

# Federated Quantum-Train with Batched Parameter Generation

Chen-Yu Liu <sup>†||</sup>, Samuel Yen-Chi Chen <sup>†\*\*</sup>

<sup>†</sup>Graduate Institute of Applied Physics, National Taiwan University, Taipei, Taiwan

<sup>‡</sup>Wells Fargo, New York, NY, USA

Email: <sup>||</sup> d10245003@g.ntu.edu.tw, <sup>\*\*</sup> yen-chi.chen@wellsfargo.com

**Abstract**—In this work, we introduce the Federated Quantum-Train (QT) framework, which integrates the QT model into federated learning to leverage quantum computing for distributed learning systems. Quantum client nodes employ Quantum Neural Networks (QNNs) and a mapping model to generate local target model parameters, which are updated and aggregated at a central node. Testing with a VGG-like convolutional neural network on the CIFAR-10 dataset, our approach significantly reduces qubit usage from 19 to as low as 8 qubits while reducing generalization error. The QT method mitigates overfitting observed in classical models, aligning training and testing accuracy and improving performance in highly compressed models. Notably, the Federated QT framework does not require a quantum computer during inference, enhancing practicality given current quantum hardware limitations. This work highlights the potential of integrating quantum techniques into federated learning, paving the way for advancements in quantum machine learning and distributed learning systems.

**Index Terms**—Quantum Machine Learning, Federated Learning, Quantum-Train

## I. INTRODUCTION

Quantum computing (QC) promises potential computational advantages for certain tasks over classical computers, particularly in areas like machine learning (ML) and combinatorial optimization problems [1]–[8]. Meanwhile, the advances in classical ML and artificial intelligence (AI) have demonstrated amazing capabilities in various tasks [9]–[12]. With the progress in quantum hardware, it is natural to consider the combination of these two fascinating technologies. While existing quantum computing devices still suffer from noises and imperfections, a hybrid quantum-classical computing paradigm [13] which divides computational tasks among quantum and classical computing resources according to their properties to leverage the best part from the both world is proposed. Variational quantum algorithms (VQAs) [13] are the fundamental algorithms framework under this hybrid paradigm. Leading quantum machine learning (QML) methods largely rely on these variational algorithms. Variational quantum circuits (VQCs) are the building blocks of existing QML models [14]. It has been shown theoretically that VQC can outperform classical models when certain conditions

The views expressed in this article are those of the authors and do not represent the views of Wells Fargo. This article is for informational purposes only. Nothing contained in this article should be construed as investment advice. Wells Fargo makes no express or implied warranties and expressly disclaims all legal, tax, and accounting implications related to this article.

are met [1], [15], [16]. VQC-based QML models have been shown to be successful in various ML tasks ranging from classification [14], [17]–[22], time-series modeling [23], [24], audio and language processing [25]–[27], quantum algorithm reconstruction [28], [29], and reinforcement learning [30]–[34].

The great success of modern AI/ML techniques not only depend on good model architecture design but also on the volume of high-quality data and QML is no exception. The requirements of data also raise the privacy concerns in the QML research and application. Among various methods to mitigate the privacy concerns, *federated learning* (FL) is a method in which various participating parties share the locally trained models but not the actual training data to avoid data leakage.

Several FL methods have been proposed in the realm of QML to enhance the privacy-preserving features [35]–[40].

While effective, these quantum FL (QFL) methods require the trained models to be used on quantum devices in the inference stage. It posts certain challenges at the moment as there are limited real quantum resources available and it is unclear whether the proposed methods are realistic in the real-world scenarios.

In this paper, we propose a Quantum-Train (QT)-based [19], [20], [24] QFL method in which the quantum neural networks (QNN) are trained to generate the well-performing classical neural network weights in the federated setting. Once the training is finished, the QNN is not used during the inference phase. Our main contributions are:

- *Addressing data encoding issue in QFL*: The QT approach integrated with FL simplifies data handling by using classical inputs and outputs, avoiding the complexities and potential information loss of encoding large datasets into quantum states. This method retains quantum computational advantages without the scaling difficulties of quantum data encoding.
- *Reduction of qubit count in QT*: Utilizing the batched parameter generation approach, we reduce the qubit usage from  $\lceil \log_2 m \rceil$  to  $\lceil \log_2 \lceil \frac{m}{n_{mlp}} \rceil \rceil$ , compared to the original QT proposal. Here,  $m$  is the number of parameters of the target classical model and  $n_{mlp}$  is the batch size in the parameter generation approach. In the example examined in this study, qubit usage is reduced from 19 to as low as 8 qubits.

