SCIENCE CHINA
Physics, Mechanics & Astronomy

• Article •

May 2024 Vol. 67 No. 5: 250311
https://doi.org/10.1007/s11433-023-2337-2

A quantum federated learning framework for classical clients

Yanqi Song[1], Yusen Wu[2], Shengyao Wu[1], Dandan Li[3], Qiaoyan Wen[1], Sujuan Qin[1*], and Fei Gao[1*]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China;*
[2] *Department of Physics, The University of Western Australia, Perth, WA 6009, Australia;*
[3] *School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China*

Received December 18, 2023; accepted February 1, 2024; published online March 19, 2024

Quantum federated learning (QFL) enables collaborative training of a quantum machine learning (QML) model among multiple clients possessing quantum computing capabilities, without the need to share their respective local data. However, the limited availability of quantum computing resources poses a challenge for each client to acquire quantum computing capabilities. This raises a natural question: Can quantum computing capabilities be deployed on the server instead? In this paper, we propose a QFL framework specifically designed for classical clients, referred to as CC-QFL, in response to this question. In each iteration, the collaborative training of the QML model is assisted by the shadow tomography technique, eliminating the need for quantum computing capabilities of clients. Specifically, the server constructs a classical representation of the QML model and transmits it to the clients. The clients encode their local data onto observables and use this classical representation to calculate local gradients. These local gradients are then utilized to update the parameters of the QML model. We evaluate the effectiveness of our framework through extensive numerical simulations using handwritten digit images from the MNIST dataset. Our framework provides valuable insights into QFL, particularly in scenarios where quantum computing resources are scarce.

quantum federated learning, classical clients, shadow tomography technique

PACS number(s): 03.67.Ac, 03.67.Lx, 07.05.Mh

Citation: Y. Song, Y. Wu, S. Wu, D. Li, Q. Wen, S. Qin, and F. Gao, A quantum federated learning framework for classical clients, Sci. China-Phys. Mech. Astron. **67**, 250311 (2024), https://doi.org/10.1007/s11433-023-2337-2

1 Introduction

Machine learning has made significant progress and brought revolutionary changes across various fields [1-5]. However, the reliance of machine learning on sensitive data within the centralized training framework, where data are collected on a single device, poses risks of data leakage. In response, federated learning [6] has emerged as a decentralized training framework that allows multiple devices to collaboratively train a machine learning model without sharing their data.

In the era of big data, as the scale of data continues to increase, the computational requirements for machine learning are expanding. Simultaneously, theoretical research indicates that quantum computing holds the potential to accelerate the solution of certain problems that pose challenges to classical computers [7-9]. Consequently, the field of quantum machine learning (QML) [10-12] has gained widespread attention, with several promising breakthroughs. On one hand, quantum basic linear algebra subroutines, such as Fourier transforms, eigenvector and eigenvalue computations, and linear equation solving, exhibit exponential quantum speedups compared to their well-established clas-

*Corresponding authors (Sujuan Qin, email: qsujuan@bupt.edu.cn; Fei Gao, email: gaof@bupt.edu.cn)

sical counterparts [13-15]. These subroutines bring quantum speedups in a range of machine learning algorithms, including least-squares fitting [16], gradient descent [17-19], principal component analysis [20], semidefinite programming [21], and support vector machines [22, 23]. Additionally, another significant development in this field is the quantum neural network (QNN), which is a hybrid quantum-classical machine learning model [24-27]. QNN has demonstrated success in various tasks, including classification [28-36], regression [37-39], generative learning [40-42], and reinforcement learning [43, 44].

Quantum federated learning (QFL) [45] extends the concept of data privacy preservation into the field of QML. Specifically, the QFL framework consists of a classical server and multiple quantum clients. During each training iteration, the client utilizes its data-encoding quantum states and its QML model to calculate the local gradient. The local gradients of all the clients are then uploaded to the server. The server performs an aggregation process using the received local gradients and shares the global gradient with all clients for updating their model parameters. Since the proposal of QFL, significant research progress has been made in various aspects, including frameworks [46, 47], privacy protocols [48-52], performance optimization techniques [53], and considerations for application-specific scenarios [54,55]. However, the scarcity of quantum computing resources poses a challenge for each client to acquire quantum computing capabilities, thereby restricting the applicability of QFL. To overcome this challenge, a solution is to deploy quantum computing capabilities on the server rather than on clients. In this regard, two relevant works were proposed [50, 56], which enable clients with limited quantum technologies to collaboratively train the QML model on the server. Specifically, Sheng et al. [56] proposed a distributed secure quantum machine learning protocol where clients perform single-qubit preparation, operation, and measurement. Additionally, Li et al. [50] proposed a private distributed learning framework based on blind quantum computing [57], where clients prepare single qubits for transmission to the server. Although the preparation of single qubits may not pose significant challenges, ensuring high fidelity in the transmission of qubits through the quantum channel remains a complex issue under current conditions. Thus, the absence of QFL applicable to classical clients emphasizes the necessity for further research in this particular scenario.

In this paper, we propose a QFL framework specifically designed for classical clients, referred to as CC-QFL. In our framework, we encode the local data of each client onto observables instead of quantum states. During the collaborative training of the QML model on the server, we employ the shadow tomography technique [58-63] to remove the necessity of quantum computing capabilities for clients. In detail, the server constructs a classical representation of the QML model, which is then transmitted to the clients. Leveraging this classical representation, clients calculate local gradients based on data-encoding observables, and these local gradients are subsequently utilized to update the parameters of the QML model. We conduct extensive numerical simulations of our framework using handwritten digit images from the MNIST dataset. The excellent numerical performance validates the feasibility and effectiveness of CC-QFL. Our framework contributes to the advancement of QFL, particularly in scenarios where quantum computing resources are limited.

2 QFL framework

In this section, we present the conventional framework of QFL. To provide a clear description, we begin with a detailed introduction to the QNN model which represents a hybrid quantum-classical machine learning model.

2.1 Structure of QNN model

The QNN model begins by encoding the classical data $\boldsymbol{x} \in \mathbb{R}^d$ onto a quantum state $\rho(\boldsymbol{x})$. This data encoding process involves applying an encoding quantum circuit $U(\boldsymbol{x})$ to an initial state ρ_0. Following the data encoding process, a parameterized quantum circuit $V(\boldsymbol{\theta})$ is applied to $\rho(\boldsymbol{x})$, where the model parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ can be updated during the training process. Afterward, the resulting state is measured with respect to an observable O, yielding the following parameterized expectation value:

$$E(\boldsymbol{\theta}) = \mathrm{Tr}[V(\boldsymbol{\theta})\rho(\boldsymbol{x})V^{\dagger}(\boldsymbol{\theta})O], \tag{1}$$

where $\rho(\boldsymbol{x}) = U(\boldsymbol{x})\rho_0 U^{\dagger}(\boldsymbol{x})$.

Now, $E(\boldsymbol{\theta})$ can be used to calculate a loss function $\mathcal{L}(\boldsymbol{\theta})$ that quantifies the difference between the predicted output of the QNN model and the expected output. The selection of $\mathcal{L}(\boldsymbol{\theta})$ depends on the task type. For regression tasks, mean squared error is commonly used, while for classification tasks, cross-entropy loss is often employed. In training the QNN model, classical optimization methods such as stochastic gradient descent (SGD) [64] are employed to iteratively update $\boldsymbol{\theta}$ to optimize $\mathcal{L}(\boldsymbol{\theta})$.

