Toward Sustainable Edge Intelligence: Quantum-inspired Spiking Neural Networks-aided Federated Learning Approach

Quan Thanh Dao, *Graduate Student Member, IEEE,* Wei Yang Bryan Lim, *Member, IEEE,* Bradley D. E. McNiven, Berk Canberk, *Senior Member, IEEE,* Nguyen-Son Vo, Hyundong Shin, *Fellow, IEEE,* and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—With the rapid expansion of edge devices, federated learning (FL) has emerged as a critical paradigm for enabling collaborative intelligence while preserving data privacy. However, deploying deep learning models on resource-constrained edge hardware remains challenging due to high computational and energy demands. To address this, we propose a novel quantum-inspired neuromorphic federated learning framework, called FL-quantum leaky integrate-and-fire (FL-QLIF). At the core of our approach is an enhanced QLIF neuron model that utilizes quantum rotation gates to simulate synaptic integration and membrane potential leakage simultaneously, offering a mathematically robust and computationally efficient alternative to traditional activation functions. We rigorously validate the framework through extensive experiments, demonstrating that its convergence behavior closely aligns with theoretical predictions regarding learning rates and local update steps. Extensive experiments on MNIST, KMNIST, and FashionMNIST datasets demonstrate that FL-QLIF achieves test accuracy comparable to classical artificial neural networks (ANNs) while significantly outperforming standard variational quantum neural networks (QNNs), particularly in statistically heterogeneous (non-iid) environments. Furthermore, our energy analysis reveals that the proposed framework delivers substantial efficiency gains, reducing energy consumption by $4.4\times$ to $10\times$ compared to ANN baselines. Finally, we validate the system's scalability, showing robust performance even under low client participation rates (10%), establishing FL-QLIF as a viable solution for sustainable, privacy-preserving edge intelligence.**

***Index Terms*—Edge intelligence, federated learning, neuromorphic computing, quantum-inspired computing, spiking neural networks.**

Quan T. Dao is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1B 3X5, Canada (e-mail: qtdao@mun.ca).

W. Y. B. Lim is with Nanyang Technological University (NTU), Singapore (e-mail: bryan.limwy@ntu.edu.sg).

B. D. E. McNiven is with Compute Everything Technologies Ltd., St. John's, Newfoundland and was with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, (e-mail: mcnivenbrad@gmail.com).

B. Canberk is with the School of Engineering and Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, U.K., (e-mail: b.canberk@napier.ac.uk).

N.-S. Vo is with the Institute of Fundamental and Applied Sciences, Duy Tan University, Ho Chi Minh City, 70000, Vietnam, and also with the Faculty of Electrical-Electronic Engineering, Duy Tan University, Da Nang, 50000, Vietnam (e-mail: vonguyenson@duytan.edu.vn).

H. Shin is with Department of Electronics and Information Convergence Engineering, Kyung Hee University, 1732 Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Korea (e-mail: hshin@khu.ac.kr).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, and with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT7 1NN Belfast, U.K., and also with the Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 17104, South Korea (e-mail: tduong@mun.ca).

The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941, and in part by the NSERC CREATE program (Grant number 596205-2025). The work of B. Canberk is partially supported by The Scientific and Technological Research Council of Türkiye (TÜBİTAK) 1515 Frontier R&D Laboratories Support Program for Türk Telekom 6G R&D Lab under project number 5249902. The work of H. Shin was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2025-00556064), and by the Ministry of Science and ICT (MSIT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2025-RS-2021-II212046), supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

This paper has been submitted in part for presentation at IEEE Global Communications Conference (GLOBECOM), Macau S.A.R., China, December 2026.

Corresponding auhors: Trung Q. Duong and Hyundong Shin.

I. Introduction

With the rapid increase in mobile and edge computing devices, the amount of data generated at the network edge has grown significantly [1]. Traditional computing methods, where data is sent to remote cloud servers for processing [2], are becoming less effective. Issues like data privacy, high communication costs, delays, limited bandwidth, and security risks greatly reduce the efficiency, scalability, and reliability of these centralized systems [3] [4]. To address these critical challenges, federated learning (FL) has emerged as a promising paradigm [5]. FL enables distributed devices (clients) to collaboratively train a shared global model under the coordination of a central server while keeping all training data localized [6]. In each global round, client devices download the current global model, perform several steps of local optimization using their private data, and then upload only the updated model parameters or gradients to the central server for aggregation. This process continues iteratively until convergence. Such a decentralized architecture allows learning directly at the data source without exposing raw information, thereby enhancing privacy and reducing the need for costly data transmission. Many variants of FL have been proposed, ranging from network topology architectures to advanced optimization strategies. In terms of structure, centralized FL offers stable convergence. In contrast, fully decentralized FL [7] removes the central server and allows direct communication among clients. This approach helps mitigate

the impact of stragglers and single-point failures [8]. Besides, semi-decentralized FL (SD-FL) [9]–[11] has been introduced as a hybrid approach that balances the coordination efficiency of centralized schemes with the resilience and flexibility of decentralized ones. Besides these topological innovations, significant research has focused on optimization strategies to address statistical heterogeneity. While federated averaging (FedAvg) remains the standard baseline, advanced algorithms, such as FedProx [12], FedAvgM [13], and FedAdam [14], have been developed to stabilize convergence in non-independent and identically distributed (non-iid) environments. Despite these architectural and algorithmic advancements, the training of deep neural networks in FL still demands substantial computational and energy resources [15]. This high energy consumption poses significant challenges for deploying FL on energy-constrained edge devices, which motivates the exploration of more energy-efficient learning paradigms.