- *Inference without quantum hardware*: The training results of QT are designed to operate seamlessly on classical hardware, eliminating the need for quantum computing resources, unlike conventional QML and QFL. This feature enhances its applicability, especially given the current limited access to quantum computers compared to classical counterparts.

## II. FEDERATED LEARNING

Federated Learning (FL) [41] has emerged in response to growing privacy concerns associated with large-scale datasets and cloud-based deep learning [42]. In the FL framework, the primary components are a *central node* and multiple *client nodes*. The central node maintains the *global model* and collects trained parameters from the client nodes. It then performs an *aggregation* process to update the global model, which is subsequently shared with all client nodes. The client nodes locally train the received model using their own data, which typically constitutes a small subset of the overall dataset. The concept of FL has been explored in the field of QML since the publications [35], [36]. In [35], the authors examined the simplest form of QFL utilizing hybrid quantum-classical models. In this approach, a pre-trained CNN compresses input images into a dimension manageable by a VQC. The locally trained hybrid model parameters are then uploaded to a central server, which aggregates these parameters and distributes the updated model to all participants. This framework has been further enhanced to process sequential data using a federated quantum LSTM network [43]. The study [36] delves into a more advanced scenario where QFL processes quantum states instead of classical images. While QFL can mitigate the risk of direct leakage of training datasets, it remains vulnerable to attacks that can extract training data entries from the trained models themselves. Such attacks pose a significant threat to data privacy. To address this issue, [40] explores the integration of differentially-private gradient optimizers with QFL, aiming to enhance the privacy of QML models. QFL can be further extended to scenarios where training is conducted on encrypted data, as demonstrated in [44]. QFL can be applied in diverse scenarios, including autonomous vehicles [39] and quantum fuzzy learning [45].

## III. VARIATIONAL QUANTUM CIRCUITS AND QUANTUM-TRAIN

At the core of the QML scheme, VQCs play a pivotal role by providing the parameterized ansatz that forms the function approximator for learning tasks. A typical VQC used as a QNN is depicted on the left side of Fig. 1. The process begins with the initial state  $|0\rangle^{\otimes N}$ , where  $N$  is the number of qubits. This is followed by parameterized single-qubit and two-qubit unitary operations  $U_3$  and controlled- $U_3$  ( $CU_3$ ) gates, characterized by their matrix representations:

$$U_3(\mu, \varphi, \lambda) = \begin{bmatrix} \cos(\mu/2) & -e^{i\lambda} \sin(\mu/2) \\ e^{i\varphi} \sin(\mu/2) & e^{i(\varphi+\lambda)} \cos(\mu/2) \end{bmatrix} \quad (1)$$

$$CU_3 = I \otimes |0\rangle\langle 0| + U_3(\mu, \varphi, \lambda) \otimes |1\rangle\langle 1|, \quad (2)$$

The parameterized quantum state (QNN) can then be described as:

$$|\psi(\theta)\rangle = \left( \prod_i CU_3^{i,i+1} \prod_j U_3^j \right)^L |0\rangle^{\otimes N}, \quad (3)$$

where  $i$  and  $j$  are qubit indices, and  $L$  is the number of repetitions. The proposed vanilla QT [19] is as follows: consider a target neural network model with parameters  $\omega$ , where  $\omega = (\omega_1, \omega_2, \dots, \omega_m)$  and  $m$  is the total number of parameters. Instead of updating all  $m$  parameters as in conventional ML, QT utilizes  $|\psi(\theta)\rangle$ , a QNN with  $N = \lceil \log_2 m \rceil$  qubits, to generate  $2^N$  distinct measurement probabilities  $|\langle \phi_i | \psi(\theta) \rangle|^2$  for  $i \in \{1, 2, \dots, 2^N\}$ , where  $|\phi_i\rangle$  is the  $i$ -th basis state. These probabilities are then input into a mapping model  $G_\beta$ , a multi-layer perceptron (MLP) type classical neural network with parameters  $\beta$ . The first  $m$  basis measurement result probabilities, along with the vector representations of the corresponding basis states  $|\phi_i\rangle$ , are mapped from values bounded between 0 and 1 to  $-\infty$  and  $\infty$  using the following equation:

$$G_\beta(|\phi_i\rangle, |\langle \phi_i | \psi(\theta) \rangle|^2) = \omega_i, \quad i = 1, 2, \dots, m. \quad (4)$$