2.2 Framework of QFL

Now, we provide a detailed introduction to QFL which comprises a classical server and N quantum clients. Given that the i-th client holds the local data:

$$D^{(i)} = \{\boldsymbol{x}_j^{(i)}\}_{j=1}^{m_i}, \tag{2}$$

where $\boldsymbol{x}_j^{(i)} \in \mathbb{R}^d$, and m_i represents the number of data held by the i-th client. In each training iteration, each client i encodes its local data $D^{(i)}$ onto quantum states:

$$\rho^{(i)} = \{\rho(\boldsymbol{x}_j^{(i)})\}_{j=1}^{m_i}, \tag{3}$$

where $\rho(\boldsymbol{x}_j^{(i)}) = U(\boldsymbol{x}_j^{(i)})\rho_0 U^\dagger(\boldsymbol{x}_j^{(i)})$. Next, each client i utilizes its data-encoding quantum states $\rho^{(i)}$ and its parameterized quantum circuit $V(\boldsymbol{\theta})$ to calculate the local gradient $\boldsymbol{g}_{\text{Local}_i}$ based on the specific form of the loss function $\mathcal{L}(\boldsymbol{\theta})$. The local gradients of all the clients $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^N$ are then uploaded to the server. The server performs an aggregation process using $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^N$ and obtains the global gradient:

$$\boldsymbol{g}_{\text{Global}} = \sum_{i=1}^{N} w_i \boldsymbol{g}_{\text{Local}_i}. \tag{4}$$

In the natural setting, the weight $w_i = m_i/m$, where m represents the total number of data from all clients. The server shares $\boldsymbol{g}_{\text{Global}}$ with all clients, allowing each client to update its corresponding model parameters as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \boldsymbol{g}_{\text{Global}}, \tag{5}$$

where η represents the learning rate. The process of calculating local gradients, aggregating these gradients, and updating the model parameters on each client is repeated iteratively until a predefined number of iterations is reached. The complete framework of QFL is illustrated in Figure 1.

However, the limited availability of quantum computing resources presents a challenge as clients may struggle to acquire their quantum computing capabilities, consequently restricting the practicality of QFL. To address this challenge, it is worth considering a paradigm shift by placing quantum computing capabilities on the server instead of the clients. In the following section, we will provide a comprehensive description of our QFL framework specifically designed for classical clients, referred to as CC-QFL.

3 Main results: CC-QFL framework

As a fundamental component of the QFL framework, we first introduce our QNN model.

3.1 Structure of our QNN model

In our QNN model, we directly apply a parameterized quantum circuit $V(\boldsymbol{\theta})$ to an initial state ρ_0 to obtain a parameterized quantum state $\rho(\boldsymbol{\theta})$, where the model parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ can be updated during the training process. Subsequently, we measure $\rho(\boldsymbol{\theta})$ with respect to a data-encoding observable $O(\boldsymbol{x})$, where the classical data $\boldsymbol{x} \in \mathbb{R}^d$, resulting in the following parameterized expectation value:

$$\tilde{E}(\boldsymbol{\theta}) = \text{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})], \tag{6}$$

where $\rho(\boldsymbol{\theta}) = V(\boldsymbol{\theta})\rho_0 V^\dagger(\boldsymbol{\theta})$, and $O(\boldsymbol{x}) = \sum_{h=1}^{l} c_h(\boldsymbol{x})O_h$. The coefficient $c_h(\boldsymbol{x}) \in \mathbb{R}$ associated with the Hermitian matrix O_h is the h-th component of $c(\boldsymbol{x}) \in \mathbb{R}^l$ obtained by a multivariable function $c(\cdot) : \mathbb{R}^d \to \mathbb{R}^l$. Specifically, $c(\cdot)$ could be a linear function that maintains the domain dimension, i.e., $l = d$. A simple example is $c(\boldsymbol{x}) = \boldsymbol{x}$, resulting in $O(\boldsymbol{x}) = \sum_{i=1}^{d} x_i O_i$, where x_i is the i-th component of $\boldsymbol{x} \in \mathbb{R}^d$. Additionally, $c(\cdot)$ could be a more complex function which makes the data encoding approach more flexible.

Now, $\tilde{E}(\boldsymbol{\theta})$ can be used to calculate a loss function $\tilde{\mathcal{L}}(\boldsymbol{\theta})$, and classical optimization methods such as SGD can be employed to iteratively update $\boldsymbol{\theta}$ during the training process

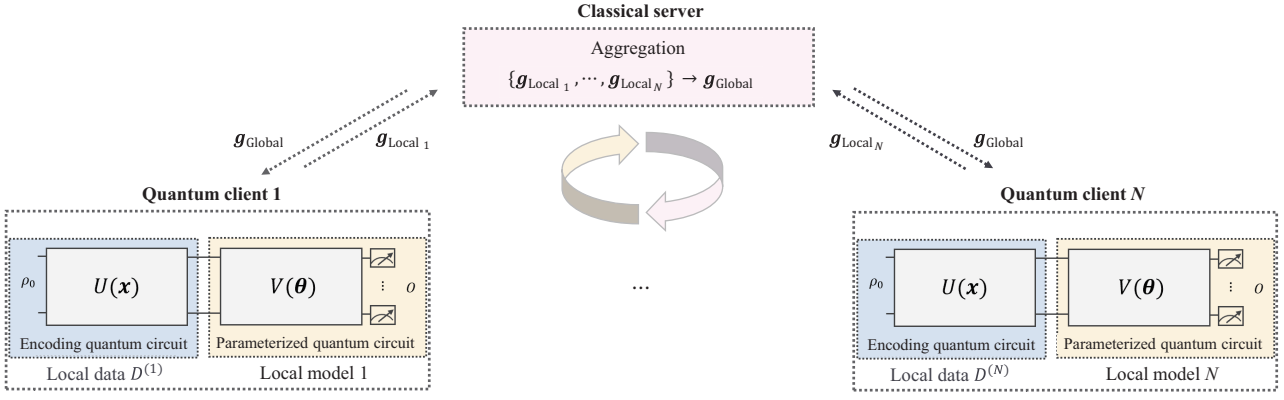

Figure 1 (Color online) Illustration of QFL framework. In each training iteration, each quantum client i encodes its local data $D^{(i)}$ onto quantum states and utilizes its parameterized quantum circuit $V(\boldsymbol{\theta})$ to calculate the local gradient $\boldsymbol{g}_{\text{Local}_i}$. These local gradients $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^N$ are then uploaded to the classical server. The server aggregates $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^N$ to obtain the global gradient $\boldsymbol{g}_{\text{Global}}$ and subsequently shares $\boldsymbol{g}_{\text{Global}}$ with all clients. Each client then updates its model parameters $\boldsymbol{\theta}$.

to optimize $\tilde{\mathcal{L}}(\boldsymbol{\theta})$. In detail, according to the chain rule, the derivatives $\{\partial\tilde{\mathcal{L}}(\boldsymbol{\theta})/\partial\theta_k\}_{k=1}^p$ can be calculated using $\tilde{E}(\boldsymbol{\theta})$ and the derivatives $\{\partial\tilde{E}(\boldsymbol{\theta})/\partial\theta_k\}_{k=1}^p$. Furthermore, due to the "parameter shift rule" [65, 66], $\partial\tilde{E}(\boldsymbol{\theta})/\partial\theta_k$ can be expressed as:

$$\partial\tilde{E}(\boldsymbol{\theta})/\partial\theta_k = [\tilde{E}_{\theta_k+}(\boldsymbol{\theta}) - \tilde{E}_{\theta_k-}(\boldsymbol{\theta})]/2, \tag{7}$$

where $\tilde{E}_{\theta_k\pm}(\boldsymbol{\theta})$ denotes $\tilde{E}(\boldsymbol{\theta})$ with θ_k being $\theta_k \pm \pi/2$. In summary, calculating $\{\partial\tilde{\mathcal{L}}(\boldsymbol{\theta})/\partial\theta_k\}_{k=1}^p$ can be transformed into calculating the following expectation values:

$$\{\tilde{E}(\boldsymbol{\theta}), \tilde{E}_{\theta_k\pm}(\boldsymbol{\theta})\}_{k=1}^p. \tag{8}$$

A notable characteristic of our QNN model is the encoding of classical data $\boldsymbol{x}$ onto the observable $O(\boldsymbol{x})$ instead of the quantum state $\rho(\boldsymbol{x})$. This characteristic distinguishes our model from conventional QNN models as described in sect. 2.1.

