{\text{Tr}(\Pi \rho(\boldsymbol{\theta}_+^{(k)})) - \text{Tr}(\Pi \rho(\boldsymbol{\theta}_-^{(k)}))}{2}. \tag{B.5}$$

In conjunction with equations (B.2), (B.3) and (B.5), the updating rule of GBLS at the tth iteration for the jth entry is

$$\boldsymbol{\theta}_j^{(k+1)} = \boldsymbol{\theta}_j^{(k)} - \eta \frac{\text{Tr}(\Pi \rho(\boldsymbol{\theta}_+^{(k)})) - \text{Tr}(\Pi \rho(\boldsymbol{\theta}_-^{(k)}))}{2} \text{ sign}\left(\frac{1}{2} - y_k\right). \tag{B.6}$$

Appendix C. More details of numerical simulations

In this section, we provide more details about the numerical simulations. Specifically, we first explain how to construct the employed synthetic dataset. We then elaborate on the implementation of GBLS and referenced classifiers, and their hyper-parameters settings. We next analyze the required circuit depth to implement these quantum classifiers. Last, we introduce the construction of the modified dataset used in the MSE_batch method.

The construction of the synthetic dataset. Given the training example $\boldsymbol{x}_i = (\omega_1^{(i)}, \omega_2^{(i)}) \in \mathbb{R}^2$ for all $i \in [N-1]$, the embedding function $g(\omega_1^{(i)}, \omega_2^{(i)}) : \mathbb{R}^2 \to \mathbb{C}^4$ that is used to encode $\boldsymbol{x}_i$ into the quantum states is formulated as

$$g(\omega_1^{(i)}, \omega_2^{(i)}) = \left(R_Y(\phi(\omega_1^{(i)}, \omega_2^{(i)})) \otimes R_Y(\phi(\omega_1^{(i)}, \omega_2^{(i)}))\right)|0\rangle^{\otimes 2}, \tag{C.1}$$

where $\phi(\omega_1^{(i)}, \omega_2^{(i)}) = (\omega_1^{(i)} - \omega_2^{(i)})^2$ is a specified mapping function. The above formulation implies that $g(\boldsymbol{x}_i)$ can be converted to a sequence of quantum operations, where its implementation is illustrated in the upper left panel of figure B2. To simultaneously encode multiple training examples into the quantum states, we should implement $g(\boldsymbol{x}_i)$ as a controlled version, where the implementation is shown in the upper right panel of figure B2.

The random unitary $V \in SU(4)$ used in the numerical simulations is formulated as $V = R_Y(\psi_1) \otimes R_Y(\psi_2)$, where ψ_1 and ψ_1 are uniformly sampled from $[0, 2\pi)$.

The details of GBLS, the referenced classifiers, and hyper-parameters setting. The implementation of GBLS is shown the lower panel of figure B2. In particular, the data encoding unitary U_{data} is composed of a set of controlled-$g(\boldsymbol{x}_i)$ quantum operations. The MPQC introduced in appendix B is employed to build $U_{L_1}(\boldsymbol{\theta})$, where each layer $U(\boldsymbol{\theta}^l)$ is composed of R_Y gates and CZ gates and the layer number is $L = 2$.

The basic components of the referenced quantum classifiers are identical to those used in GBLS. In particular, for all employed quantum kernel classifiers, the implementation of variational quantum circuits $U_{L_1}(\boldsymbol{\theta})$ are the same with GBLS, where the layer number is $L = 2$ and each layer is composed of R_Y gates and CZ gates as shown in figure B2. The implementation of the encoding unitary U_{data} depends on the batch size B. For the quantum kernel classifiers with the BCE loss and MSE loss ($B = N$), following equation (C.1), the encoding unitary is

$$U_{\text{data}} = R_Y(\phi(\omega_1^{(i)}, \omega_2^{(i)})) \otimes R_Y(\phi(\omega_1^{(i)}, \omega_2^{(i)})). \tag{C.2}$$

For the quantum kernel classifier with the MSE loss ($B = N/4$), the implementation of the encoding unitary U_{data} is the same with GBLS as shown in figure B2.

The detailed hyper-parameters settings for GBLS and the referenced classifiers are as follows. The learning rate for GBLS, the quantum kernel classifier with the BCE loss, the quantum kernel classifier with the MSE loss ($B = N$ and $B = N/4$) is identical, which is set as $\eta = 1.0$. Moreover, when we explore the statistical performance of different quantum classifiers under the noise setting, the random seeds are set as $\{i\}_{i=1}^{R}$ with R being the total number of repetitions.