Here, it can be observed that the parameter  $\omega$  of the target model is generated from the QNN  $|\psi(\theta)\rangle$  and the mapping model  $G_\beta$ . Notably, the required number of parameters for both  $\theta$  and  $\beta$  scales as  $O(\text{polylog}(m))$  [19], allowing for the effective training of the target model with  $m$  parameters by only tuning  $O(\text{polylog}(m))$  parameters of  $\theta$  and  $\beta$ . Unlike conventional QML approaches, which require the QNN during the inference stage, the QT approach decouples the quantum computing resource after training. Since the QNN is used solely for generating the parameters of the target model, the resulting trained model is a classical neural network. This classical model can then be executed entirely on classical computing hardware. This characteristic is particularly practical given that quantum computing hardware is currently a relatively rare and expensive resource.

## IV. QUANTUM-TRAIN WITH BATCHED PARAMETER GENERATION

Building upon the previously proposed QT method, which generates a single parameter of the target network model from a single basis measurement probability, this study introduces a batch parameter generation approach. This method generates a batch of parameters from a single basis measurement probability, as illustrated in Fig. 1. In this approach, the  $m$  parameters of the target model are divided into  $n_{ch}$  chunks, each containing  $n_{mlp}$  parameters, such that  $n_{ch} = \lceil m/n_{mlp} \rceil$ . The mapping model, now denoted as  $\tilde{G}_\beta$ , takes as input  $|\phi_i\rangle$  and  $|\langle \phi_i | \psi(\theta) \rangle|^2$  and generates a batch of parameters in  $\omega$  of size  $n_{mlp}$ :

$$\tilde{G}_\beta(|\phi_i\rangle, |\langle \phi_i | \psi(\theta) \rangle|^2) = \vec{\omega}_i, \quad i = 1, 2, \dots, n_{ch}, \quad (5)$$

$$\vec{\omega}_i = (\omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,j}), \quad j = 1, 2, \dots, n_{mlp}. \quad (6)$$

This setup is realized through a decoder-like architecture of the MLP in the mapping model  $\tilde{G}_\beta$ , where the output size

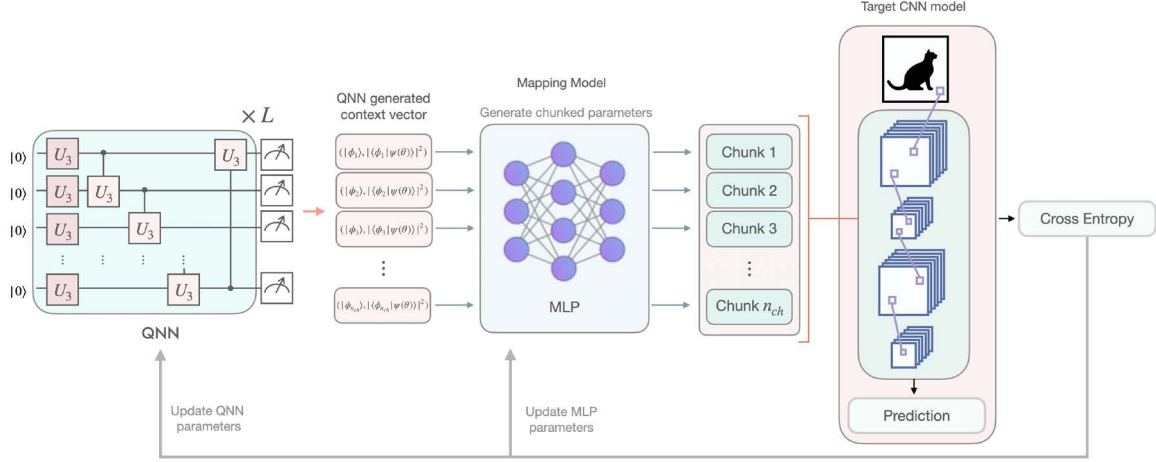


Fig. 1: Illustration of the QT method with batched parameter generation. The target model parameters  $\omega$  are divided into chunks, each containing  $n_{mlp}$  parameters. The QNN generates measurement probabilities which are mapped by  $\tilde{G}_\beta$  to produce a batch of parameters  $\tilde{\omega}_i$ . This approach reduces qubit usage and maintains the benefits of the QT method.

