Asynchronous and Freshness Quantum Federated Learning in Heterogeneous Networks

Lam Nguyen[1], James Hoang[1], Qin Wang[1], Thu Le[2], Lu Qinghua[1], and Chen Shiping[1]
CSRIO Data61[1], Sydney, New South Wales, Australia
[2]University of Science and Technology of Hanoi, Vietnam.

***Abstract*—Quantum Federated Learning (QFL) is a promising approach that leverages quantum computational resources within a distributed federated learning framework, enabling collaborative model training across multiple quantum devices. However, existing QFL methods, such as Slimmable Quantum Federated Learning (SlimQFL), rely on synchronous communication rounds that require all devices to update the global model simultaneously, leading to inefficiencies like idle times and the straggler problem in practical deployments. To address these challenges, in this paper, we propose Asynchronous SlimQFL (`ASQFL`), a novel QFL protocol that allows devices to perform local training and communicate their model parameters to the server asynchronously. `ASQFL` introduces a freshness-based aggregation strategy to prioritize recent updates and a staleness mitigation mechanism to handle delayed updates, ensuring robust convergence even under heterogeneous device and network conditions.**

I. Introduction

Quantum Federated Learning (QFL) [1] has gained considerable attention as an emerging field that merges the principles of Quantum Machine Learning (QML) [2] and Federated Learning (FL) [3], paving the way for distributed quantum intelligence. The integration of QML with FL enables the use of Quantum Neural Networks (QNNs) [4] to train models across multiple quantum devices while preserving data privacy by keeping local datasets decentralized. This distributed learning approach holds significant promise in scenarios where quantum computational power is distributed across multiple entities, such as edge computing and multi-site quantum data centers. However, most current QFL frameworks rely on a synchronous communication protocol, where all participating devices must complete their local training and transmit their parameters to a central server within a strict time frame [5]. While effective in homogeneous environments, these synchronous approaches suffer from inefficiencies and slow convergence when deployed in heterogeneous or resource-constrained environments, where device capabilities and network conditions can vary significantly.

One prominent synchronous QFL method is Slimmable Quantum Federated Learning (SlimQFL) [6], which improves communication efficiency by leveraging a dynamic architecture of Quantum Slimmable Neural Networks (QSNNs) [7]. SlimQFL dynamically adapts to communication and energy limitations by selectively training and transmitting different sets of parameters, such as angle and pole parameters, based on the available resources and channel conditions. However, SlimQFL's reliance on synchronized communication rounds introduces a major bottleneck in real-world deployments. Devices with limited computational power or poor network connectivity act as stragglers, causing delays that prevent the global model from updating until all devices have synchronized. This results in prolonged idle times for high-capacity devices and hampers overall convergence, making SlimQFL less practical for large-scale distributed quantum networks with heterogeneous device participation.

To overcome these limitations, we propose Asynchronous SlimQFL (`ASQFL`), a novel strategy of the QFL framework that allows devices to train and transmit their local QSNN parameters to the server asynchronously, without waiting for other devices to complete their training. `ASQFL` introduces a freshness-based aggregation strategy that assigns a dynamic weight to each update based on its reliability, ensuring that newer updates have a greater influence on the global model. Additionally, `ASQFL` includes a staleness mitigation mechanism that compensates for delayed updates, thus reducing the negative impact of stale parameters on the convergence process. By enabling asynchronous updates and using an adaptive aggregation approach, `ASQFL` not only accelerates the convergence speed, but also improves the robustness and scalability of QFL in dynamic and heterogeneous environments.

II. Proposed Solution Design

A. Overview of Asynchronous SlimQFL (`ASQFL`)

The main challenge in Quantum Federated Learning (QFL) is synchronizing quantum devices that have heterogeneous computational resources and are connected through unreliable or variable communication channels [8]. To address this, `ASQFL` adopts an **asynchronous communication and aggregation mechanism** that allows each device to update the global model independently based on its local training progress. `ASQFL` improves the standard SlimQFL approach by implementing the following key components:

- **Asynchronous Local Training and Uploading**: Each device trains its local Quantum Neural Network (QSNN) and uploads its parameters to the server as soon as local convergence is achieved or predefined training criteria are met.
- **Freshness-Based Aggregation at the Server**: The global server assigns a **freshness score** to each incoming update based on the time elapsed since the previous global

update. This score controls the weight of each update in the aggregation process.
- **Staleness Mitigation and Partial Aggregation**: The server dynamically adjusts the influence of significantly delayed updates and can perform partial aggregation when enough updates are received.