The analysis of the quantum circuit depth. Here we analyze the required circuit depth to implement quantum kernel classifiers used in numerical simulations. As explained in the above subsection, the quantum kernel classifiers with $B = N$ can be efficiently realized, since the data encoding unitary U_{data} and the variational quantum circuits only involve single and two qubits gates. In particular, the circuit depth to construct the unitary U_{data} in equation (C.2) is 1. Moreover, the circuit depth to construct $U_L(\boldsymbol{\theta})$ as shown in figure B2 is 4. In total, when the number of batches B equals to N, the required depth for the quantum kernel classifier with the BCE or MSE loss is 5.

Compared with the setting $B = N$, the implementation of the quantum kernel classifier with $B = N/4$ and GBLS requires a relatively deep circuits. The substantial reason is that the fabrication of the data encoding unitary U_{data} involves multi-controlled qubits gates as shown in figure B2 (highlighted by the brown region). Specifically, when we decompose the CC–R_Y gate into single-qubit and two-qubit gates, the required circuit depth is 27. Therefore, following figure B2, the circuit depth to implement U_{data} is 113. Considering that the circuit depth to implement U_{L_1} is 4, the total circuit depth to implement the quantum kernel classifier with $B = N/4$ is 117. As shown in figure B2, the quantum circuit in GBLS is composed of U_{data}, U_{L_1}, and U_{init}. The implementation of U_{data} and U_{L_1} is identical to the quantum kernel classifier with $B = N/4$. Moreover, based on Grover-search algorithm, the circuit depth to implement U_{init} is 15, which includes 4 Hadamard gates and 1 CCZ gate. Therefore, the total circuit depth to implement GBLS is 132.

We remark that the circuit depth of the quantum kernel classifier with $B = N/4$ and GBLS is dominated by the implementation of U_{data}, which exploits multi-controlled qubits gates to load different training examples in superposition. Such an observation implies that efficient encoding methods can dramatically reduce the required circuit depth to construct these quantum classifiers. A possible solution is proposed by [37], which constructs a target multi-qubits gate by optimizing a variational quantum circuit which consists of tunable single-qubit gates and fixed two qubits gates.

The modified training dataset for the MSE_batch method. We note that naively employing the original training dataset $\hat{\mathcal{D}}$ to optimize the quantum kernel classifier with the MSE_batch loss is infeasible. Let us illustrate a simple example. Suppose the input state is $\frac{1}{\sqrt{2}}\sum_{i=1}^{2} |g(\boldsymbol{x}^{(i)})\rangle_F |i\rangle_I$ with the batch size 2, where the

subscript 'I' ('F') refers to the index (feature) register. When the trainable quantum circuits $U_L(\boldsymbol{\theta}) \otimes \mathbb{I}_I$ and the measurement operator are applied to this state, the output corresponds to the averaged predictions of the examples $\{\boldsymbol{x}^{(i)}\}_{i=1}^2$. Such a setting is ill-posed once the labels $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(i)}$ of are opposite, e.g. the former is 0 and the latter is 1, since a wrong prediction (the former is 1 and the latter is 0) also leads to the averaged truth label 0.5.

To conquer the above issue, we build a modified dataset instead of $\hat{\mathcal{D}}$ to optimize the quantum kernel classifier with the MSE_batch loss. Specifically, we shuffle the given dataset $\hat{\mathcal{D}}$ and ensure that for the modified dataset, the training examples in each batch $\mathcal{B}_i$ for $\forall i \in [B]$ must possess the same label. In doing so, the averaged truth label can either be 0 and 1 without any confusion.

Appendix D. The computational complexity of GBLS and the quantum kernel classifier with the BCE loss

We now separately derive the required number of measurements, or equivalently, the computational complexity, for GBLS and the quantum kernel classifier with the BCE loss at each epoch. For both methods, the hyper-parameters setting is supposed to be identical, i.e. the size of the dataset $\hat{\mathcal{D}}$ is N, the layer number of MPQC U_{L_1} is L, the number of qubits to load data features is N_F, the total number of trainable parameters $\boldsymbol{\theta}$ is $N_F L$, and the number of measurements applied to estimate the quantum expectation value is M.

We say one query when the variational quantum circuit used in the quantum classifier takes the encoded data and then be measured by the measurement operator once. Following the training mechanism of the quantum classifier, its query complexity amounts to counting the total number of measurements to the variational quantum circuits to acquire the gradients in one epoch.