3.2 Framework of CC-QFL

The significance of the aforementioned characteristic becomes particularly evident in scenarios where N classical clients collaboratively train a QNN model, leveraging the quantum computing capabilities of the server, without the need to share their respective local data. In this context, each client i encodes its local data $D^{(i)}$ (eq. (2)) onto observables:

$$O^{(i)} = \{O(\boldsymbol{x}_j^{(i)})\}_{j=1}^{m_i}, \tag{9}$$

where $O(\boldsymbol{x}_j^{(i)}) = \sum_{h=1}^{l} c_h(\boldsymbol{x}_j^{(i)})O_h$. The initial state ρ_0 and the parameterized quantum circuit $V(\boldsymbol{\theta})$ are deployed on the server. Furthermore, in order to remove the necessity of quantum computing capabilities for clients, the collaborative training of the QNN model is assisted by the shadow tomography technique [58]. This technique constructs the classical representation of the given state to evaluate the expectation values of certain observables classically. A more detailed description of the shadow tomography technique is provided in Appendix.

The collaborative training of the QNN model consists of two important phases: (i) An acquisition phase where the server constructs a classical representation of the parameterized quantum states $\{(\rho(\boldsymbol{\theta}), \rho_{\theta_k\pm}(\boldsymbol{\theta})\}_{k=1}^p$, where $\rho(\boldsymbol{\theta}) = V(\boldsymbol{\theta})\rho_0 V^\dagger(\boldsymbol{\theta})$, and $\rho_{\theta_k\pm}(\boldsymbol{\theta})$ denotes $\rho(\boldsymbol{\theta})$ with θ_k being $\theta_k \pm \pi/2$. (ii) An evaluation phase where clients leverage this classical representation and data-encoding observables of all clients $\{O^{(i)}\}_{i=1}^N$ to calculate local gradients $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^N$ for updating the model parameters $\boldsymbol{\theta}$. The complete training procedure of CC-QFL is described in Algorithm 1.

During the acquisition phase, the server repeatedly executes a simple measurement procedure. This procedure involves randomly selecting a unitary operator W from a fixed ensemble $\mathcal{W}$ to rotate the n-qubit parameterized quantum state $\rho(\boldsymbol{\theta})$, followed by a computational-basis measurement. Upon receiving the n-bit measurement outcome $|b(\boldsymbol{\theta})\rangle : b(\boldsymbol{\theta}) \in \{0,1\}^n$, a completely classical post-processing step applies the inverted quantum channel $\mathcal{M}^{-1}$ to $W^\dagger|b(\boldsymbol{\theta})\rangle\langle b(\boldsymbol{\theta})|W$, where $\mathcal{M}^{-1}$ depends on the ensemble $\mathcal{W}$. Consequently, a classical snapshot $\hat{\rho}(\boldsymbol{\theta})$ of $\rho(\boldsymbol{\theta})$ is obtained from a single measurement, given by

$$\hat{\rho}(\boldsymbol{\theta}) = \mathcal{M}^{-1}(W^\dagger|b(\boldsymbol{\theta})\rangle\langle b(\boldsymbol{\theta})|W). \tag{10}$$

By repeating the aforementioned procedure M times, the classical shadow (classical representation) $S(\rho(\boldsymbol{\theta}); M)$ of $\rho(\boldsymbol{\theta})$

Algorithm 1 Training procedure of CC-QFL

Input: the initial state ρ_0, the parameterized quantum circuit $V(\boldsymbol{\theta})$ with $\boldsymbol{\theta} \in \mathbb{R}^p$, the number of measurements M, the learning rate η, the number of clients N, the number of data held by the i-th client m_i, the loss function $\tilde{\mathcal{L}}(\boldsymbol{\theta})$, and the number of iterations T

Output: the trained model parameters $\boldsymbol{\theta}^{(T)}$

Initialize: the random model parameters $\boldsymbol{\theta}^{(1)}$
while $1 \le t \le T$ **do**
  Construct the classical shadows $\mathcal{S}_{\text{Global}}^{(t)}$ (eq. (12)) of the parameterized quantum states $\{(\rho(\boldsymbol{\theta}^{(t)}), \rho_{\theta_k\pm}(\boldsymbol{\theta}^{(t)})\}_{k=1}^p$ by the server, where $\rho(\boldsymbol{\theta}^{(t)}) = V(\boldsymbol{\theta}^{(t)})\rho_0 V^\dagger(\boldsymbol{\theta}^{(t)})$, and $\rho_{\theta_k\pm}(\boldsymbol{\theta}^{(t)})$ denotes $\rho(\boldsymbol{\theta}^{(t)})$ with θ_k being $\theta_k \pm \pi/2$
  Transmit $\mathcal{S}_{\text{Global}}^{(t)}$ to clients
  while $1 \le i \le N$ **do**
    Calculate local gradient $\boldsymbol{g}_{\text{Local}_i}^{(t)}$ by the i-th client, leveraging $\mathcal{S}_{\text{Global}}^{(t)}$, data-encoding observables $O^{(i)}$ (eq. (9)), and the specific form of the loss function $\tilde{\mathcal{L}}(\boldsymbol{\theta}^{(t)})$
    Set $i \leftarrow i + 1$
  end while
  Upload local gradients of all clients $\{\boldsymbol{g}_{\text{Local}_i}^{(t)}\}_{i=1}^N$ to the server
  Aggregate $\{\boldsymbol{g}_{\text{Local}_i}^{(t)}\}_{i=1}^N$ as described in eq. (4) to obtain the global gradient $\boldsymbol{g}_{\text{Global}}^{(t)}$ by the server
  Update model parameters $\boldsymbol{\theta}^{(t)}$ by the server according to eq. (5)
  Set $t \leftarrow t + 1$
end while
return the trained model parameters $\boldsymbol{\theta}^{(T)}$

is obtained and defined as:

$$S(\rho(\boldsymbol{\theta}); M) = \{\hat{\rho}_j(\boldsymbol{\theta})\}_{j=1}^{M}, \tag{11}$$

where $\hat{\rho}_j(\boldsymbol{\theta}) = \mathcal{M}^{-1}(W_j^{\dagger}|b_j(\boldsymbol{\theta})\rangle\langle b_j(\boldsymbol{\theta})|W_j)$. The server ultimately constructs the classical shadows of parameterized quantum states $\{(\rho(\boldsymbol{\theta}), \rho_{\theta_k\pm}(\boldsymbol{\theta}))\}_{k=1}^{p}$. These classical shadows are expressed as:

$$\mathcal{S}_{\text{Global}} = \{(S(\rho(\boldsymbol{\theta}); M), S(\rho_{\theta_k\pm}(\boldsymbol{\theta}); M)\}_{k=1}^{p}. \tag{12}$$

Then, $\mathcal{S}_{\text{Global}}$ are transmitted to clients. Note that the server needs to explicitly indicate to the clients the corresponding parameter component for each classical shadow.