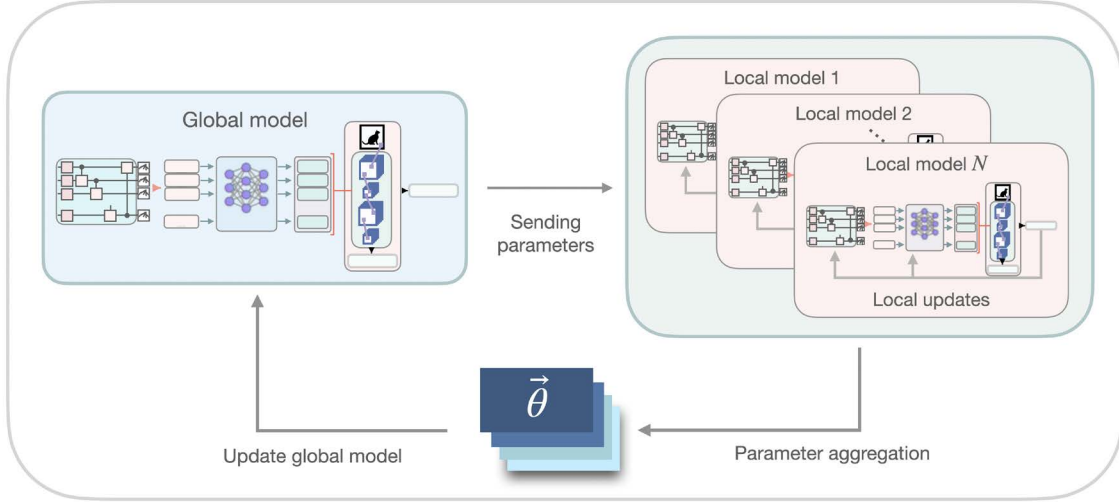


Fig. 2: Diagram of the Federated QT framework. Each quantum client node employs a QNN and a mapping model  $\tilde{G}_\beta$  to generate local target model parameters. These parameters are updated based on local datasets and aggregated at a central node to update the global model. The framework leverages quantum computing for training while only requiring classical computation during inference.

is expanded from 1 to  $n_{mlp}$ , or `mlp_out` in the following syntax. Consequently, the qubit usage  $N$  is reduced from  $N = \lceil \log_2 m \rceil$  to

$$N = \lceil \log_2 n_{ch} \rceil = \lceil \log_2 \lceil \frac{m}{n_{mlp}} \rceil \rceil, \quad (7)$$

effectively saving approximately  $\lceil \log_2 n_{mlp} \rceil$  qubits from the original QT proposal, the original method can be considered as a special case with  $n_{mlp} = 1$ . Reducing qubit usage also mitigates the issue of the exponential requirement of measurement shots, as mentioned in the original QT study.

The remaining training process is similar to the vanilla QT method, as depicted in Fig. 1.

## V. FEDERATED QUANTUM-TRAIN

Following the original idea of FL and QFL, we introduce the concept of the QT model within the federated framework. In this approach, each quantum client node employs QNN  $|\psi(\theta)\rangle$  and mapping model  $\tilde{G}_\beta$  to generate the local target model parameters. During each training round, every quantum client nodes update their QNN parameters and the associated mapping model parameters based on their local datasets. These updated parameters are sent to the central node, where they

are aggregated to update the global model, as depicted in Fig. 2. This process ensures that the global model benefits from the QT performed at each client node, leading to improved performance and efficiency. By integrating the QT model into FL, we leverage the advantages of quantum computing to reduce the number of training parameters and enhance the scalability of distributed learning systems. Notably, compared to traditional QFL, federated QT does not require a quantum computer during the inference stage.

## VI. RESULT AND DISCUSSION

To examine the applicability of the proposed federated QT framework, we tested it using a VGG-like convolutional neural network (CNN) structure with the CIFAR-10 dataset. The target CNN model has 285226 parameters. We tested three different QT setups:  $n_{mlp} \in \{2000, 1000, 500\}$ , while fixing the repetition  $L = 5$ . The required qubit usage, derived from Eq. 7, is 8, 9, and 10 qubits, respectively. Compared to the original QT with the same CNN model [19], which required 19 qubits, our new batched parameter generation significantly reduces the qubit usage. Fig. 3 illustrates the number of model parameters for the models investigated in this study, with  $n_{mlp}$  denoted as `mlp_out`.