We now derive the required number of measurements of the quantum kernel classifier with the BCE loss in one epoch. Given the dataset $\hat{\mathcal{D}}$, the BCE loss yields

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i=0}^{N-1} y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i)), \tag{D.1}$$

where y_i is the label of the ith example and $p(y_i)$ is the predicted probability of the label y_i, or equivalently, the output of the quantum circuit used in the quantum kernel classifier

$$p(y_i) = \mathrm{Tr}(\Pi\rho(\boldsymbol{\theta})), \tag{D.2}$$

where $\rho(\boldsymbol{\theta}) = U_{L_1}(\boldsymbol{\theta})\,|g(\boldsymbol{x}_i)\rangle\,\langle g(\boldsymbol{x}_i)|\,U_{L_1}(\boldsymbol{\theta})^\dagger$, $U_{L_1}(\boldsymbol{\theta})$ refers to variational quantum circuits defined in equation (B.1), $|g(\boldsymbol{x}_i)\rangle$ represents the encoded quantum state defined in equation (C.1), and Π is the measurement operator. Following the parameter shift rule, the derivative of BCE loss satisfies

$$\frac{\partial \mathcal{L}_{\mathrm{BCE}}}{\partial \boldsymbol{\theta}_j} = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{1-y_i}{1-p(y_i)} - \frac{y_i}{p(y_i)}\right)\frac{\mathrm{Tr}(\Pi\rho(\boldsymbol{\theta}_+)) - \mathrm{Tr}(\Pi\rho(\boldsymbol{\theta}_-))}{2}, \tag{D.3}$$

where $\boldsymbol{\theta}_\pm$ is defined in equation (B.4). The above equation implies that to acquire the gradients of the BCE loss, it necessitates to feed the training example one by one to the quantum kernel classifier to estimate $p(y_i)$, and then conduct the classical post-processing to compute the coefficient $\frac{1-y_i}{1-p(y_i)} - \frac{y_i}{p(y_i)}$. In other words, the number of batches for this quantum classifier can only be $B = N$. Since the estimation of $p(y_i)$, $\mathrm{Tr}(\Pi\rho(\boldsymbol{\theta}_+))$, and $\mathrm{Tr}(\Pi\rho(\boldsymbol{\theta}_-))$ are completed by using M measurements, the derivative $\partial\mathcal{L}_{\mathrm{BCE}}/\partial\boldsymbol{\theta}_j$ can be estimated by using $3NM$ measurements. Considering that there are in total $N_F L$ trainable parameters, the total number of measurements at each epoch for the quantum kernel classifier with the BCE loss is $3NMN_F L$.

Unlike the quantum kernel classifier with the BCE loss, GBLS uses a simple loss function $\mathcal{L}$ defined in equation (14), which allows us to efficiently acquire the gradient $\partial\mathcal{L}/\partial\boldsymbol{\theta}_j$ by leveraging the superposition property. Recall equation (B.6). The gradient of GBLS satisfies

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_k)}{\partial \boldsymbol{\theta}_j} = \frac{\mathrm{Tr}(\Pi\rho(\boldsymbol{\theta}_+^{(k)})) - \mathrm{Tr}(\Pi\rho(\boldsymbol{\theta}_-^{(k)}))}{2}\ \mathrm{sign}\left(\frac{1}{2} - y_k\right),$$

where y_k refers to the label of the last pair in the extended training example $\mathcal{D}_k$. The above equation indicates that the gradient for $\mathcal{D}_k$, which contains K training examples in $\hat{\mathcal{D}}$, can be estimated by using $2M$ measurements, where the first (last) M measurements aim to approximate $\mathrm{Tr}(\Pi\rho(\boldsymbol{\theta}_-^{(k)}))$ $(\mathrm{Tr}(\Pi\rho(\boldsymbol{\theta}_+^{(k)})))$. Therefore, the total number of measurements to collect $\{\frac{\partial\mathcal{L}(\boldsymbol{\theta},\mathcal{D}_k)}{\partial\boldsymbol{\theta}_j}\}$ for all possible $\mathcal{D}_k$ is $2MB = 2MN/K$.

Considering that there are in total $N_F L$ trainable parameters, the query complexity at each epoch for GBLS is $2N_F LMN/K$. Note that when $K \to N$, the required number of measurements of GBLS can be dramatically reduced.