B. Mathematical Formulation of `ASQFL`

The `ASQFL` protocol is built on a foundation of asynchronous gradient updates, where local devices optimize their QSNN parameters independently and transmit them to the server without requiring global synchronization. Let us define the main components of the model:

- **Global Model Parameters**: The global QSNN model at time t is represented as $\Theta_t = \{\theta_t, \phi_t\}$, where θ_t and ϕ_t are the **pole parameters** and **angle parameters** of the Quantum Neural Network (QNN), respectively.
- **Local Device Parameters**: Each device n maintains its own version of the model, denoted as $\Theta_t^n = \{\theta_t^n, \phi_t^n\}$. The device performs local training to minimize its local loss function $L_n(\Theta_t^n)$, calculated as:
$$L_n(\Theta_t^n) = E_{(x,y)\sim D_n}\left[\ell\left(f(x;\Theta_t^n), y\right)\right],$$
where D_n is the local dataset of device n, $f(x;\Theta_t^n)$ is the output of the QSNN, and $\ell(\cdot)$ is the loss function (e.g., mean squared error or cross-entropy).
- **Local Gradient Descent Update**: Each device updates its local parameters using stochastic gradient descent:
$$\Theta_t^n \leftarrow \Theta_t^n - \eta_n \nabla_\Theta L_n(\Theta_t^n),$$
where η_n is the learning rate of device n.
- **Asynchronous Transmission to the Server**: After a local update, each device transmits its local parameters Θ_t^n to the server asynchronously.

C. Freshness-Based Aggregation at the Server

The server receives local updates from devices at different times. To effectively aggregate these updates, we define a **freshness score** $F_n(t)$ for each device n based on the time T_n when its update is received. The freshness score is calculated as:

$$F_n(t) = \exp\left(-\lambda(t - T_n)\right),$$

where t is the current global time, T_n is the time of the latest update from device n, and λ is a decay constant that controls the rate at which the freshness score decreases. The freshness score reflects the recency of the update: a higher value indicates a fresher update, while a lower value suggests a stale update.

D. Global Model Update with Freshness Weighting

Given the local parameters $\Theta_t^n = \{\theta_t^n, \phi_t^n\}$ received from multiple devices, the server aggregates these parameters using a **freshness-weighted average**:

$$\Theta_t \leftarrow \Theta_t + \eta \cdot \sum_{n=1}^{N} F_n(t) \cdot \left(\Theta_t^n - \Theta_t\right),$$

where η is the global learning rate, and $F_n(t)$ is the freshness score for device n at time t. This aggregation ensures that recent updates have a stronger influence on the global model, while stale updates are weighted down to mitigate their impact. To separate the impact of the two types of parameters (angle and pole), the aggregation can be split into two distinct updates:

$$\theta_t \leftarrow \theta_t + \eta \cdot \sum_{n=1}^{N} F_n(t) \cdot \left(\theta_t^n - \theta_t\right),$$

$$\phi_t \leftarrow \phi_t + \eta \cdot \sum_{n=1}^{N} F_n(t) \cdot \left(\phi_t^n - \phi_t\right).$$

E. Staleness Mitigation with Temporal Decay

For significantly delayed updates (e.g., due to network interruptions or high latency), `ASQFL` applies a **temporal decay penalty** to reduce their contribution to the global model. Let $\Delta t = t - T_n$ be the time difference between the global time and the update time of device n. We define a staleness factor $S_n(\Delta t)$ as:

$$S_n(\Delta t) = \frac{1}{1 + \gamma \cdot \Delta t},$$

where γ is a scaling factor that controls the degree of penalty applied to delayed updates. The staleness factor is used to further adjust the freshness score:

$$\tilde{F}_n(t) = F_n(t) \cdot S_n(\Delta t).$$

The updated global model parameters are then computed using $\tilde{F}_n(t)$ instead of $F_n(t)$:

$$\Theta_t \leftarrow \Theta_t + \eta \cdot \sum_{n=1}^{N} \tilde{F}_n(t) \cdot \left(\Theta_t^n - \Theta_t\right).$$

F. Partial Aggregation and Early Stopping

To further enhance efficiency, the server can perform **partial aggregation** if a majority of devices have sent their updates, without waiting for every device to complete its training. Define a participation ratio $P(t)$ as the fraction of devices that have sent updates up to time t:

$$P(t) = \frac{1}{N} \sum_{n=1}^{N} 1(T_n \leq t),$$

where $1(\cdot)$ is an indicator function that returns 1 if the condition is true. If $P(t) \geq \alpha$, where α is a predefined participation threshold (e.g., $\alpha = 0.7$), the server performs a global update using the available updates, thereby reducing idle times.