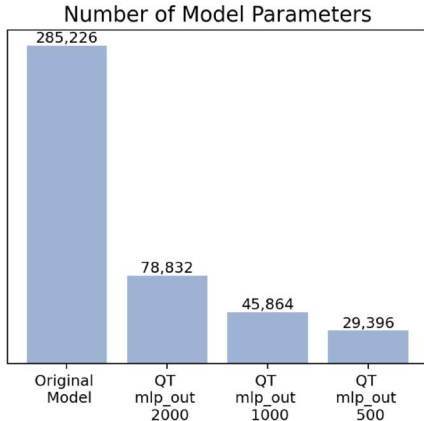


Fig. 3: Comparison of the number of training parameters for different models used in this study.

In the upper row of Fig. 4, the Cross-Entropy loss for the CIFAR-10 image classification task, involving 10 classes over multiple rounds, is presented with different setups of local epochs and `mlp_out`, while fixing the number of clients at 4. It can be observed that a larger number of local epochs leads to lower loss values. This outcome is expected, as the model undergoes more frequent updates, providing more opportunities to correct incorrect predictions.

In the lower row of Fig. 4, the local epoch is fixed to 1, and the effect of varying the number of clients is investigated. In this investigation, the dataset is divided into as many pieces as the number of clients. Interestingly, increasing the number of clients results in better performance in terms of training loss. This improvement can be attributed to the flexibility provided by different local models, which adjust

distinct sets of parameters corresponding to different parts of the dataset. The parameter aggregation step then incorporates these updates from diverse perspectives, enhancing the overall model performance.

A noticeable trend in the figures is that models with more parameters tend to perform better in terms of training loss. This can be attributed to the increased expressiveness of the corresponding models, which enhances their ability to fit the training data. However, this observation only indicates the model’s effectiveness on the training dataset and does not necessarily reflect its general performance on unseen data.

As illustrated in Fig. 5, we present the testing and training accuracy of the models investigated in Fig. 4. A notable observation is the significant overfitting in the purely classical case. While the training accuracy of the classical model is extremely high, its testing accuracy is slightly lower than that of the  $n_{mlp} = 2000$  case. This behavior underscores an advantage of the QT method, as highlighted in previous studies [19]: the QT method can reduce the deviation between training and testing accuracy, which is proportional to the generalization error. Moreover, our batched parameter generation approach not only significantly reduces qubit usage but also preserves the advantage of generalization error reduction inherent in the vanilla QT method.

While there is no clear trend for different local epochs and the number of clients in the classical,  $n_{mlp} = 2000$ , and  $n_{mlp} = 1000$  cases, the  $n_{mlp} = 500$  case shows an increase in both training and testing accuracy with an increase in local epochs and the number of clients. This behavior demonstrates that the federated framework combined with QT can improve models with extreme compression, such as the  $n_{mlp} = 500$  case, which uses only about 10% of the original CNN model’s parameters.

## VII. CONCLUSION

In this work, we introduced the Federated QT framework, integrating the QT model into federated learning to leverage quantum computing for distributed learning systems. Each quantum client node employs QNNs and a mapping model  $\tilde{G}_\beta$  to generate local target model parameters. These parameters are updated based on local datasets and aggregated at a central node, enhancing the global model through quantum-enhanced training.

Our experiments, using a VGG-like CNN on the CIFAR-10 dataset, demonstrate the efficacy of the Federated QT framework. We tested three different QT setups with varying  $n_{mlp}$  values, significantly reducing the required qubit usage compared to the original QT method. Specifically, our batched parameter generation approach reduced qubit usage from 19 to as low as 8, while maintaining the benefits of generalization error reduction.

Results indicate that models with more parameters perform better in training loss due to increased expressiveness, but overfitting was observed in purely classical models. The QT method mitigated this issue, resulting in a closer alignment