To ease of understanding, let us illustrate an intuitive example. Define two extended training examples, where the first one includes all positive examples in $\mathcal{D}$ and one negative example, and the second one includes all negative examples in $\mathcal{D}$ and one positive example. Since these two extended examples cover the whole dataset $\mathcal{D}$, when GBLS uses these two examples to update $\boldsymbol{\theta}$, it completes one epoch. Celebrated by the simple form of $\mathcal{L}$, the number of measurements to estimate the gradients for the jth entry $\boldsymbol{\theta}_j$ given these two extended examples is $O(1)$. Considering there are in total $O(N_F L)$ trainable parameters, the total number of measurements at each epoch for GBLS is $O(LN_F)$.

ORCID iDs

Yuxuan Du https://orcid.org/0000-0002-5997-7882
Min-Hsiu Hsieh https://orcid.org/0000-0002-3396-8427

References

[1] Goodfellow I, Bengio Y, Courville A and Bengio Y 2016 *Deep Learning* vol 1 (Cambridge, MA: MIT Press)
[2] Carrasquilla J and Melko R G 2017 *Nat. Phys.* **13** 431
[3] Van Nieuwenburg E P L, Liu Y-H and Huber S D 2017 *Nat. Phys.* **13** 435
[4] Carleo G and Troyer M 2017 *Science* **355** 602–6
[5] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195
[6] Harrow A W, Hassidim A and Lloyd S 2009 *Phys. Rev. Lett.* **103** 150502
[7] Lloyd S, Mohseni M and Rebentrost P 2014 *Nat. Phys* **10** 631
[8] Wang G 2017 *Phys. Rev.* A **96** 012335
[9] Du Y, Liu T, Li Y, Duan R and Tao D 2018 Quantum divide-and-conquer anchoring for separable non-negative matrix factorization *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence* pp 2093–9
[10] Rebentrost P, Steffens A, Marvian I and Lloyd S 2018 *Phys. Rev.* A **97** 012327
[11] Preskill J 2018 *Quantum* **2** 79
[12] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 *Quantum Sci. Technol.* **4** 043001
[13] Du Y, Hsieh M H, Liu T, You S and Tao D 2020 arXiv: 2007.12369
[14] Cerezo M *et al* 2020 arXiv: 2012.09265
[15] Farhi E and Neven H 2018 arXiv: 1802.06002
[16] Arute F *et al* 2019 *Nature* **574** 505–10
[17] Farhi E and Harrow A W 2019 arXiv: 1602.07674
[18] Bremner M J, Jozsa R and Shepherd D J 2010 Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy *Proc. R. Soc.* A **467** 459–72
[19] Schuld M, Fingerhuth M and Petruccione F 2017 *Europhys. Lett.* **119** 60002
[20] Schuld M, Bocharov A, Svore K M and Wiebe N 2020 *Phys. Rev.* A **101** 032308
[21] Wilson C *et al* 2019 arXiv: 1806.08321
[22] Mitarai K, Negoro M, Kitagawa M and Fujii K 2018 *Phys. Rev.* A **98** 032309
[23] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 *Nature* **567** 209
[24] Schuld M and Killoran N 2019 *Phys. Rev. Lett.* **122** 040504
[25] Cong I, Choi S and Lukin M D 2019 *Nat. Phys.* **15** 1273–8
[26] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc. of the Twenty-eighth Annual ACM Symp. on Theory of Computing (ACM)* 212–9
[27] Morales M E, Tlyachev T and Biamonte J 2018 *Phys. Rev.* A **98** 062333
[28] Bergholm V *et al* 2018 arXiv: 1811.04968
[29] Aleksandrowicz G *et al* 2019 https://github.com/Qiskit/qiskit (Accessed: 16 March 2019)
[30] Smith R S, Curtis M J and Zeng W J 2016 arXiv: 1608.03355
[31] Schuld M, Bergholm V, Gogolin C, Izaac J and Killoran N 2019 *Phys. Rev.* A **99** 032331
[32] Brassard G, Hoyer P, Mosca M and Tapp A 2000 arXiv: quant-ph/0005055
[33] Benedetti M, Garcia-Pintos D, Perdomo O, Leyton-Ortega V, Nam Y and Perdomo-Ortiz A 2019 *npj Quantum Inf.* **5** 1–9
[34] Du Y, Hsieh M H, Liu T and Tao D 2020 *Phys. Rev. Res.* **2** 033125
[35] Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, Green A G and Severini S 2018 *npj Quantum Inf.* **4** 1–8
[36] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 *New J. Phys.* **18** 023023
[37] Heya K, Suzuki Y, Nakamura Y and Fujii K 2018 arXiv: 1810.12745