QUANTUM FEDERATED LEARNING WITH QUANTUM DATA

Mahdi Chehimi and Walid Saad
Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, USA,
Emails: {mahdic,walids}@vt.edu

ABSTRACT

Quantum machine learning (QML) has emerged as a promising field that leans on the developments in quantum computing to explore complex machine learning problems. Recently, some QML models were proposed for performing classification tasks, however, they rely on centralized solutions that cannot scale well for distributed quantum networks. Hence, it is apropos to consider more practical quantum federated learning (QFL) solutions tailored towards emerging quantum networks to allow for distributing quantum learning. This paper proposes the first fully quantum federated learning framework that can operate over purely quantum data. First, the proposed framework generates the first quantum federated dataset in literature. Then, quantum clients share the learning of quantum circuit parameters in a decentralized manner. Extensive experiments are conducted to evaluate and validate the effectiveness of the proposed QFL solution, which is the first implementation combining Google's TensorFlow Federated and TensorFlow Quantum.

***Index Terms*—** Quantum machine learning (QML), federated learning (FL)

1. INTRODUCTION

Recent advances in quantum computing revolutionized the way computations are done and allowed for solving complex large-scale problems in an extremely fast manner [1]. In particular, the ability of quantum computers to handle the exponential growth in the dimensions of data more efficiently than classical computers lead to the blossoming of the *quantum machine learning* (QML) field [2]. Advances in this field led to multiple successful hybrid quantum-classical ML models that achieve a superior performance compared to classical models. However, such models fall short when dealing with the highly complex purely quantum data coming from quantum many-body systems [3]. This is due to the complex nature and exponentially large Hilbert spaces associated with such systems, which makes their theoretical analysis intractable.

Thus, *purely quantum ML* models that can address these challenges were developed, such as *quantum neural networks* (QNNs) [4]. In particular, QNNs are quantum circuits with tunable parameters that can be "learned" similar to classical models. There exist several QNN architectures that can be used to perform classification on near-term quantum computers, such as *quantum convolutional neural networks* (QCNNs) [5]. This also includes quantum recurrent neural networks [6] that deal with sequence modeling and speech synthesis.

Along with the advances in QML, integrating quantum computers in future communication networks is necessary to handle the challenges caused by the rapidly growing volume of data. This integration is of particular importance to the fields of computer vision, natural language processing, and acoustic and speech analysis, which require huge computations. However, due to the fragile nature of the carriers of quantum data, i.e., qubits, *it is much more difficult to transfer the quantum data of QML models using quantum channels in an efficient manner, compared to classical data*. Thus, there is a natural need for distributed learning solutions such as federated learning (FL) [7, 8] in those emerging quantum networks (QNs). FL algorithms would allow for the exchange of the quantum model parameters, rather than the quantum data itself and, thus, minimize the losses incorporated in the distributed quantum learning process. However, remarkably, to the best of our knowledge, no prior work has proposed a thorough, comprehensive framework for implementing FL over purely quantum data in QNs, and there does not exist any quantum federated dataset in the literature. In addition, existing QML models rely on centralized solutions that cannot scale well for future, large-scale, and distributed QNs.

Contributions. The main contribution of this paper is to develop a novel comprehensive *quantum federated learning* (QFL) framework that allows for distributing quantum learning collaboratively between clients with purely quantum data while leveraging existing classical communication infrastructure. In particular, we propose the first purely quantum QFL framework for clients with quantum computing capabilities that employ QCNN models to perform a classification task. Moreover, the proposed QFL framework integrates quantum clients with the classical wireless infrastructure without relying on quantum channels. Thus, the proposed framework is amenable to practical implementations in existing communication networks. On top of that, we generate a novel quantum federated dataset which can be used for distributed quantum learning, and is the first of its kind in the literature. Finally, the effectiveness of the proposed approach is validated with extensive experiments since we conduct the first extensive experiments that combine Google's TensorFlow Quantum (TFQ) [2] and TensorFlow Federated (TFF) [9]. The results show that the proposed QFL framework achieves comparable, and in some cases superior, performance compared to the centralized, purely QML setup. Our results also show that the model can handle IID and non-IID quantum data. A key finding is that classical FL algorithms can be applied to decentralize the learning in purely quantum QML applications. The pro-

This paper was supported by the U.S. National Science Foundation under Grant CNS-2114267.

posed framework can easily admit to speech, language, and signals processing with quantum computing. For example, the authors in [10] proposed an end-to-end learning quantum framework that includes a quantum tensor network to generate quantum embeddings, followed by a trainable variational quantum circuit. Such a model can be adopted in our proposed QFL framework. However, since we are focusing on purely quantum data, we design our QFL setup based on QCNNs.