At the same time, increasing attention has been directed toward alternative computational paradigms that can overcome the scalability and energy constraints of conventional digital architectures [16]. Among these, neuromorphic and quantum computing have emerged as two promising directions [17]. Each offers complementary benefits. Neuromorphic computing, inspired by biological neural systems, employs networks of spiking neurons that exchange information through discrete, event-driven signals [18]. This class of models, known as spiking neural networks (SNNs), has attracted significant attention for demonstrating deep-network-level performance while requiring far less computation and exhibiting exceptional energy efficiency [19]. In parallel, quantum computing (QC) leverages fundamental quantum phenomena such as superposition and entanglement to explore exponentially large state spaces, offering massive parallelism unattainable by classical systems [20], [21]. Given these complementary advantages, quantum-enhanced spiking neural networks (QSNNs) have recently been explored as a hybrid framework that inherits the benefits of both. By utilizing spiking dynamics and quantum states, QSNNs can achieve compact representations with high computational expressiveness while maintaining energy-efficient, biologically inspired behavior [17] [22].

Building upon these advances of both FL and alternative computational methods, it becomes natural state to consider federated quantum spiking learning frameworks, where FL provides privacy-preserving and scalable distributed training, while QSNNs contribute quantum-level efficiency and adaptability. Combining FL with QSNNs thus holds the potential to achieve decentralized, energy-efficient, and privacy-aware learning.

A. Related works

1) Quantum federated learning: While classical FL has demonstrated significant potential across a wide range of applications [23]–[29], it remains constrained by challenges such as high computational cost and security of transmitted messages [30]. Consequently, the motivation for quantum FL (QFL) stems from the desire to address these classical limitations by harnessing the unique capabilities of QC, such as superposition and entanglement, to enhance learning efficiency [31]. By synergizing the computational advantages of QC with the decentralized, privacy-preserving architecture of FL, QFL paves the way for building more secure, energy-efficient, and scalable solutions that can overcome the bottlenecks of traditional digital hardware [32]. Pioneering this direction, the authors in [33] established the viability of hybrid quantum–classical FL models by combining pre-trained CNNs for feature extraction with QNNs, laying the foundational steps for the field. Building on this, researchers have developed robust general-purpose frameworks to enable collaborative quantum training. For instance, the FedQNN framework proposed in [34] utilizes a hybrid architecture where classical data is encoded via angle embedding into local variational circuits. By aggregating only updated quantum parameters, this method achieved robust accuracies across genomics and healthcare datasets, with successful validation on real quantum processors. Similarly, the work in [35] introduced QuantumFed, a framework designed to enable multiple quantum nodes to collaboratively train a global model without sharing local data. By optimizing perceptron unitaries via a fidelity-based cost function and transmitting update matrices instead of traditional gradients, this approach demonstrated strong feasibility and robustness against noisy training data using both gradient descent and stochastic strategies. Beyond basic feasibility, recent research has focused on addressing inherent FL challenges such as client heterogeneity and communication bottlenecks. To tackle non-idd data and diverse hardware constraints, the authors in [36] developed a personalized QFL (PQFL) framework. By employing a variational quantum eigensolver (VQE) architecture with specialized regularization, this strategy balances global alignment with client-specific encodings, significantly outperforming state-of-the-art baselines in anomaly detection. Addressing similar challenges, the wp-QFL framework in [37] introduced a weighted personalization strategy to mitigate local model drift. To handle non-iid data distributions in distributed quantum networks, the authors designed a mechanism that dynamically blends global and local quantum circuit parameters (such as rotation angles) using tunable weight factors, alongside a drift-aware update rule based on Euclidean distance metrics. Validated on real IBM quantum processors and simulators like PennyLane and Qiskit, this approach significantly improved convergence stability and accuracy for variational quantum classifiers across diverse datasets. Furthermore, to enhance system efficiency, a communication-efficient QFL framework was proposed in [38]. By extending variational quantum algorithms (VQA) to use tensor networks where local gradients are computed on client devices, this method effectively balances data privacy with computational utility, achieving high accuracy on near-term processors. The versatility of QFL is further evidenced by its successful deployment across diverse critical sectors. In the healthcare domain, the digital twin-assisted QFL model in [39] leverages 5G-enabled digital twins to train personalized variational QNNs, achieving centralized-level accuracy without raw data sharing. In the financial sector, the QFNN-FFD framework [40] secures fraud detection by optimizing local quantum circuits with rotation and entanglement gates,

demonstrating exceptional resilience against hardware noise such as bit-flip errors. Applications extend to smart grids, where a QFL-based dynamic security assessment (QFLDSA) framework [41] integrates hybrid quantum-classical learning to process grid dynamics locally, significantly outperforming traditional methods in dynamic security assessment. Finally, in the internet of vehicles (IoV), the QFSM model [42] improves speech emotion recognition using quantum minimal gated units, surpassing classical LSTM and GRU baselines in speed, privacy, and noise robustness. Despite its potential, the field of QFL remains in its infancy and faces significant hurdles, primarily stemming from early-stage quantum hardware limitations such as low qubit counts, short coherence times, and high susceptibility to noise. Furthermore, practical deployment is complicated by the need to manage model and data heterogeneity across participating devices while ensuring seamless interoperability between quantum and classical systems. Consequently, addressing these technical constraints, alongside standardization and ethical considerations, is essential for the evolution and maturity of QFL.