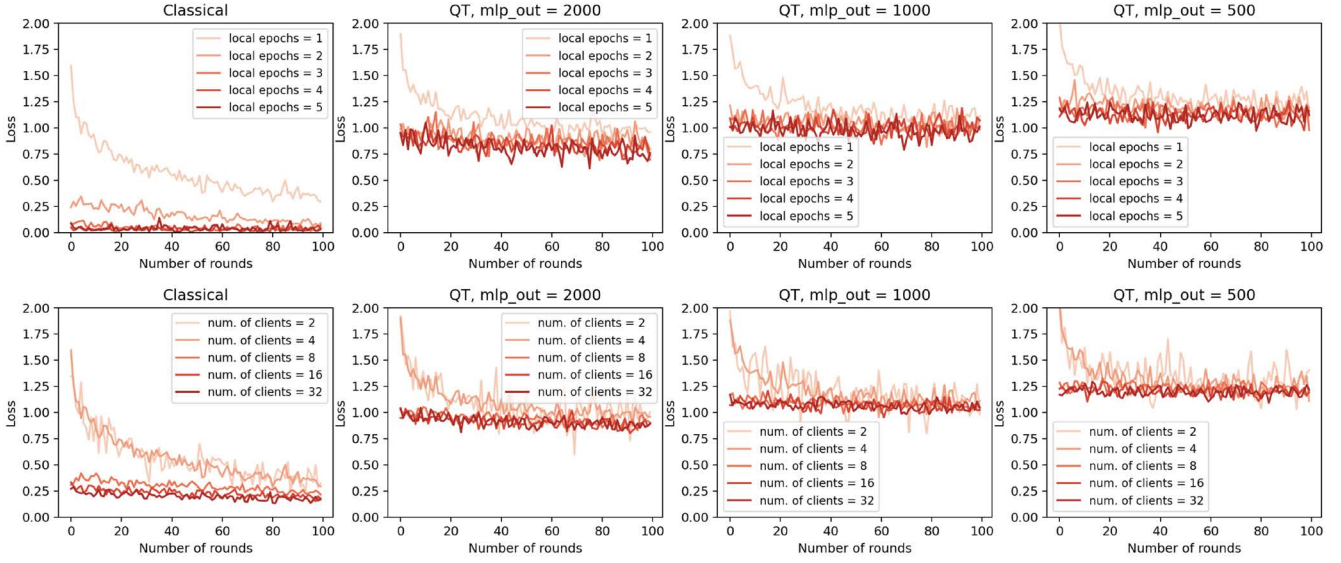


Fig. 4: The Cross-Entropy loss for the CIFAR-10 image classification task across multiple training rounds. The upper row illustrates the effect of varying the number of local epochs and  $mlp\_out$ , with the number of clients fixed at 4. The lower row demonstrates the effect of varying the number of clients while keeping the local epoch fixed at 1. The results indicate that a larger number of local epochs and clients lead to lower loss values, suggesting improved model performance.

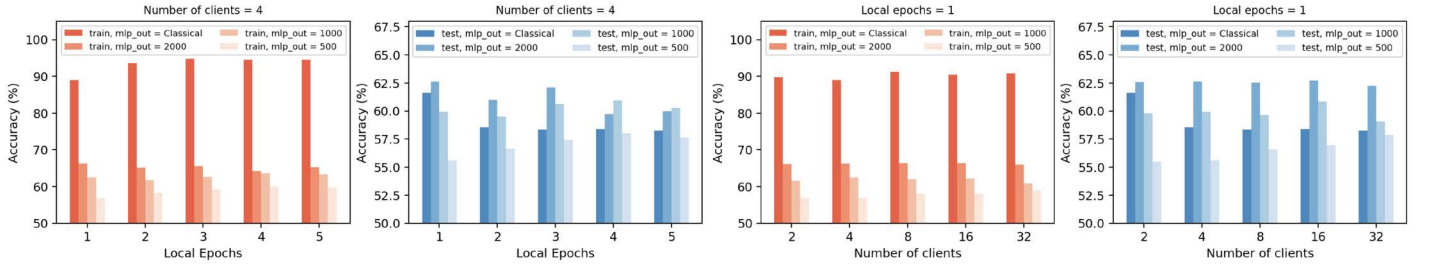


Fig. 5: The training and testing accuracy for the models investigated in Fig. 4. The results reveal significant overfitting in the purely classical model, with high training accuracy but lower testing accuracy compared to the  $n_{mlp} = 2000$  case. The QT method reduces the deviation between training and testing accuracy, highlighting its advantage in minimizing generalization error. Additionally, the federated QT framework demonstrates improved performance in highly compressed models, such as the  $n_{mlp} = 500$  case, which uses only about 10% of the original CNN model’s parameters.

between training and testing accuracy. The federated framework combined with QT also showed improved performance in highly compressed models, such as the  $n_{mlp} = 500$  case, which uses only about 10% of the original CNN model’s parameters.