Related Works. A handful of prior works [11–13] exist on decentralized QML models, but those works neither address the previously posed challenges nor provide a solution to practically distribute quantum learning in emerging QNs. First, the authors in [12] considered a vertical federated learning approach for decentralized feature extraction in automatic speech recognition tasks. Their model includes a quantum server that uses a QCNN for feature extraction. However, this approach is totally different from our proposed framework as it primarily studies the usage of QCNNs for extracting useful features from classical data, and only assumes a quantum server.

Moreover, a concise vision of a QFL architecture for general optimization in wireless communication networks is discussed in [11]. This prior work considers a radically different scenario than the one treated here. In particular, it considers a wireless network of classical non-quantum mobile users, communicating with access points that run a shallow QNN model for optimizing the wireless network. The access points use FL and share the learning parameters with a server having another deeper QNN model. However, the work in [11] only discusses a conceptual architecture without any implementation, verification, or concrete results. Moreover, it does not consider quantum clients or quantum data in the communication network, and it relies solely on classical data.

Finally, the most relevant prior work is the work in [13] which considered a hybrid quantum-classical ML model trained in a federated setup. In contrast to our QFL framework, the work in [13] adopts a transfer learning approach where a pre-trained CNN model is used to extract features from classical data and compress it into a vector passed to a QNN to make predictions. Although the authors in [13] discuss federating the QML models, their implementation uses classical data and has a very limited contribution to the purely quantum setup since it does not address the biggest challenges such as the lack of a purely quantum federated dataset in the literature and the lack of a holistic implementation. Clearly, there is no prior work that addresses QFL while leveraging purely quantum data learning. This gap in the literature must be extensively addressed as it could provide a breakthrough in the way QNs are looked at.

The rest of this paper is organized as follows. Section 2 describes the problem setup and the QFL model in details. Next, the proposed quantum federated dataset is developed and the proposed process to generate it is presented in Section 3. In Section 4, we conduct the experiments and discuss the key results. Finally, conclusions are drawn in Section 5.

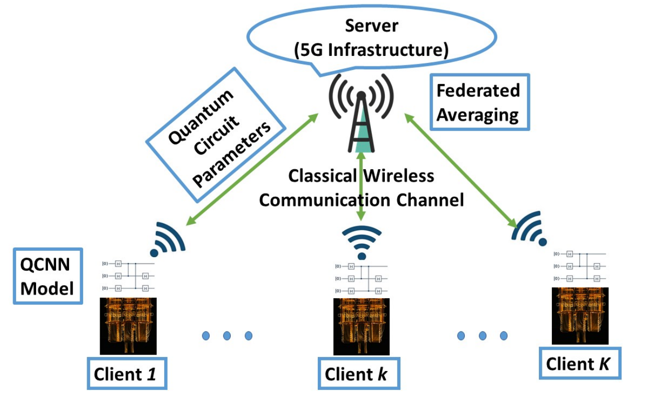

Fig. 1: Proposed general QFL setup.

2. PROBLEM SETUP

We consider a quantum network setup (Figure 1) consisting of a server and a set $\mathcal{K}$ of K quantum computing clients sharing a QCNN model such as the one proposed in [5]. Each client generates its own quantum data locally and trains a local QCNN model to perform binary classification. The generated data consists of excitations for an N-qubits quantum cluster state fed as labeled inputs in pairs $(|\psi_m\rangle, y_m)$: $m = 1, 2, ..., M$, where $|\psi_m\rangle$ represents the m-th sample input quantum state, y_m is a binary label that classifies whether the cluster state is excited or not, and M is the number of data samples.