2) Federated learning utilizing neuromorphic computing: The integration of neuromorphic computing with FL offers distinct advantages, particularly when leveraging SNNs. This synergy enables highly efficient, low-power collaborative training [43]. Establishing the baseline for this integration, the authors in [44] introduced the first on-device collaborative training of SNNs, demonstrating the feasibility of low-power neuromorphic edge intelligence on dynamic vision datasets. Moreover, the study in [45] provided comprehensive benchmarks, showing that SNNs could outperform traditional ANNs in accuracy across hundreds of clients while achieving substantial energy savings through sparse computation. To address data heterogeneity, the framework proposed in [15] demonstrated that federated neuromorphic learning could maintain comparable accuracy under uneven data distributions while significantly reducing data traffic and latency. Furthermore, addressing the physical layer challenges of wireless edge networks, the authors in [46] proposed the adaptive memristor neuron encoding-decoding (AMNED) framework. By integrating SNNs into the over-the-air computation (AirComp) pipeline, this model employs a memristor-based mechanism to encode continuous updates into robust discrete spike trains, effectively mitigating channel noise and minimizing energy consumption. Building on these foundational capabilities, SNN-based FL has been successfully adapted for diverse practical applications. In the realm of intelligent transportation systems, the work in [47] utilized FL-SNN frameworks to enable efficient, noise-resistant traffic sign recognition as a low-power alternative to CNNs. Similarly, the authors in [48] proposed a lightweight spectrum detection framework for cognitive vehicular networks (CVNs). By leveraging the event-driven nature of SNNs, their model performs energy-efficient, on-device sensing that reduces computational complexity for vehicular units. Application-specific studies also show the practical value of SNNs. In aerial networks, the authors in [49] introduced a decentralized FL framework for UAV swarms driven by SNNs. This approach combines low-power local training with an intelligent leader election mechanism based on Bayes' theorem, significantly outperforming ANN-based methods in terms of energy efficiency and transmission latency for real-time decision-making. In the domain of sensing, the authors in [50] developed a privacy-preserving FL framework for radar gesture recognition leveraging SNNs trained via spike timing-dependent plasticity (STDP). To address edge resource constraints, the method incorporates weight pruning during local training, demonstrating competitive accuracy on radar datasets while significantly reducing communication overhead and energy usage compared to non-spiking baselines.

3) Quantum-inspired neuromorphic models: Beyond federated settings, many studies have investigated the intersection of quantum and spiking computation to exploit complementary strengths of both paradigms [51]. Early research focused on leveraging quantum principles to enhance the generalization and efficiency of classical SNNs. For instance, the work in [52] introduced quantum superposition–inspired SNNs, utilizing quantum image encoding and superposition states to improve the recognition of complex visual patterns. Building on this direction, the parallel proportional fusion QSNN (PPF-SQNN) framework proposed in [53] combined classical SNNs with variational quantum circuits through a parallel proportional fusion strategy, demonstrating high accuracy and noise immunity in image classification. Similarly, the deep spiking QNN (DSQ-Net) model presented in [54] offered a hybrid quantum–classical approach that improved optimization efficiency on noisy intermediate-scale quantum devices, outperforming classical baselines on chaotic datasets. Furthermore, addressing computational complexity, the authors in [55] introduced a quantum algorithm for accelerating SNNs, achieving logarithmic–polynomial complexity while maintaining high classification accuracy. Parallel to architecture-level innovations, significant efforts have been made to model biological neuronal dynamics using quantum circuits. The authors in [56] developed the quantum biological neuron (QBN), which models dendrite, soma, and synapse dynamics via quantum circuits, validating its capability through MNIST classification. In a different approach, the shallow hybrid SQNN proposed in [57] replaced biologically inspired STDP with variational quantum training, yielding superior robustness to noise across multiple benchmarks. A major step toward physically grounded architectures was made by the authors in [22], who proposed the QLIF neuron. This compact circuit uses only two rotation gates to model integration, decay, and reset processes. However, this early iteration performed integration and leakage sequentially rather than concurrently, limiting its biological fidelity during dynamic input streams. Recent works have addressed these limitations by integrating QLIF neurons into more complex frameworks for advanced tasks. The QSNN architecture proposed in [58] synergizes VQC with QLIF neurons to enhance multiclass classification. In this hybrid framework, classical data is mapped to quantum states via amplitude encoding, and expectation values from the VQC are fed into QLIF neurons to generate spikes, significantly outperforming traditional models on datasets like MNIST and Fashion-MNIST. Extending this to the biological domain, the authors in [59] proposed a quantum neuromorphic framework for classifying EEG brain signals. By integrating a QLIF neuron into a 3D SNN architecture,

this model effectively handles complex spatio-temporal data, demonstrating superior performance in analyzing experimental wrist movement recordings compared to classical counterparts.