During the evaluation phase, each client i leverages $\mathcal{S}_{\text{Global}}$ and its data-encoding observables $O^{(i)}$ to evaluate the expectation values $\{\tilde{E}^{(i)}(\boldsymbol{\theta}), \tilde{E}^{(i)}_{\theta_k\pm}(\boldsymbol{\theta})\}_{k=1}^{p}$ (eq. (8)) using median of means estimation as described in Algorithm a1. This estimation method provides the approximations $\{\tilde{E}^{(i)}_{\text{Approx}}(\boldsymbol{\theta}), \tilde{E}^{(i)}_{\text{Approx};\theta_k\pm}(\boldsymbol{\theta})\}_{k=1}^{p}$ of $\{\tilde{E}^{(i)}(\boldsymbol{\theta}), \tilde{E}^{(i)}_{\theta_k\pm}(\boldsymbol{\theta})\}_{k=1}^{p}$, which are used to calculate the corresponding local gradient $\boldsymbol{g}_{\text{Local}_i}$ based on the specific form of the loss function $\tilde{\mathcal{L}}(\boldsymbol{\theta})$.

Subsequently, the local gradients of all clients $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^{N}$ are uploaded to the server. The server performs an aggregation process using $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^{N}$ to obtain the aggregated global gradient $\boldsymbol{g}_{\text{Global}}$ as described in eq. (4). Finally, the server updates its model parameters $\boldsymbol{\theta}$ according to eq. (5). The process of constructing classical shadows, calculating local gradients, aggregating these gradients, and updating the model parameters on the server is repeated iteratively until a predefined number of iterations is reached.

In summary, our framework allows multiple classical clients to collaborate in training a QNN model. By encoding the local data of each client onto observables and employing the shadow tomography technique, the clients can actively participate in the training process without the need to share their local data while leveraging the quantum computing capabilities of the server. The complete framework of CC-QFL is depicted in Figure 2.

3.3 Analysis of complexity

In our framework, the central step involves the client utilizing a classical shadow of the n-qubit parameterized quantum state $\rho(\boldsymbol{\theta})$, i.e., $S(\rho(\boldsymbol{\theta}); M) = \{\hat{\rho}_j(\boldsymbol{\theta})\}_{j=1}^{M}$, to evaluate the expectation value $\text{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ on a classical computer. Here, M represents the size of the classical shadow, and $O(\boldsymbol{x}) = \sum_{i=1}^{d} x_i O_i$ encodes d-dimensional classical data $\boldsymbol{x} = [x_i]_{i=1}^{d}$.

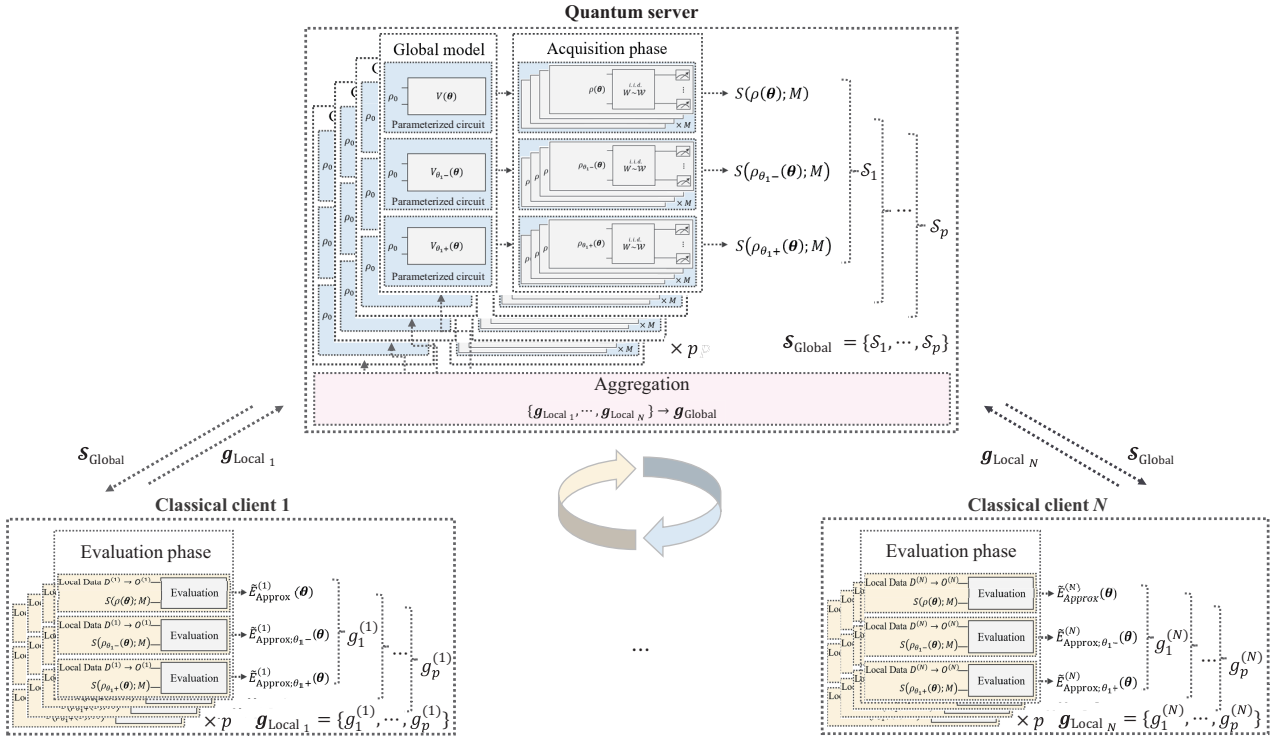

Figure 2 (Color online) Illustration of CC-QFL framework. Each classical client i encodes its local data $D^{(i)}$ onto observables $O^{(i)}$, while the initial state ρ_0 and the parameterized quantum circuit $V(\boldsymbol{\theta})$ are deployed on the quantum server. In each training iteration, the server constructs the classical shadows $\mathcal{S}_{\text{Global}}$ of the parameterized quantum states $\{(\rho(\boldsymbol{\theta}), \rho_{\theta_k\pm}(\boldsymbol{\theta}))\}_{k=1}^{p}$. $\mathcal{S}_{\text{Global}}$ are then transmitted to the clients. Each client i leverages $\mathcal{S}_{\text{Global}}$ and its data-encoding observables $O^{(i)}$ to calculate the corresponding local gradient $\boldsymbol{g}_{\text{Local}_i}$. The local gradients of all clients $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^{N}$ are uploaded to the server. The server performs an aggregation process using $\{\boldsymbol{g}_{\text{Local}_i}\}_{i=1}^{N}$ to obtain the aggregated global gradient $\boldsymbol{g}_{\text{Global}}$. Finally, the server updates its model parameters $\boldsymbol{\theta}$.

Based on the detailed setting of shadow tomography technique in our framework, including the ensemble $\mathcal{W}$ used to construct the classical shadow and the specific form of Hermitian matrix O_i, we elaborately analyze the sample complexity, i.e., the size of the classical shadow M, and the computational complexity of evaluating $\mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ up to an additive error $\epsilon' = O(d\epsilon)$ as follows.

3.3.1 *Sample complexity*

Here, we summarize the main results given in Theorem S1 and Lemma S1 of ref. [58] as follows.