Analogously to classical convolutional neural networks, the QCNN model includes a sequence of quantum convolution layers followed by quantum pooling layers. After incorporating a sufficient number of layers, a quantum fully connected layer is applied, and predictions are made by performing quantum measurements. More formally, each convolution layer $C \in \mathcal{L}_c$, with $\mathcal{L}_c$ being the set of convolution layers, is a single quasi-local *unitary*. In particular, unitarity is the main constraint that must be imposed on matrices that represent quantum gates [1]. A unitary is called quasi-local if a quasi-local Hamiltonian generated that unitary as explained in [14].

In a quantum pooling layer $P \in \mathcal{L}_p$, with $\mathcal{L}_p$ being the set of pooling layers, the results of measuring some qubits are used to determine unitary rotations applied to close qubits. This results in a reduced system size, and introduces non-linearities. After a sufficient size reduction, a quantum fully connected layer F is applied on the remaining qubits. Measurements on some output qubits will represent the predictions.

At each client, the learnable parameters are the entries of all unitaries, i.e., the quantum circuit parameters, which are classical values. Let $\boldsymbol{\theta}^k = (C, P, F)$, where $k \in \mathcal{K}$, be the vector of all parameters for client k, then, the predicted output value by the QCNN model f to input quantum state $|\psi_m\rangle$ for client k will be $f_{\boldsymbol{\theta}^k}(|\psi_m\rangle)$. Initially, the model parameters are initialized, and, then, updated by minimizing the following mean squared error (MSE) loss function:

$$\arg\min_{\boldsymbol{\theta}^k} \mathcal{J}(\boldsymbol{\theta}^k) := \frac{1}{2M} \sum_{m=1}^{M} (y_m - f_{\boldsymbol{\theta}^k}(|\psi_m\rangle))^2. \quad (1)$$

In order to benefit from each other's experience, all K clients will collaborate in training the same QCNN model. Such a collaboration is fundamental for distributing quantum learning in emerging QNs. To perform this collaboration, the general setup of the QFL framework follows a similar structure to classical FL [15]. The collaborative learning is done by using existing wireless communication infrastructure to send the clients' "classical" model parameters to the server.

Each round h of the QFL training starts with the server sending its current version of the model parameters $\boldsymbol{\theta}_h^s$ to all K clients. Each client $k \in \mathcal{K}$ uses its local quantum data to run an optimization algorithm such as stochastic gradient descent (SGD), in order to update its model parameters. Next, each client k sends its updated model parameters back to the server which aggregates the parameters from all clients. Then, the server applies the popular Federated Averaging[1] FL algorithm [16] to estimate an average update of the model parameters and send the newly updated parameters to all clients according to the following rule:

$$\boldsymbol{\theta}_{\boldsymbol{h+1}}^{\boldsymbol{s}} = \sum_{k \in \mathcal{K}} w_k \cdot \boldsymbol{\theta}_{\boldsymbol{h}}^{\boldsymbol{k}}, \tag{2}$$

where the weighting vector $\boldsymbol{w} = (w_1, w_2, ..., w_k)$ is assigned by the server to weigh different clients in the network. This process is repeated until convergence. Next, we describe the steps to generate the quantum federated dataset necessary to practically implement the QFL framework.

3. QUANTUM FEDERATED DATASET

Given the lack of existing quantum federated dataset in the literature, our proposed QFL framework begins by generating the first quantum federated dataset that can be used for distributed quantum learning. The generated dataset has a hierarchical data format and consists of purely quantum data.

Quantum Data. A variety of purely quantum data exists for different quantum many-body systems [3], and is generated using different quantum devices or complex quantum simulations.[2] For instance, [5] considered symmetry-protected topological phases [18] as input data for classification by a QCNN. We adopt a simpler, yet practical, form of quantum data consisting of quantum clusters inspired from [19].