B. Motivation and contributions

Although recent efforts have advanced federated quantum and neuromorphic learning [15], [33]–[39], [41], [42], [42], [44]–[50], their joint integration remains in an early stage. While the FL-QDSNNs framework [60] demonstrated the feasibility of combining FL with quantum spiking neural networks, it exhibits several key limitations. Primarily, it operates at a gate-level abstraction without an explicit neuron model, thereby failing to accurately simulate fundamental biophysical properties such as simultaneous integration of membrane potential and leaky decay. Furthermore, its federated aggregation relies on complex circuit parameters rather than physically meaningful neuron dynamics, which can complicate integration with advanced FL optimization strategies. Lastly, the study omits necessary comparisons with established quantum and classical baselines, preventing a comprehensive evaluation of its relative performance. Likewise, existing QFLs achieve privacy-preserving training via variational circuits but incur slow convergence. Meanwhile, the QLIF neuron [22] offers an efficient, noise-tolerant quantum spiking model, yet its sequential integration–leak process. Moreover, most prior quantum-neuromorphic studies [15] [44] [45] [47] rely exclusively on standard FedAvg. Few attempts have been made to comprehensively evaluate advanced variants, which are essential to address statistical heterogeneity. Taking these limitations into account, in this paper, we develop an enhanced QLIF neuron that unifies integration and leaky decay within a single rotation expression, ensuring both biophysical realism and computational stability. Furthermore, we embed this improved neuron into an FL pipeline to construct a scalable and privacy-preserving learning framework suitable for non-iid data. In particular, our main contributions are summarized in Tab. I and listed as follows:

- We introduce FL-QLIF, a first FL framework that integrates the proposed QLIF neuron. This neuron employs quantum rotation gates to model integration and leakage simultaneously, while remaining executable on classical neuromorphic hardware without the need for physical quantum devices.
- We conduct an extensive evaluation across diverse data environments (iid and non-iid) using the MNIST, FashionMNIST, and KMNIST datasets. We rigorously benchmark performance against competitive baselines, including QNN and ANN, as well as various federated strategies. Finally, we explicitly quantify the energy efficiency, demonstrating that QLIF achieves accuracy comparable to ANNs with significantly lower computational cost.
- We provide a theoretical convergence analysis of the proposed framework under relaxed semi-smoothness conditions, deriving critical bounds for the learning rate and local epochs. We further corroborate these theoretical findings through empirical experiments, confirming the impact of hyperparameters on the stability and convergence speed of the quantum-inspired model.

TABLE I: Overview of prior studies and this work.

Studies	FL	QC	SNNs
[33]–[42]	✓	✓	×
[15], [44]–[50]	✓	×	✓
[22], [52]–[59]	×	✓	✓
This work	✓	✓	✓

C. Paper structure

The remainder of this paper is organized as follows. Section II provides essential background on FL and the conventional LIF spiking neuron model relevant to this work. Section III introduces the proposed FL, while Section IV presents a convergence analysis of federated learning under relaxed conditions. Section V reports experimental results Section VI concludes the paper. Furthermore, the notations used in this paper are summarized in Tab. II

TABLE II: Summary of main notations

Notation	Description
$F(\mathbf{w})$	Global loss function
$F_c(\mathbf{w})$	Local loss function for node c
$\mathbf{w}_{c,k}^t$	Local model parameter at node c in local iteration k during round t
$\mathbf{w}_t$	Global model parameter in iteration t
$\mathbf{w}^*$	True optimal model parameter that minimizes $F(\mathbf{w})$
η	Gradient descent step size
K	Number of local update epochs at each node
n_c	Number of samples held by client c.
N	Total number of samples.
$\mathcal{C}$	Set of participating clients.
p_c	Data-proportional client weight.
ξ_p	Weighted drift during local updates.
$\rho[t]$	Excited-state population (membrane potential analog).
θ	Quantum rotation angle for input intensity.
$\gamma[t]$	Negative rotation modeling leaky decay.
$\phi[t]$	Restoring rotation for previous excited state.
$X[t]$	Input spike signal.
$S_{\text{out}}[t]$	Output spike at time t.
U_{thr}	Spiking threshold.
$U[t]$	Membrane potential at time t.
β_{mem}	Membrane potential decay rate.
τ_{mem}	Membrane time constant in RC circuit.
α_{dir}	Dirichlet concentration (data heterogeneity).