Fact 1 Fix the ensemble $\mathcal{W}$ used to construct the classical shadow of the n-qubit parameterized quantum state $\rho(\boldsymbol{\theta})$. The classical shadow of size

$$M = O(\log(d)\max_{i\in[d]}\|O_i - (\mathrm{Tr}(O_i)/2^n)\mathbb{I}\|^2_{\mathrm{shadow}}/\epsilon^2) \tag{13}$$

suffices to evaluate d expectation values $\mathrm{Tr}(\rho(\boldsymbol{\theta})O_1), \cdots, \mathrm{Tr}(\rho(\boldsymbol{\theta})O_d)$ up to an additive error ϵ. Here, each O_i is an arbitrary $2^n\times 2^n$ Hermitian matrix, and the shadow norm $\|\cdot\|^2_{\mathrm{shadow}}$ is defined by

$$\max_{\gamma}\mathbb{E}_{W\sim\mathcal{W}}\sum_{b(\boldsymbol{\theta})\in\{0,1\}^n}\langle b(\boldsymbol{\theta})|W\gamma W^\dagger|b(\boldsymbol{\theta})\rangle\langle b(\boldsymbol{\theta})|W\mathcal{M}^{-1}(\cdot)W^\dagger|b(\boldsymbol{\theta})\rangle^2, \tag{14}$$

where γ represents an arbitrary n-qubit density matrix, and $\mathcal{M}^{-1}(\cdot)$ denotes an inverted quantum channel that depends on the ensemble $\mathcal{W}$.

From eq. (13), it is evident that the sample complexity is determined by the norm $\|O_i - (\mathrm{Tr}(O_i)/2^n)\mathbb{I}\|^2_{\mathrm{shadow}}$ for $i\in[d]$. The shadow norm $\|\cdot\|^2_{\mathrm{shadow}}$ as described in eq. (14) depends on the ensemble $\mathcal{W}$ and the specific form of O_i. In our framework, $\mathcal{W} = \mathrm{Cl}(2)^{\otimes n}$, where $\mathrm{Cl}(2)$ represents the single-qubit Clifford group generated by Hadamard and phase gates. In addition, each O_i generally represents a k-local Pauli observable, with the specific form of

$$O_i = P_{i,1}\otimes\cdots\otimes P_{i,n}, \tag{15}$$

where $\{P_{i,s_1},\cdots,P_{i,s_k}\}\in\{\sigma^x,\sigma^y,\sigma^z\}$, and $P_{i,s} = \mathbb{I}$ for $s\in\{1,\cdots,n\}\setminus\{s_1,\cdots,s_k\}$. As a result, the norm $\|O_i - (\mathrm{Tr}(O_i)/2^n)\mathbb{I}\|^2_{\mathrm{shadow}} = 3^k$. The detailed derivation process can be found in Lemma S3 of ref. [58]. Therefore, to evaluate $\mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ up to an additive error $\epsilon' = O(d\epsilon)$, the sample complexity is $M = O(\log(d)3^k/\epsilon^2)$.

3.3.2 *Computational complexity*

The client utilizes the classical shadow $S(\rho(\boldsymbol{\theta});M) = \{\hat{\rho}_j(\boldsymbol{\theta})\}_{j=1}^M$ of the n-qubit parameterized quantum state $\rho(\boldsymbol{\theta})$ to evaluate the expectation value $\mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$. Essentially, the client needs to evaluate $\mathrm{Tr}(\rho(\boldsymbol{\theta})O_i)$ for $i\in[d]$, where the evaluation of each $\mathrm{Tr}(\rho(\boldsymbol{\theta})O_i)$ requires values of $\{\mathrm{Tr}(\hat{\rho}_1(\boldsymbol{\theta})O_i),\cdots,\mathrm{Tr}(\hat{\rho}_M(\boldsymbol{\theta})O_i)\}$. Given that the sample complexity M of evaluating $\mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ up to an additive error $\epsilon' = O(d\epsilon)$ has been analyzed in sect. 3.3.1, we only focus on the analysis of the computational complexity involved in calculating each $\mathrm{Tr}(\hat{\rho}_j(\boldsymbol{\theta})O_i)$.

In the context of our framework, the ensemble $\mathcal{W} = \mathrm{Cl}(2)^{\otimes n}$, resulting in the corresponding inverted quantum channel $\mathcal{M}^{-1}$ defined by

$$\mathcal{M}^{-1}(X_1\otimes\cdots\otimes X_n) = \otimes_{s=1}^n(3X_s - \mathbb{I}), \tag{16}$$

where each X_s is an arbitrary 2×2 Hermitian matrix. Subsequently, the classical snapshot $\hat{\rho}_j(\boldsymbol{\theta})$ as described in eq. (10) has a more concrete representation, given by

$$\hat{\rho}_j(\boldsymbol{\theta}) = \otimes_{s=1}^n(3W^\dagger_{j,s}|b_{j,s}(\boldsymbol{\theta})\rangle\langle b_{j,s}(\boldsymbol{\theta})|W_{j,s} - \mathbb{I}), \tag{17}$$

where $W_{j,s}\in\mathrm{Cl}(2)$, and $|b_{j,s}(\boldsymbol{\theta})\rangle : b_{j,s}(\boldsymbol{\theta})\in\{0,1\}$. The detailed derivation process can be found in Proposition S2 of ref. [58]. Furthermore, each Hermitian matrix O_i is a k-local Pauli observable as described in eq. (15). As a result, the computational complexity of calculating each $\mathrm{Tr}(\hat{\rho}_j(\boldsymbol{\theta})O_i)$ is $O(n)$ by using the Gottesman-Knill theorem [67].

In conclusion, the client's computational complexity of evaluating $\mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ up to an additive error $\epsilon' = O(d\epsilon)$ is $O(nd\log(d)3^k/\epsilon^2)$, which implies the computational complexity scales polynomially with the system size n and the data dimension d. In other words, our framework is also suitable for practical applications when handling high-dimensional datasets. Table 1 briefly summarizes the detailed setting of shadow tomography technique in our framework and the corresponding analytical results.

4 Experiment results

We perform numerical simulations of our CC-QFL framework using the TensorFlow Quantum [68] simulation platform. These simulations employ handwritten digit images from the MNIST dataset, which comprises 70000 images. The experimental setting is as follows.

Table 1 The detailed setting of shadow tomography technique in our framework and the corresponding analytical results

Symbol	Definition
Ensemble $\mathcal{W}$	$\mathcal{W} = \mathrm{Cl}(2)^{\otimes n}$
Hermitian matrix O_i	O_i is a k-local Pauli observable
Classical snapshot $\hat{\rho}_j(\boldsymbol{\theta})$	$\hat{\rho}_j(\boldsymbol{\theta}) = \otimes_{s=1}^n(3W^\dagger_{j,s}\|b_{j,s}(\boldsymbol{\theta})\rangle\langle b_{j,s}(\boldsymbol{\theta})\|W_{j,s} - \mathbb{I})$
Additive error ϵ'	$\epsilon' = O(d\epsilon)$
Complexity	**Result**
Sample complexity	$O(\log(d)3^k/\epsilon^2)$
Computational complexity	$O(nd\log(d)3^k/\epsilon^2)$

First, we preprocess the handwritten digit images. Each image is flattened into a one-dimensional vector of length 784, representing its 28 × 28 pixels. Due to the computational limitations of the simulation, we normalize the image and perform dimensionality reduction using the principal component analysis algorithm [69]. This reduces the vector length to 8. The resulting image is denoted as $\boldsymbol{x} = [x_i]_{i=1}^{8}$. The classical clients encode $\boldsymbol{x}$ onto the 8-qubit observable $O(\boldsymbol{x}) = \sum_{i=1}^{8} x_i \sigma_i^z$, where σ_i^z is the Pauli Z operator acting on the i-th qubit. The parameterized quantum circuit is a 5-layer hardware efficient ansatz [70], which is deployed on the quantum server. To construct the corresponding classical shadows, the server measures each qubit independently in a random Pauli basis.