The proposed quantum dataset consists of excitations of quantum cluster states [20, 21], labeled as either excited or not based on the rotations of the qubits. This type of data is of important use for quantum sensing networks like the ones applied for metrology [22]. In addition, quantum cluster states are particularly important because of their applicability in distributed QNs, and their ability to teleport quantum states between quantum clients communicating through a quantum channel [23]. Thus, the adopted data type perfectly fits our targets and allows for future extensions to QNs incorporating both classical and quantum clients.

Generating Single Client Data. We used TFQ and Google's framework for quantum circuit programming: Cirq [24] for generating the quantum data of each client. We begin by generating a rectangular grid of 1×8 qubits using Cirq, such a size is reasonable for QML simulations and is sufficient for validating the proposed QFL framework. Then, we create cluster states as TFQ circuits consisting of the Hadamard and Controlled-Z quantum gates [1], and apply the circuit on the generated qubits. In order to define the excitations of cluster states, we observe that most quantum gates operating on a single qubit can be described as rotations around an axis in the Bloch sphere [25, 26]. Thus, they are usually referred to by their axis of rotation [24]. As proposed by [19], we consider excitations to be represented by rotation gates around the x-axis of the Bloch sphere (R_x quantum gates [1]). An excitation of the cluster state is declared if a large enough rotation is achieved and the state is labeled with 1. If the rotation is not sufficiently large, then the state is declared as unexcited, and is labeled with a 0.

Generating Federated Dataset. The input of the previously described quantum data is represented by quantum circuits. In order to be able to store the data, the quantum circuit is transformed into a tensor that is represented by strings. In fact, these strings represent an encoding of the serialized binary data of the quantum circuits with TensorFlow data type "lS5000". We particularly generate a hierarchical data format version 5 (HDF5) federated dataset file which includes examples of 30 clients (Different numbers are considered in Section 4). Each client has its own quantum dataset consisting of an input vector with a single feature with 160 serialized binary sample, and a labels' vector of binary integers. In addition, we generate a quantum federated dataset consisting of non-IID clients' datasets (explained in Section 4). The generated dataset can be found using the following link[3]. Next, we describe the conducted experiments and discuss the obtained results.

4. EXPERIMENTS AND RESULTS

Our experiments constitute the first implementation that combines the usage of Google's TFQ and TFF frameworks, which are used to implement our proposed QFL framework. In particular, we build upon the implementations in [19] and [27], and we run our simulations on Google's Cloud Platform known as "Google Colaboratory [28]" using CPU computing nodes. We consider QCNNs with 8 qubits (this is a typical value for quantum sensing networks). Moreover, the width of the QCNN is not considered as an optimization parameter since it solely relies on the number of qubits in the system. In our implementation, we find that having three pairs of quantum convolution-pooling layers, with 64 learnable parameters, is the most suitable QCNN architecture that fits our setup.

Compared to classical FL scenarios, it is natural to assume that the number of quantum computing clients in a QN will be in the range of tens of clients. For ease of simulations, we assume that all quantum clients join the training, and we reserve the data of a small portion of clients for testing. We adopt the

[1] Other advanced FL algorithms can be accommodated in our framework.

[2] Note that classical data can be encoded into quantum states and fed to quantum computers as quantum data as discussed in [17]

[3] https://github.com/MahdiChehimi/Quantum-Federated-Learning-with-Quantum-Data.git

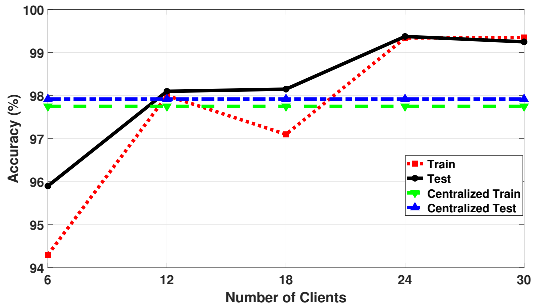

Fig. 2: Evaluation of QFL accuracy vs number of clients.