II. Background

A. Federated learning

FL is a new approach to distributed machine learning that balances the need for collaboration and data privacy [5]. Instead of sharing raw data, FL lets different clients (denoted by $\mathcal{C}$) work together to build a shared model by sending only model updates to a central server, keeping their local data private. For each data sample j at client c with features x_j

and label y_j, define the per–sample loss $f_j(\mathbf{w}) = \ell(\mathbf{w}; x_j, y_j)$, where w represents the model parameter vector. Let $\mathcal{D}_c$ be client c's dataset with size $n_c = |\mathcal{D}_c|$. The local empirical loss at client c is given by

$$F_c(\mathbf{w}) \triangleq \frac{1}{n_c} \sum_{j \in \mathcal{D}_c} f_j(\mathbf{w}). \tag{1}$$

Let $N = \sum_{c \in \mathcal{C}} n_c$ be the total number of samples across all clients. The global objective minimized by FL is the sample–size–weighted average

$$F(\mathbf{w}) = \sum_{c \in \mathcal{C}} \frac{n_c}{N} F_c(\mathbf{w}). \tag{2}$$

Note that $F(\mathbf{w})$ cannot be evaluated at the server without accessing all local data.

Regarding the training process, at the start of communication round t, the server broadcasts the current global model $\mathbf{w}_t$ to the participating clients. Each client c initializes $\mathbf{w}_{c,0}^t = \mathbf{w}_t$ and performs K local update epochs

$$\mathbf{w}_{c,k+1}^t = \mathbf{w}_{c,k}^t - \eta \nabla F_c\big(\mathbf{w}_{c,k}^t\big), \qquad k = 0, \ldots, K-1. \tag{3}$$

The client's output after K epochs is given by

$$\mathbf{w}_c^{t+1} \triangleq \mathbf{w}_{c,K}^t. \tag{4}$$

The server aggregates the returned local models using the same sample–size weights as in Eq. (2), given by

$$\mathbf{w}_{t+1} = \sum_{c \in \mathcal{C}} \frac{n_c}{N} \mathbf{w}_c^{t+1}, \tag{5}$$

and then broadcasts $\mathbf{w}_{t+1}$ to clients. In this scheme, as the number of communication rounds increases, the model asymptotically converges to the optimal global solution $\mathbf{w}^*$.

B. FL variants

1) FedProx: FedProx [12] extends the FedAvg framework to explicitly address the challenges arising from statistical data heterogeneity and partial client participation (stragglers). It introduces a proximal term to the local training objective that penalizes the divergence of the local model parameters from the global model. By regularizing the local updates, this modification ensures that the client models do not stray too far from the global parameters during local training. The modified local objective for client c is defined as

$$\min_{\mathbf{w}} F_c(\mathbf{w}) + \frac{\mu}{2} ||\mathbf{w} - \mathbf{w}_t||^2, \tag{6}$$

where μ is the proximal coefficient controlling the strength of the regularization. While the server aggregation step remains identical to FedAvg, this constraint on the local objective effectively limits client drift, resulting in a more stable global model in non-iid environments.

2) FedAvgM: FedAvgM [13] introduces a momentum buffer to the server to dampen oscillations caused by diverse client data distributions. To formulate this, we first define the global pseudo-gradient Δ_t at round t as the difference between the current global model and the weighted average of client updates

$$\Delta_t = \mathbf{w}_t - \sum_{c \in \mathcal{C}} \frac{n_c}{N} \mathbf{w}_c^{t+1}. \tag{7}$$

Using Δ_t, the server accumulates the update history in a momentum buffer v_t rather than updating w_{t+1} directly. The update rule is defined as

$$\begin{aligned} \mathbf{v}_{t+1} &= \beta \mathbf{v}_t + \Delta_t, \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \mathbf{v}_{t+1}, \end{aligned} \tag{8}$$

where β is the momentum parameter and $\mathbf{v}_0$ is initialized to zero.

3) FedAdagrad: To address convergence issues arising from sparse data features, FedAdagrad [14] adapts the learning rate for each parameter dimension. Using the pseudo-gradient Δ_t defined in Eq. 7, the server accumulates past squared gradients in a state vector $\mathbf{v}_t$ is written as

$$\begin{aligned} \mathbf{v}_{t+1} &= \mathbf{v}_t + (\Delta_t)^2, \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \frac{\eta_g}{\sqrt{\mathbf{v}_{t+1}} + \epsilon} \odot \Delta_t, \end{aligned} \tag{9}$$

where $\odot$ represents element-wise multiplication, η_g is the server learning rate, and ϵ is a stability constant.

4) FedAdam: FedAdam [14] combines the benefits of momentum and adaptive scaling at the server level. It utilizes Δ_t to maintain both a first moment estimate $\mathbf{m}_t$ and a second moment estimate $\mathbf{v}_t$ as

$$\begin{aligned} \mathbf{m}_{t+1} &= \beta_1 \mathbf{m}_t + (1 - \beta_1)\Delta_t, \\ \mathbf{v}_{t+1} &= \beta_2 \mathbf{v}_t + (1 - \beta_2)(\Delta_t)^2, \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \frac{\eta_g}{\sqrt{\mathbf{v}_{t+1}} + \epsilon} \odot \mathbf{m}_{t+1}, \end{aligned} \tag{10}$$

where β_1, β_2 are decay rates and ϵ is an adaptivity parameter.