The Federated QT framework provides a scalable and efficient approach to distributed learning, utilizing quantum computing to reduce training parameters and enhance model performance. Notably, QT does not require a quantum computer during the inference stage, making it highly practical given the current limitations of quantum hardware. Our findings highlight the practical benefits of integrating quantum techniques into federated learning, paving the way for future advancements in QML and distributed learning systems.

## REFERENCES

- [1] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, “The power of quantum neural networks,” *Nature Computational Science*, vol. 1, no. 6, pp. 403–409, 2021.
- [2] C.-Y. Liu, H. Matsuyama, W.-h. Huang, and Y. Yamashiro, “Quantum local search for traveling salesman problem with path-slicing strategy,” *arXiv preprint arXiv:2407.13616*, 2024.
- [3] J. Raymond, R. Stevanovic, W. Bernoudy, K. Boothby, C. C. McGeoch, A. J. Berkley, P. Farré, J. Pasvolsky, and A. D. King, “Hybrid quantum annealing for larger-than-qpq lattice-structured problems,” *ACM Transactions on Quantum Computing*, vol. 4, no. 3, pp. 1–30, 2023.
- [4] C.-Y. Liu and H.-S. Goan, “Hybrid gate-based and annealing quantum computing for large-size ising problems,” *Bulletin of the American Physical Society*, vol. 68, 2023.
- [5] M. Booth, S. P. Reinhardt, and A. Roy, “Partitioning optimization problems for hybrid classical,” *quantum execution. Technical Report*, pp. 01–09, 2017.
- [6] C.-Y. Liu and K.-C. Chen, “Parallel quantum local search via evolutionary mechanism,” *arXiv preprint arXiv:2406.06445*, 2024.