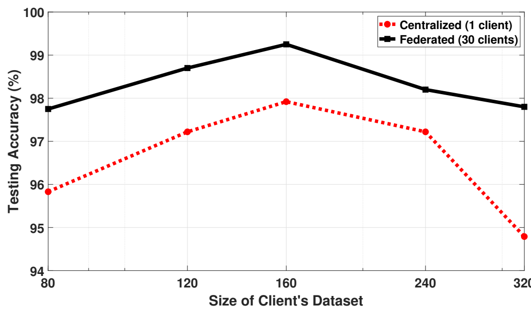

Fig. 3: Size of client's dataset vs testing accuracy.

binary accuracy metric with a threshold of 0.5 as a validation metric. This is a typical choice for binary classification. Since TFF is currently only available in simulation environments, then data is assumed to be available locally to clients.

Impact of Number of Clients. We begin our experiments by analyzing the impact of the number of available clients in the QFL network. In Figure 2, we compare the achieved testing accuracy when the quantum federated dataset has 1 (centralized), 6 (4 for training, 2 for testing), 12 (9 for training, 3 for testing), 18 (14 for training, 4 for testing), 24 (19 for training, 5 for testing), and 30 (25 for training, 5 for testing) clients while fixing the number of data samples for each client to 160 samples. We observe that, in general, as the number of participating clients in the QFL setup increases, a higher testing accuracy is achieved without overfitting the training data. The reason why the case of 6 clients achieves a smaller accuracy is because the number of clients in a federated setup must be large enough in order to achieve efficient learning.

Impact of the Size of Clients' Data Samples. In Figure 3, we consider a QFL network of 30 active clients, and we compare the achieved testing accuracy with the centralized case (single client) while varying the size of the individual client's datasets. Since the adopted QCNN is shallow with a small number of trainable parameters, we observe that increasing the size of the dataset does not necessarily guarantee an improvement in the performance. In fact, as long as each client has enough data, increasing the size of the dataset may slightly increase or decrease the achieved testing accuracy. Moreover, we observe that, in this setup, the federated framework achieves a superior performance compared to the centralized case. This is because each client in the federated setup benefits from the data of the other clients joining the learning process.

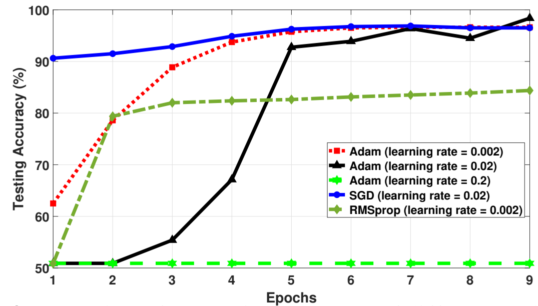

Fig. 4: Evolution of the testing accuracy of different optimizers over the training epochs.

Table 1: Performance comparison using IID and non-IID data.

Dataset	Testing Accuracy	Testing MSE
IID Data	99.25	5.66
Non-IID Data	98.625	6.57

Impact of Optimizers and Learning Rates. In Figure 4, we compare the performance of the QFL framework with different optimizers (Adam, SGD, and RMSprop) and learning rates. We observe that the RMSprop optimizer with a learning rate of 0.002 is slow compared to the other optimizers and converges at a smaller testing accuracy. For the SGD optimizer with a learning rate of 0.02, we observe that it is the only one that achieves a high accuracy from the first epoch, and it converges to a value around 96.5%. However, the Adam optimizer with a learning rate of 0.02 converges to a higher testing accuracy after few training epochs. Finally, we observe that a learning rate of 0.2 for Adam optimizer is very large that it cannot learn, while a learning rate of 0.002 results in a smoother curve at the expense of a smaller accuracy.

Impact of Non-IID Data. When generating the quantum cluster states from the input qubits, the rotation values fed to the R_x gate were drawn from a uniform distribution between $-\pi$ and π. In order to vary the underlying distribution of the clients' data, we consider generating the data for half of the clients using a truncated normal distribution, so that we generate non-IID quantum data. Then, the performance of the QFL framework on the IID and non-IID federated datasets is compared for a network of 30 clients (25 for training, 5 for testing) with 160 data samples each. In Table 1, we show the testing accuracy and MSE loss for both datasets and observe that the proposed QFL framework achieves a similar performance on both IID and non-IID quantum federated datasets.