5) FedYogi: FedYogi [14] is a variant of FedAdam designed to handle non-convex settings more effectively by controlling the growth of the effective learning rate. It modifies the second moment update ($\mathbf{v}_t$) to be additive, preventing the preconditioner from changing too rapidly when client updates exhibit high variance

$$\begin{aligned} \mathbf{v}_{t+1} &= \mathbf{v}_t - (1 - \beta_2)(\Delta_t)^2 \odot \text{sign}(\mathbf{v}_t - (\Delta_t)^2), \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \frac{\eta_g}{\sqrt{\mathbf{v}_{t+1}} + \epsilon} \odot \mathbf{m}_{t+1}. \end{aligned} \tag{11}$$

C. The conventional LIF neural networks

Among various spiking neuron models [61], the LIF neuron is the most widely used in deep learning since it balances biological plausibility with computational simplicity [62]. The behavior of the biological LIF neuron can be represented by a resistor-capacitor (RC) low-pass filter circuit [63] shown in Fig. 1(a). This circuit accumulates the membrane potential $U(t)$, which increases with incoming current spikes and decays exponentially when no input is present. The following differential equation describes this behavior as

$$\tau_{mem} \frac{dU(t)}{dt} = -U(t) + I_{\text{in}} R, \tag{12}$$

where $\tau_{mem} = RC$ is the time constant of the circuit. In discrete form, the update rule is shown as

$$U[t] = \beta_{mem} U[t-1] + (1-\beta_{mem}) I_{\text{in}}[t], \tag{13}$$

where the decay rate is defined as $\beta_{mem} = \exp(-1/\tau_{mem})$. The "fire" behavior of the LIF neuron arises from the condition that once the membrane potential reaches a threshold value U_{thr}, the neuron emits an output spike and resets its potential. For application in deep learning, the input current $I_{in}[t]$ is modeled as a weighted sum of binary input spikes, $I_{in}[t] = WX[t]$, where W is a trainable weight matrix and $X[t]$ the input spike vector. Substituting this into the update rule yields

$$U[t] = \underbrace{\beta_{mem} U[t-1]}_{\text{decay}} + \underbrace{WX[t]}_{\text{input}} - \underbrace{S_{\text{out}}[t] U_{\text{thr}}}_{\text{reset}}, \tag{14}$$

where the output spike $S_{\text{out}}[t]$ is defined by

$$S_{\text{out}}[t] = \begin{cases} 1, & \text{if } U[t] > U_{\text{thr}} \\ 0, & \text{otherwise} \end{cases}. \tag{15}$$

Regarding the SNNs, this model is the same type of network topology as ANN; however, instead of standard activation functions, SNNs treat a spiking neural model, which is LIF in this paper. The model's input consists of spike signals, which can either come directly from neuromorphic sensors or be converted from static data into spike trains using various existing methods [64]. The architecture of the SNN is depicted in Fig. 1. (b).

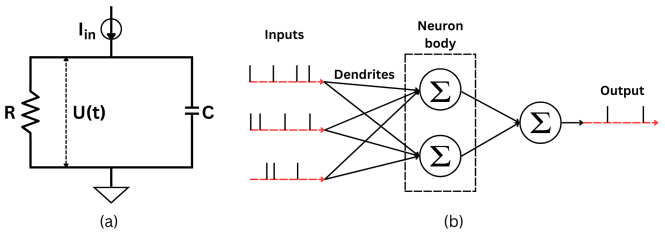

Fig. 1: (a) A resistor-capacitor (RC) low-pass filter circuit. (b) A spiking neural network model.

For model updates, as with traditional machine learning models, there are various learning rules and training methods available to choose [65]. In the model presented here, we have chosen a gradient descent methodology and backpropagation using spikes to train the network [66]. This method works by backpropagating the gradient from the final output through the network. Gradients are computed for all paths, whether or not a neuron fires [67]. In this sense, training an SNN is very similar to training a recurrent neural network (RNN), using repeated applications of the chain rule [62].

Specifically, SNNs need to calculate the gradient of a time step. The effect of $W[s]$ on $\mathcal{L}[t]$ when $s = t$ is called the *immediate influence*. When $s < t$, the influence of $W[s]$ on $\mathcal{L}[t]$ is referred to as the *prior influence*. As the gradient of the loss function with respect to the weights being trained, this can be generally expressed as

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_t \frac{\partial \mathcal{L}[t]}{\partial W} = \sum_t \sum_{s \le t} \frac{\partial \mathcal{L}[t]}{\partial W[s]} \cdot \frac{\partial W[s]}{\partial W}. \tag{16}$$

In the current system, the weight is shared across all time steps, meaning $W[0] = W[1] = \cdots = W$. As a result, any change in $W[s]$ affects all other instances of W equally. This implies that $\frac{\partial W[s]}{\partial W} = 1$, which simplifies Eq. (16) to

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_t \sum_{s \le t} \frac{\partial \mathcal{L}[t]}{\partial W[s]}. \tag{17}$$

For the calculation of the loss at each time step, the gradient can be expanded using the chain rule for all of the connected variables used in the network and expressed as follows:

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial S_{out}} \cdot \frac{\partial S_{out}}{\partial U} \cdot \frac{\partial U}{\partial W}. \tag{18}$$

Nevertheless, the step function defining the spike is non-differentiable, which blocks gradient flow and causes the well-known "dead-neuron" problem. To address this, a widely used solution is the application of *surrogate gradients* [68]. In this method, the forward pass operates as usual, but during backpropagation, the non-differentiable spike function is replaced with a smooth, differentiable approximation that closely mimics spike behavior, thus enabling effective training. Several surrogate gradient functions have been proposed. In the network presented here, we adopt the *arctangent surrogate gradient* [69], which is defined as

$$\frac{\partial S}{\partial U} \approx \frac{\partial \tilde{S}}{\partial U} = \frac{1}{\pi} \cdot \frac{1}{1 + (U\pi)^2}. \tag{19}$$

III. Federated Learning with Quantum-Inspired LIF Neuron Model

In this section, we integrate the proposed quantum-inspired LIF (QLIF) neuron model into the FL framework to form a quantum–inspired neuromorphic learning paradigm.