G. Convergence Analysis

The convergence of ASQFL depends on the rate of asynchronous updates and the balance between fresh and stale parameters. Define the global loss function as:

$$L(\Theta_t) = \sum_{n=1}^{N} w_n \cdot L_n(\Theta_t),$$

where w_n are the weights assigned based on the freshness and staleness factors. We can show that ASQFL converges under standard assumptions of smoothness and convexity by bounding the expected decrease in $L(\Theta_t)$ after each asynchronous update:

$$E\left[L(\Theta_{t+1})\right] \leq L(\Theta_t) - \eta \cdot \left\| \sum_{n=1}^{N} F_n(t) \nabla_{\Theta} L_n(\Theta_t^n) \right\|^2.$$

This inequality guarantees that the global model converges as long as the cumulative freshness scores are bounded and the staleness penalties are properly controlled. The proposed ASQFL framework enhances the efficiency and robustness of quantum federated learning by enabling asynchronous updates, employing freshness-based aggregation, and mitigating staleness through temporal decay mechanisms.

III. Performance Evaluation

We evaluated the performance of the ASQFL framework through simulations on a binary classification task. Using the *sklearn*[1], we generated a dataset of 100 samples with 5 features, distributed across 10 devices to simulate a federated learning setup. Each device trained a linear model with *5* weights, and the global model aggregated updates asynchronously, applying a freshness score and staleness mitigation via temporal decay. The simulation run for *1000* epochs with a global learning rate of *0.01*, a freshness decay rate of *0.1*, and a staleness factor of *0.05*. Key metrics tracked included the convergence of global model weights and the magnitude of weight changes, which reflect the model's learning and stability.

As shown in the Fig. 1, the convergence of the global model's weights was observed by tracking the evolution of the *5* trainable weights over the course of the *1000* training epochs. Initially, the weights fluctuated significantly, as the global model aggregated asynchronous updates from the randomly initialized local models on the devices. However, as the simulation progressed, the weights began to stabilize. The convergence behavior is a direct result of the collaborative learning across devices, where local models improve as they train on their respective datasets, and the server aggregates these updates in an asynchronous manner. The weights followed a pattern of gradual adjustment, with smaller changes occurring in later epochs as the model approached convergence (Fig. 2). This behavior indicates that the model successfully learned from the distributed data and that the asynchronous updates, along with freshness weighting and staleness mitigation, contributed to stable learning dynamics.

[1]https://scikit-learn.org/

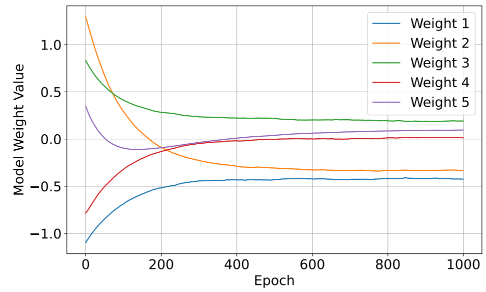

Fig. 1. Covergence of Model Weights

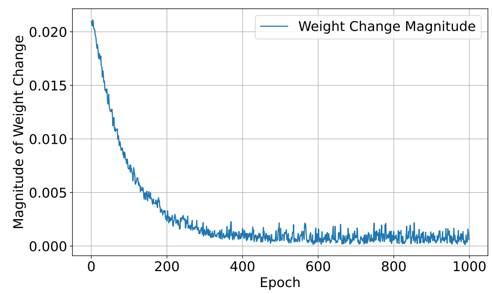

Fig. 2. Magnitude of Weight Changes

IV. Conclusion

In this paper, we proposed an Asynchronous Slimmable Quantum Federated Learning (ASQFL) framework for FL in heterogeneous environments. By leveraging asynchronous updates, freshness weighting, and staleness mitigation, the framework effectively addresses device heterogeneity and communication delays. Our simulations demonstrated successful global model convergence with diminishing weight changes, indicating stable learning.

References

[1] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[2] J. D. Martín-Guerrero and L. Lamata, "Quantum machine learning: A tutorial," *Neurocomputing*, vol. 470, pp. 457–461, 2022.

[3] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–44, 2023.

[4] Y. Du, M.-H. Hsieh, T. Liu, S. You, and D. Tao, "Learnability of quantum neural networks," *PRX quantum*, vol. 2, no. 4, p. 040337, 2021.

[5] R. Huang, X. Tan, and Q. Xu, "Quantum federated learning with decentralized data," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 28, no. 4: Mach. Learn. in Photon. Commun. and Meas. Syst., pp. 1–10, 2022.

[6] W. J. Yun, J. P. Kim, S. Jung, J. Park, M. Bennis, and J. Kim, "Slimmable quantum federated learning," *arXiv preprint arXiv:2207.10221*, 2022.

[7] S. Park, S. Jung, and J. Kim, "Dynamic quantum federated learning for satellite-ground integrated systems using slimmable quantum neural networks," *IEEE Access*, 2024.

[8] 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)*, pp. 8617–8621, IEEE, 2022.