5. CONCLUSION

In this paper, we have proposed a novel framework, QFL, for distributing quantum learning over quantum data in emerging QNs, by leveraging classical communication networks. To implement this framework, we have generated the first quantum federated dataset in the literature and performed a unique implementation that combines TFQ and TFF. We have conducted extensive experiments whose results validated the effective behavior of the proposed QFL framework using the federated averaging algorithm. Future work includes incorporating quantum algorithms in the federated averaging process.

6. REFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[2] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters *et al.*, "Tensorflow quantum: A software framework for quantum machine learning," *arXiv preprint arXiv:2003.02989*, 2020.

[3] H. Tasaki, *Physics and mathematics of quantum many-body systems*. Springer Nature, 2020.

[4] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, p. 043001, 2019.

[5] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.

[6] J. Bausch, "Recurrent quantum neural networks," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[8] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, "On the tradeoff between energy, precision, and accuracy in federated quantized neural networks," *arXiv preprint arXiv:2111.07911*, 2021.

[9] A. Ingerman and K. Ostrowski, "Introducing tensorflow federated," 2019. [Online]. Available: https://medium.com/tensorflow/introducing-tensorflow-federated-a4147aa20041

[10] J. Qi, C.-H. H. Yang, and P.-Y. Chen, "Qtn-vqc: An end-to-end learning framework for quantum neural networks," *arXiv preprint arXiv:2110.03861*, 2021.

[11] B. Narottama and S. Y. Shin, "Quantum federated learning for wireless communications," *Journal of the Korean Institute of Communication Sciences*, 2020.

[12] C.-H. H. Yang, J. Qi, S. Y.-C. Chen, P.-Y. Chen, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6523–6527.

[13] S. Y.-C. Chen and S. Yoo, "Federated quantum machine learning," *Entropy*, vol. 23, no. 4, p. 460, 2021.

[14] M. Mariën, K. M. Audenaert, K. Van Acoleyen, and F. Verstraete, "Entanglement rates and the stability of the area law for the entanglement entropy," *Communications in Mathematical Physics*, vol. 346, no. 1, pp. 35–73, 2016.

[15] F. Yu, A. S. Rawat, A. Menon, and S. Kumar, "Federated learning with only positive labels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 946–10 956.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[17] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Physical Review A*, vol. 103, no. 3, p. 032430, 2021.

[18] N. Schuch, D. Pérez-García, and I. Cirac, "Classifying quantum phases using matrix product states and projected entangled pair states," *Physical review b*, vol. 84, no. 16, p. 165139, 2011.

[19] "Quantum convolutional neural network: Tensorflow quantum," 2020. [Online]. Available: https://www.tensorflow.org/quantum/tutorials/qcnn

[20] M. A. Nielsen, "Cluster-state quantum computation," *Reports on Mathematical Physics*, vol. 57, no. 1, pp. 147–161, 2006.

[21] H. J. Briegel, "Cluster states," in *Compendium of Quantum Physics*. Springer, 2009, pp. 96–105.

[22] N. Friis, D. Orsucci, M. Skotiniotis, P. Sekatski, V. Dunjko, H. J. Briegel, and W. Dür, "Flexible resources for quantum metrology," *New Journal of Physics*, vol. 19, no. 6, p. 063044, 2017.

[23] I. B. Djordjevic, "On global quantum communication networking," *Entropy*, vol. 22, no. 8, p. 831, 2020.

[24] Qantum AI team and collaborators, "Cirq," Oct 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4062499

[25] T. Lu, X. Miao, and H. Metcalf, "Bloch theorem on the bloch sphere," *Physical Review A*, vol. 71, no. 6, p. 061405, 2005.

[26] I. Glendinning, "The bloch sphere," in *QIA Meeting. Vienna*, 2005.

[27] "Federated learning workshop using tensorflow federated," July 2020. [Online]. Available: https://colab.research.google.com/drive/1kCSSFUCU_rxW7MElwZXENe50_VTnvGH1?usp=sharing

[28] E. Bisong, *Google Colaboratory*. Berkeley, CA: Apress, 2019, pp. 59–64.