- [7] F. Phillipson and H. S. Bhatia, "Portfolio optimisation using the d-wave quantum annealer," in *International Conference on Computational Science*. Springer, 2021, pp. 45–59.
- [8] C.-Y. Liu, H.-Y. Wang, P.-Y. Liao, C.-J. Lai, and M.-H. Hsieh, "Implementation of trained factorization machine recommendation system on quantum annealer," *arXiv preprint arXiv:2210.12953*, 2022.
- [9] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, no. 5, pp. 431–434, 2017.
- [10] A. Seif, M. Hafezi, and C. Jarzynski, "Machine learning the thermodynamic arrow of time," *Nature Physics*, vol. 17, no. 1, pp. 105–113, 2021.
- [11] C.-Y. Liu and H.-S. Goan, "Reinforcement learning quantum local search," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 2. IEEE, 2023, pp. 246–247.
- [12] C.-Y. Liu and D.-W. Wang, "Random sampling neural network for quantum many-body problems," *Physical Review B*, vol. 103, no. 20, p. 205107, 2021.
- [13] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, "Noisy intermediate-scale quantum algorithms," *Reviews of Modern Physics*, vol. 94, no. 1, p. 015004, 2022.
- [14] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, no. 3, p. 032309, 2018.
- [15] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, "Expressive power of parametrized quantum circuits," *Physical Review Research*, vol. 2, no. 3, p. 033125, 2020.
- [16] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, "Generalization in quantum machine learning from few training data," *Nature communications*, vol. 13, no. 1, pp. 1–11, 2022.
- [17] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, "Quantum convolutional neural networks for high energy physics data analysis," *Physical Review Research*, vol. 4, no. 1, p. 013231, 2022.
- [18] S. Y.-C. Chen, C.-M. Huang, C.-W. Hsing, and Y.-J. Kao, "An end-to-end trainable hybrid classical-quantum classifier," *Machine Learning: Science and Technology*, vol. 2, no. 4, p. 045021, 2021.
- [19] C.-Y. Liu, E.-J. Kuo, C.-H. A. Lin, J. G. Young, Y.-J. Chang, M.-H. Hsieh, and H.-S. Goan, "Quantum-train: Rethinking hybrid quantum-classical machine learning in the model compression perspective," *arXiv preprint arXiv:2405.11304*, 2024.
- [20] C.-Y. Liu, E.-J. Kuo, C.-H. A. Lin, S. Chen, J. G. Young, Y.-J. Chang, and M.-H. Hsieh, "Training classical neural networks by quantum machine learning," *arXiv preprint arXiv:2402.16465*, 2024.
- [21] K.-C. Chen, X. Li, X. Xu, Y.-Y. Wang, and C.-Y. Liu, "Quantum-hpc framework with multi-gpu-enabled hybrid quantum-classical workflow: Applications in quantum simulations," *arXiv preprint arXiv:2403.05828*, 2024.
- [22] K.-C. Chen, X. Xu, H. Makhnov, H.-H. Chung, and C.-Y. Liu, "Quantum-enhanced support vector machine for large-scale stellar classification with gpu acceleration," *arXiv preprint arXiv:2311.12328*, 2023.
- [23] S. Y.-C. Chen, S. Yoo, and Y.-L. L. Fang, "Quantum long short-term memory," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8622–8626.
- [24] C.-H. A. Lin, C.-Y. Liu, and K.-C. Chen, "Quantum-train long short-term memory: Application on flood prediction problem," *arXiv preprint arXiv:2407.08617*, 2024.
- [25] R. Di Sipio, J.-H. Huang, S. Y.-C. Chen, S. Mangini, and M. Worring, "The dawn of quantum natural language processing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8612–8616.
- [26] J. Stein, I. Christ, N. Kraus, M. B. Mansky, R. Müller, and C. Linnhoff-Popien, "Applying qnlp to sentiment analysis in finance," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 2. IEEE, 2023, pp. 20–25.
- [27] S. S. Li, X. Zhang, S. Zhou, H. Shu, R. Liang, H. Liu, and L. P. Garcia, "PqIm-multilingual decentralized portable quantum language model," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [28] C.-Y. Liu, "Practical quantum search by variational quantum eigensolver on noisy intermediate-scale quantum hardware," in *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2023, pp. 397–403.
- [29] C.-Y. Liu, C.-H. A. Lin, and K.-C. Chen, "Learning quantum phase estimation by variational quantum circuits," *arXiv preprint arXiv:2311.04690*, 2023.
- [30] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, "Variational quantum circuits for deep reinforcement learning," *IEEE access*, vol. 8, pp. 141 007–141 024, 2020.
- [31] O. Lockwood and M. Si, "Reinforcement learning with quantum variational circuit," in *Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment*, vol. 16, no. 1, 2020, pp. 245–251.
- [32] S. Y.-C. Chen, C.-M. Huang, C.-W. Hsing, H.-S. Goan, and Y.-J. Kao, "Variational quantum reinforcement learning via evolutionary optimization," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015025, 2022.
- [33] W. J. Yun, J. Park, and J. Kim, "Quantum multi-agent meta reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11 087–11 095.
- [34] C.-Y. Liu, C.-H. A. Lin, C.-H. H. Yang, K.-C. Chen, and M.-H. Hsieh, "Qrl: Toward practical quantum reinforcement learning via quantum-train," *arXiv preprint arXiv:2407.06103*, 2024.
- [35] S. Y.-C. Chen and S. Yoo, "Federated quantum machine learning," *Entropy*, vol. 23, no. 4, p. 460, 2021.
- [36] M. Chehimi and W. Saad, "Quantum federated learning with quantum data," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8617–8621.
- [37] Y. Kwak, W. J. Yun, J. P. Kim, H. Cho, J. Park, M. Choi, S. Jung, and J. Kim, "Quantum distributed deep learning architectures: Models, discussions, and applications," *ICT Express*, vol. 9, no. 3, pp. 486–491, 2023.
- [38] M. Chehimi, S. Y.-C. Chen, W. Saad, D. Towsley, and M. Debbah, "Foundations of quantum federated learning over classical and quantum networks," *IEEE Network*, 2023.
- [39] J. Kim, "Quantum federated learning for vehicular computing scenarios," in *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2023, pp. 168–172.
- [40] R. Rofougaran, S. Yoo, H.-H. Tseng, and S. Y.-C. Chen, "Federated quantum machine learning with differential privacy," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 9811–9815.
- [41] 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.
- [42] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.
- [43] M. Chehimi, S. Y.-C. Chen, W. Saad, and S. Yoo, "Federated quantum long short-term memory (fedqlstm)," *Quantum Machine Intelligence*, vol. 6, no. 2, p. 43, Jul 2024. [Online]. Available: <https://doi.org/10.1007/s42484-024-00174-z>
- [44] C. Chu, L. Jiang, and F. Chen, "Cryptoqfl: quantum federated learning on encrypted data," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2023, pp. 1231–1237.
- [45] Z. Qu, L. Zhang, and P. Tiwari, "Quantum fuzzy federated learning for privacy protection in intelligent information processing," *IEEE Transactions on Fuzzy Systems*, 2024.