In the subsequent sections, we will present the performance of single-client and multi-client CC-QFL frameworks in a binary classification task. To achieve this, we employ the cross-entropy loss function.

4.1 Performance of single-client CC-QFL

In the single-client CC-QFL framework, a classical client utilizes the quantum computing capabilities of the server to train its classifier model without the need to upload its local data to the server. The classification task involves accurately classifying handwritten digit images of the numbers "3" and "6" from the MNIST dataset. The client has a total of 2640 images, with 2000 images allocated for the training set and the remaining 640 images reserved as the test set. The specific parameter settings of the experiment can be found in Table 2.

The training and testing results of our numerical simulations are plotted in Figure 3(a). During the first 20 epochs, it is observed that the training loss exhibits a rapid decrease, while both the training accuracy and test accuracy show a rapid increase following a slight decline. Upon reaching convergence, the training accuracy and test accuracy reached 98.69% and 98.24%, respectively. These results demonstrate that the single-client CC-QFL achieves excellent numerical performance in the binary classification task on the MNIST dataset.

Table 2 Parameter settings for single-client and multi-client CC-QFL

Parameter	Single-client	Multi-client
Num of clients	1	3
Num of classes	2 ("3" or "6")	2 ("3" or "6")
Num of qubits	8	8
Circuit depth	5	5
Data distribution	–	non-i.i.d
Training set (per client)	2000	700
Test set (per client)	640	220
Optimizer	Adam	Adam
Learning rate	0.003	0.003
Loss function	cross-entropy	cross-entropy

4.2 Performance of multi-client CC-QFL

Compared with the single-client scenario, our primary focus lies in the performance of the multi-client CC-QFL, where N classical clients leverage the quantum computing capabilities of the server to collaboratively train the classifier model, without the need to upload their respective local data to the server. In this scenario, with N=3, the classification task focuses on accurately classifying handwritten digit images of the numbers "3" and "6" from the MNIST dataset. Each client is assigned a total of 920 images, with 700 images allocated for the training set and the remaining 220 images reserved for the test set. Considering the uniqueness of each client in real-world scenarios, the training data should satisfy non-i.i.d. To address this, we utilize Leaf [71] to process the MNIST dataset, ensuring that each client has a distinct proportion of handwritten digit labels. To visualize the non-i.i.d nature of the training data, the corresponding heterogeneity

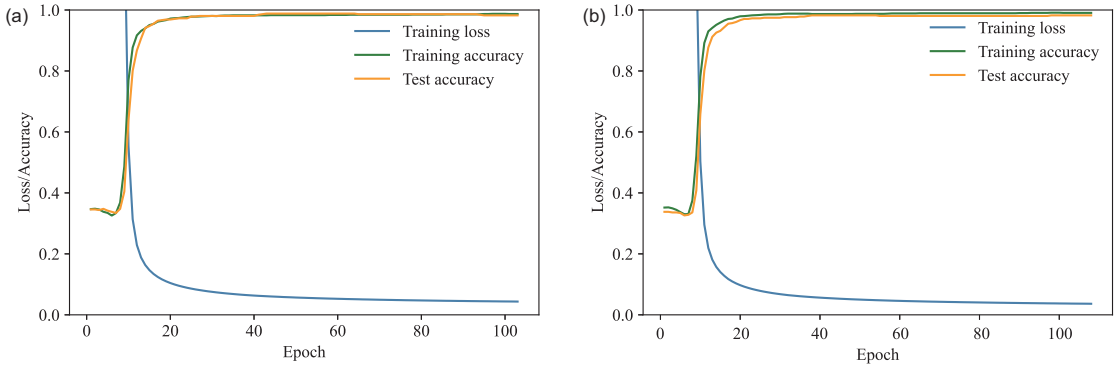

Figure 3 (Color online) Performance of CC-QFL in the binary classification task on the MNIST dataset. (a) The numerical results of the single-client scenario. (b) The numerical results of the multi-client scenario (N = 3). Here, the training loss, training accuracy, and test accuracy are represented by solid lines in blue, green, and orange, respectively.

is depicted in Figure 4. The specific parameter settings of the experiment can be found in Table 2.

The training and testing results obtained from our numerical simulations are illustrated in Figure 3(b). The numerical results reveal a significant decrease in training loss during the first 20 epochs. Furthermore, both the training accuracy and test accuracy demonstrate a notable increase around the 10th epoch. At convergence, the training accuracy achieves 98.69%, and the test accuracy reaches 98.05%, which is comparable to the numerical results achieved in the single-client scenario. The excellent numerical performance validates the feasibility and effectiveness of multi-client CC-QFL.

5 Conclusions

In this paper, we present a novel QFL framework designed to address scenarios involving limited quantum computing resources. This framework allows classical clients to collaboratively train a QML model while ensuring the privacy of their data. In contrast to the conventional QFL framework, our framework involves deploying the QML model on the server instead of the clients and encoding the classical data onto observables rather than quantum states. Furthermore, the shadow tomography technique assists in the training of the QML model and eliminates the need for clients to possess quantum computing capabilities in the conventional QFL framework. Our framework extends the potential applications of QFL, particularly in situations where quantum computing resources are scarce.

Furthermore, there remain several intriguing aspects that require further investigation. One aspect of the investigation involves exploring different approaches for encoding classical data onto observables and analyzing their impact on the effectiveness of the QML model training to derive more appropriate data encoding approaches. An additional investigation aspect involves mitigating the potential leakage of client data to the server through gradient inversion attacks [72, 73]. In this regard, the adoption of secure gradient aggregation strategies is crucial. Our framework can naturally integrate with existing techniques such as homomorphic encryption [74] or differential privacy [75,76] to enhance client data security. Moreover, the exploration of novel strategies holds significant value.

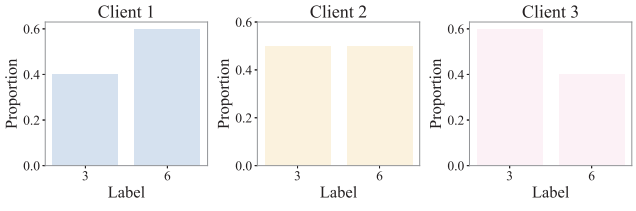

Figure 4 (Color online) Heterogeneity of the training data. We illustrate the proportion of handwritten digit labels ("3" and "6") on each client, where each client contains a total of 700 images.

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62371069, 62272056, and 62372048), Beijing Natural Science Foundation (Grant No. 4222031), and China Scholarship Council (Grant No. 202006470011).

Conflict of interest The authors declare that they have no conflict of interest.