A. SNN with QLIF neuron model

The method to replicate spiking behavior on quantum hardware is inspired by the work in [22] where a QLIF neuron uses a rotation gate $R_X(\theta)$ to encode an input spike. Based on this mechanism, the model can simulate the integration, firing, and leaky behaviors of spiking neurons.

The $R_X(\theta)$ gate performs a rotation of a qubit's state vector around the X-axis of the Bloch sphere by an angle θ. It is represented by the unitary matrix

$$R_X(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}. \tag{20}$$

As mentioned in the classical LIF model, the membrane potential increases with input current spikes. In the quantum counterpart, however, a spike is represented by applying a rotation matrix $R_X(\theta)$ (represented in Eq. (20)) to a qubit in the ground state $|0\rangle$, where θ corresponds to the input intensity. This results in an excited state population analogous to the membrane potential in classical neurons. After each

time step, regardless of whether a spike was present or not, a measurement is performed to determine the current excited state population. If the calculated measurement exceeds a predefined firing threshold, the neuron emits an output spike and resets the qubit to the ground state $|0\rangle$ (similar to the reset phase in classical LIF neuron model). Otherwise, the neuron continues processing incoming spikes.

The measurement probability of the excited state $|1\rangle$ after rotation is written as

$$\rho = P(|1\rangle) = |\langle 1|R_X(\theta)|0\rangle|^2 = \sin^2\left(\frac{\theta}{2}\right). \tag{21}$$

However, an important difference from the classical model is the bounded nature of quantum rotations. In classical LIF models, the membrane potential can continue to increase with stronger inputs, as described in Eq. (14). However, in the quantum case, rotation gates are cyclical because they repeat every 2π. More importantly, if the rotation angle goes beyond π, ρ starts to decrease instead of increasing due to the nature of the $\sin^2$ function. This constraint introduces non-intuitive effects if not properly addressed, potentially resulting in degraded neuron behavior or misfiring during training. Therefore, it is essential to restrict the rotation angles in QLIF models to the range $[0, \pi)$ to ensure consistent and meaningful state updates across time steps.

After obtaining $\rho[t]$ from the simulated measurement, the model must retain this value as the neuron's state for the next time step. Because our QLIF implementation simulates the qubit independently at each step rather than keeping a persistent quantum state, the circuit does not carry the post-measurement state forward. Instead, we reconstruct a qubit state that encodes the previously computed $\rho[t]$. Since the QLIF model uses only R_X rotation gate, this is achieved by applying a restoring rotation whose angle produces the same excited-state probability

$$\phi[t] = 2\arcsin\left(\sqrt{\rho[t]}\right). \tag{22}$$

The leaky behavior of LIF neurons can be simulated in the quantum version through the quantum system's interaction with its environment property. In particular, the qubit experiences a process called T_1 *relaxation*, where its excited state slowly decays back to the ground state due to environmental noise [70]. This behavior closely matches the gradual leak of membrane potential in classical LIF neurons. After a qubit's excited state has been increased by an R_X gate, this decay can be simulated by applying another R_X gate in the opposite direction. The angle of this reverse rotation, $\gamma[t]$, depends on how much the qubit would have decayed over time as

$$\gamma[t] = -2\arcsin\left(\sqrt{\rho[t]\exp(-\frac{\tau}{T_1})}\right). \tag{23}$$

In the original idea shown in [22], the integration and leaky behavior do not appear at the same time. They depend on whether the spike stimulus is present. However, as shown in the Eq. (14), they should exist simultaneously. This can be expressed as

$$\rho[t+1] = \sin^2\left(\frac{\theta \cdot X[t] + \gamma[t] + \phi[t]}{2}\right). \tag{24}$$

In this formula, the circuit will reinstate the previous state memory, and then implement a negative rotation to implement the exponential state leakage. At the end, depending on whether the spike exists, the positive rotation gate is presented to simulate the stimulus input. Finally, similar to the output in Eq. (15), the output spike is defined as:

$$S_{\text{out}}[t+1] = \begin{cases} 1, & \text{if } \rho[t+1] > U_{\text{thr}} \\ 0, & \text{otherwise} \end{cases}. \tag{25}$$

Regarding the QLIF neuron network, the idea remains the same as the one in conventional SNN, where the loss at each step is defined by

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial S_{out}} \cdot \frac{\partial S_{out}}{\partial \rho} \cdot \frac{\partial \rho}{\partial W}. \tag{26}$$

The weight W includes θ and τ, and the surrogate gradient function changes to

$$\frac{\partial S}{\partial \rho} \approx \frac{\partial \tilde{S}}{\partial \rho} = \frac{1}{\pi} \cdot \frac{1}{1 + (\rho\pi)^2}. \tag{27}$$

This allows gradient-based optimization while maintaining the spiking characteristics of the network.