1 K. Simonyan, and A. Zisserman, in *Very Deep Convolutional Networks for Large-Scale Image Recognition: 3rd International Conference on Learning Representations* (ICLR, San Diego, 2015).
2 C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, in *Going Deeper with Convolutions: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR, Boston, 2015).
3 A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, Comput. Intel. Neurosci. **2018**, 1 (2018).
4 L. Sutskever, O. Vinyals, and Q, Le, in *Sequence to Sequence Learning with Neural Networks: Advances in Neural Information Processing Systems* (NIPS, 2014).
5 D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Nature **529**, 484 (2016).
6 B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, in *Communication-Efficient Learning of Deep Networks from Decentralized Data: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (AISTATS, Valencia, 2017).
7 A. W. Harrow, and A. Montanaro, Nature **549**, 203 (2017), arXiv: 1809.07442.
8 P. W. Shor, SIAM Rev. **41**, 303 (1999).
9 L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997), arXiv: quant-ph/9706033.
10 J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195 (2017), arXiv: 1611.09347.
11 V. Dunjko, and H. J. Briegel, Rep. Prog. Phys. **81**, 074001 (2018).
12 K. Schütt, S. Chmiela, V Lilienfeld, O. Anatole, A. Tkatchenko, K. Tsuda, and K. Müller, in *Machine Learning Meets Quantum Physics: Lecture Notes in Physics* (LNP, 2020).
13 A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009), arXiv: 0811.3171.
14 P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd, Phys. Rev. A **97**, 012327 (2018).
15 R. Somma, M. A. Childs, and R. Kothari, in *Quantum Linear Systems Algorithm with Exponentially Improved Dependence On Precision: APS March Meeting Abstracts* (APS March Meeting, College Park, 2016).
16 N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109**, 050505 (2012), arXiv: 1204.5242.
17 P. Rebentrost, M. Schuld, L. Wossnig, F. Petruccione, and S. Lloyd, New J. Phys. **21**, 073023 (2019).
18 J. M. Liang, S. J. Wei, and S. M. Fei, Sci. China-Phys. Mech. Astron. **65**, 250313 (2022), arXiv: 2204.07284.
19 P. Gao, K. Li, S. Wei, and G. L. Long, Sci. China-Phys. Mech. Astron. **64**, 100311 (2021).
20 S. Lloyd, M. Mohseni, and P. Rebentrost, Nat. Phys. **10**, 631 (2014), arXiv: 1307.0401.
21 F. Brandao, and K. Svore, in *Quantum Speed-ups for Solving Semidefi-*

nite Programs: 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS, Berkeley, 2017).
22 P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014), arXiv: 1307.0471.
23 Z. Ye, L. Li, H. Situ, and Y. Wang, Sci. China Inf. Sci. **63**, 189501 (2020).
24 M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Nat. Rev. Phys. **3**, 625 (2021).
25 K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W. K. Mok, S. Sim, L. C. Kwek, and A. Aspuru-Guzik, Rev. Mod. Phys. **94**, 015004 (2022), arXiv: 2101.08448.
26 Y. Song, Y. Wu, S. Qin, Q. Wen, J. B. Wang, and F. Gao, arXiv: 2310.06270.
27 H. L. Huang, X. Y. Xu, C. Guo, G. Tian, S. J. Wei, X. Sun, W. S. Bao, and G. L. Long, Sci. China-Phys. Mech. Astron. **66**, 250302 (2023), arXiv: 2211.08737.
28 E. Farhi, and H. Neven, arXiv: 1802.06002.
29 Z. Abohashima, M. Elhosen, E. H. Houssein, and W. M. Mohamed, arXiv: 2006.12270.
30 W. Li, and D. L. Deng, Sci. China-Phys. Mech. Astron. **65**, 220301 (2022), arXiv: 2108.13421.
31 D. L. Deng, Sci. China-Phys. Mech. Astron. **64**, 100331 (2021).
32 W. Ren, W. Li, S. Xu, K. Wang, W. Jiang, F. Jin, X. Zhu, J. Chen, Z. Song, P. Zhang, H. Dong, X. Zhang, J. Deng, Y. Gao, C. Zhang, Y. Wu, B. Zhang, Q. Guo, H. Li, Z. Wang, J. Biamonte, C. Song, D. L. Deng, and H. Wang, Nat. Comput. Sci. **2**, 711 (2022).
33 S. J. Wei, Y. H. Chen, Z. R. Zhou, and G. L. Long, AAPPS Bull. **32**, 2 (2022).
34 W. Li, Z. Lu, and D. L. Deng, SciPost Phys. Lect. Notes **2022**, 61 (2022).
35 Z. Liu, P. X. Shen, W. Li, L. M. Duan, and D. L. Deng, Quantum Sci. Technol. **8**, 015016 (2023).
36 X. Hou, G. Zhou, Q. Li, S. Jin, and X. Wang, Sci. China-Phys. Mech. Astron. **66**, 270362 (2023), arXiv: 2211.11228.
37 S. Y. C. Chen, S. Yoo, and Y. L. L. Fang, in *Quantum Long Short-term Memory: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP, Singapore, 2022).
38 O. Kyriienko, A. E. Paine, and V. E. Elfving, Phys. Rev. A **103**, 052416 (2021), arXiv: 2011.10395.
39 Y. Wu, B. Wu, J. Wang, and X. Yuan, Quantum **7**, 981 (2023).
40 C. Zoufal, A. Lucchi, and S. Woerner, npj Quantum Inf. **5**, 103 (2019), arXiv: 1904.00043.
41 K. Nakaji, and N. Yamamoto, Sci. Rep. **11**, 19649 (2021), arXiv: 2010.13727.
42 H. Situ, Z. He, Y. Wang, L. Li, and S. Zheng, Inf. Sci. **538**, 193 (2020).
43 S. Y. C. Chen, C. H. H. Yang, J. Qi, P. Y. Chen, X. Ma, and H. S. Goan, IEEE Access **8**, 141007 (2020).
44 O. Lockwood, and M. Si, in *Reinforcement Learning With Quantum Variational Circuit: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (AIIDE, Salt Lake City, 2020).
45 Q. Xia, and Q. Li, in *Quantumfed: A Federated Learning Framework for Collaborative Quantum Training: 2021 IEEE Global Communications Conference* (GLOBECOM, Madrid, 2021).
46 M. Chehimi, and W. Saad, in *Quantum Federated Learning with Quantum Data: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP, Singapore, 2022).
47 R. Huang, X. Tan, and Q. Xu, IEEE J. Sel. Top. Quantum Electron. **28**, 1 (2022).
48 Q. Xia, Z. Tao, and Q. Li, in *Defending Against Byzantine Attacks in Quantum Federated Learning: 2021 17th International Conference on Mobility, Sensing and Networking* (MSN, Exeter, 2021).
49 W. Yamany, N. Moustafa, and B. Turnbull, IEEE Trans. Intell. Transp. Syst. **24**, 893 (2023).
50 W. Li, S. Lu, and D. L. Deng, Sci. China-Phys. Mech. Astron. **64**, 100312 (2021), arXiv: 2103.08403.
51 Y. Zhang, C. Zhang, C. Zhang, L. Fan, B. Zeng, and Q. Yang, arXiv: 2207.07444.
52 C. Li, N. Kumar, Z. Song, S. Chakrabarti, and M. Pistoia, arXiv: 2312.04447.
53 A. S. Bhatia, S. Kais, and M. A. Alam, Quantum Sci. Technol. **8**, 045032 (2023).
54 H. Zhao, Quantum Mach. Intell. **5**, 3 (2023).
55 W. J. Yun, J. P. Kim, S. Jung, J. Park, M. Bennis, and J. Kim, arXiv: 2207.10221.
56 Y. B. Sheng, and L. Zhou, Sci. Bull. **62**, 1025 (2017).
57 A. Broadbent, J. Fitzsimons, and E. Kashefi, in *Universal Blind Quantum Computation: 2009 50th Annual IEEE Symposium on Foundations of Computer Science* (FOCS, Atlanta, 2009).
58 H. Y. Huang, R. Kueng, and J. Preskill, Nat. Phys. **16**, 1050 (2020), arXiv: 2002.08953.
59 H. Y. Huang, R. Kueng, and J. Preskill, Phys. Rev. Lett. **127**, 030503 (2021), arXiv: 2103.07510.
60 H. C. Nguyen, J. L. Bönsel, J. Steinberg, and O. Gühne, Phys. Rev. Lett. **129**, 220502 (2022).
61 C. Hadfield, S. Bravyi, R. Raymond, and A. Mezzacapo, Commun. Math. Phys. **391**, 951 (2022).
62 B. Wu, J. Sun, Q. Huang, and X. Yuan, Quantum **7**, 896 (2023).
63 Y. Wu, and J. B. Wang, Quantum Sci. Technol. **7**, 025006 (2022), arXiv: 2109.10486.
64 D. P. Kingma, and J. Ba, arXiv: 1412.6980.
65 K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Phys. Rev. A **98**, 032309 (2018), arXiv: 1803.00745.
66 M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Phys. Rev. A **99**, 032331 (2019), arXiv: 1811.11184.
67 D. Gottesman, *Stabilizer Codes and Quantum Error Correction* (California Institute of Technology, Pasadena, 1997).
68 M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, R. Halavati, M. Y. Niu, and A. Zlokapa, arXiv: 2003.02989.
69 A. Maćkiewicz, and W. Ratajczak, Comput. Geosci. **19**, 303 (1993).
70 A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature **549**, 242 (2017), arXiv: 1704.05018.
71 S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, arXiv: 1812.01097.
72 L. Zhu, Z. Liu, and S. Han, in *Deep Leakage from Gradients: Advances in Neural Information Processing Systems 32* (NeurIPS, Vancouver, 2019).
73 J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, in *Inverting Gradients-how Easy is it to Break Privacy in Federated Learning? Advances in Neural Information Processing Systems 33* (NeurIPS, Vancouver, 2020).
74 C. Gentry, in *Fully Homomorphic Encryption Using Ideal Lattices: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (STOC, Bethesda, 2009).
75 C. Dwork, in *Differential Privacy: A survey of results: International Conference on Theory and Applications of Models of Computation* (TAMC, Xi'an, 2008).
76 C. Dwork, in *Differential Privacy: International Colloquium on Automata, Languages, and Programming* (ICALP, Rennes, 2006).
77 M. R. Jerrum, L. G. Valiant, and V. V. Vazirani, Theor. Comput. Sci. **43**, 169 (1986).

Appendix Shadow tomography technique

The shadow tomography technique, proposed by Huang et al. [58], is a method used to extract meaningful information from an n-qubit unknown quantum state ρ. One important ap-

Algorithm a1 Shadow tomography technique for evaluating expectation values

Stage 1: Construct classical shadow

Input: the n-qubits unknown state ρ, the set of unitary operators $\mathcal{W}$, and the number of measurements M

Output: the classical shadow $S(\rho; M) = \{\hat{\rho}_1, \cdots, \hat{\rho}_M\}$

for $j = 1$ to M **do**
  Randomly select a unitary operator W_j from $\mathcal{W}$ and apply W_j to ρ
  Measure in the computational basis and obtain $b_j \in \{0,1\}^n$
  Apply the inverted quantum channel $\mathcal{M}^{-1}$ to $W_j^\dagger|b_j\rangle\langle b_j|W_j$ and obtain $\hat{\rho}_j = \mathcal{M}^{-1}(W_j^\dagger|b_j\rangle\langle b_j|W_j)$, where $\mathcal{M}^{-1}$ depends on the set $\mathcal{W}$
end for
return the classical shadow $S(\rho; M) = \{\hat{\rho}_1, \cdots, \hat{\rho}_M\}$

Stage 2: Evaluate expectation values

Input: the set of observables $\{O_i\}_{i=1}^d$, and the classical shadow $S(\rho; M) = \{\hat{\rho}_1, \cdots, \hat{\rho}_M\}$

Output: the approximations $\{\tilde{o}_i\}_{i=1}^d$ of the expectation values $\{o_i\}_{i=1}^d$, where $o_i = \mathrm{Tr}(O_i\rho)$

Divide $S(\rho; M)$ into M_2 equally sized chunks $\{\hat{\rho}_1^{(1)}, \cdots, \hat{\rho}_{M_1}^{(1)}\}, \cdots, \{\hat{\rho}_1^{(M_2)}, \cdots, \hat{\rho}_{M_1}^{(M_2)}\}$, where $M = M_1 \times M_2$
for $i = 1$ to d **do**
  for $h = 1$ to M_2 **do**
    Evaluate o_i using the h-th chunk and obtain the h-th approximation $\tilde{o}_i^{(h)} = \frac{1}{M_1}\sum_{l=1}^{M_1} \mathrm{Tr}(O_i\hat{\rho}_l^{(h)})$
  end for
  Calculate the median of the M_2 approximations and obtain the final approximation $\tilde{o}_i = \mathrm{median}\left\{\tilde{o}_i^{(1)}, \cdots, \tilde{o}_i^{(M_2)}\right\}$
end for
return the approximations $\{\tilde{o}_i\}_{i=1}^d$

plication of this technique is to construct the classical shadow of ρ in order to evaluate the expectation values $\{o_i\}_{i=1}^d$ of certain observables $\{O_i\}_{i=1}^d$ classically. Here, $o_i = \mathrm{Tr}(O_i\rho)$.

Specifically, a simple measurement procedure is repeatedly executed to construct the classical shadow of ρ. This procedure involves randomly selecting a unitary operator W from a fixed ensemble $\mathcal{W}$ to rotate ρ, followed by a computational-basis measurement. Upon receiving the n-bit measurement outcome $|b\rangle : b \in \{0,1\}^n$, the classical memory can efficiently store a classical description of $W^\dagger|b\rangle\langle b|W$, according to the Gottesman-Knill theorem [67]. It is instructive to consider the average mapping from ρ to $W^\dagger|b\rangle\langle b|W$ as a quantum channel given by

$$\mathcal{M}(\rho) = \mathbb{E}[W^\dagger|b\rangle\langle b|W], \tag{a1}$$

where $\mathcal{M}$ depends on the ensemble $\mathcal{W}$. Consequently, a completely classical post-processing step applies the inverted quantum channel $\mathcal{M}^{-1}$ to the measurement outcome $W^\dagger|b\rangle\langle b|W$. Subsequently, a classical snapshot $\hat{\rho}$ of ρ is obtained from a single measurement, given by

$$\hat{\rho} = \mathcal{M}^{-1}(W^\dagger|b\rangle\langle b|W). \tag{a2}$$

By repeating the aforementioned procedure M times, the classical shadow of ρ is obtained and defined as:

$$S(\rho; M) = \{\hat{\rho}_j\}_{j=1}^M, \tag{a3}$$

where $\hat{\rho}_j = \mathcal{M}^{-1}(W_j^\dagger|b_j\rangle\langle b_j|W_j)$. $S(\rho; M)$, with a sufficient size M, can efficiently evaluate $\{o_i\}_{i=1}^d$. Specifically, $S(\rho; M)$ is divided into equally sized chunks, and multiple independent estimators are constructed. Subsequently, the approximations $\{\tilde{o}_i\}_{i=1}^d$ of $\{o_i\}_{i=1}^d$ are obtained by using the median of means estimation [77]. The detailed procedure is summarized in Algorithm a1.

The described procedure can be applied to various distributions of random unitary operators. One prominent example is tensor products of random single-qubit Clifford circuits. In this example, where each qubit is independently measured in a random Pauli basis, it can also be referred to as random Pauli measurements. The resulting classical shadow can be efficiently stored in classical memory using the stabilizer formalism. Furthermore, an interesting result demonstrates that $O(\log(d)3^k/\epsilon^2)$ random Pauli measurements of ρ suffice to evaluate H bounded observables $\{O_i\}_{i=1}^d$ that are tensor products of k single-qubit observables. This evaluation guarantees $|\tilde{o}_i - o_i| \leq \epsilon$ with high probability for any $i \in [d]$.