B. Federated QLIF training

In classical FL, each client $c \in \mathcal{C}$ updates its local model by minimizing the empirical loss $F_c(\mathbf{w})$ using stochastic gradient descent (SGD) shown in Eq. (3). In the proposed system, each client holds a local quantum-inspired SNN whose neurons follow QLIF dynamics rather than conventional activation functions. Firstly, the server initializes and broadcasts a common model $\mathbf{w}_0$ to all clients. At each communication round t, each client performs local training for K epochs on its private dataset $\mathcal{D}_c$, consisting of neuromorphic spike-encoded samples (X_j, y_j). Spike trains are generated by encoding static data via Poisson rate encoding.

The server (or base station) then performs a weighted average across all participating clients using the sample-size weights provided in Eq. 5. Finally, the global server broadcasts the current global QLIF model parameters $\mathbf{w}_{t+1}$ to all participating clients $c \in \mathcal{C}$. The complete architecture of the proposed FL-QLIF framework is depicted in Fig. 2. Furthermore, Algorithm 1 summarizes the training process.

Algorithm 1 Training FL quantum-inspired SNNs framework

Initialize global model $\mathbf{w}_0$ and broadcast to clients
for each round $t = 1, 2, \ldots$ **do**
  for each client c **in parallel do**
    $\mathbf{w}_c^{t+1} \leftarrow \text{ClientUpdate}(c, \mathbf{w}_t)$
  end for
  $\mathbf{w}_{t+1} \leftarrow \sum_{c \in \mathcal{C}} \frac{n_c}{N} \mathbf{w}_c^{t+1}$
  Broadcast $\mathbf{w}_{t+1}$ to clients
end for

ClientUpdate(c, w):
for each local epoch $i = 1$ to K **do**
  Generate spike train using rate encoding
  for each time step of spike trains **do**
    Compute the instantaneous loss
  end for
  $\nabla \ell(\mathbf{w}) \leftarrow$ Accumulated loss over all steps
  $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla \ell(\mathbf{w})$
end for
return $\mathbf{w}$ to server

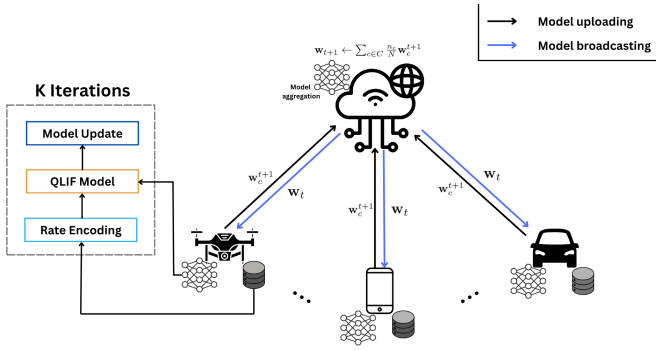

Fig. 2: The complete system architecture of the proposed FL-QLIF framework.

IV. Convergence analysis for Federated Learning under weaker conditions

A. Assumptions

In this section, we define three conditions for proving the convergence. We replace the traditional smoothness condition with semi-smoothness. The no critical point condition simplifies the bounded gradient assumption used in prior federated learning analyses. The semi-Lipschitz condition relaxes the classical Lipschitz property.

1) Smoothness property: As established in Section III, QLIF-based networks represent a variant of ANNs where classical neurons are replaced by quantum-inspired spiking units while preserving the standard feedforward architecture and gradient-based training paradigm. In traditional neural network optimization, the loss function is commonly assumed to satisfy L-smoothness. However, this assumption is incompatible with the proposed framework due to the non-differentiable Heaviside step function (Eq. (25)). Although surrogate gradients (Eq. (27)) enable backpropagation through the network, the true forward pass remains inherently non-smooth. Consequently, rather than adopting the standard smoothness assumption, we characterize the QLIF optimization landscape using the semi-smoothness condition [71]. This assumption provides a more realistic framework while preserving the theoretical machinery needed for convergence analysis.

We define (a, b)-semi-smoothness to capture the behavior of functions F over parameters $\mathbf{w}$ and $\mathbf{w}'$, where an additional linear term enhances adaptability to non-smooth models.

Definition 1. For any function F, we say it is (a, b)-semi-smooth if for any $\mathbf{w}, \mathbf{w}'$,

$$F(\mathbf{w}) \leq F(\mathbf{w}') + \langle \nabla F(\mathbf{w}), \mathbf{w} - \mathbf{w}' \rangle + b \left\| \mathbf{w} - \mathbf{w}' \right\|^2 + a \left\| \mathbf{w} - \mathbf{w}' \right\| \cdot F(\mathbf{w})^{1/2}.$$

Noting that, unlike traditional smoothness definitions, this includes an extra first-order term $\|\mathbf{w} - \mathbf{w}'\|$ on the right-hand side, reflecting the semi-smooth nature.

2) No critical point: The no critical point property ensures the absence of saddle points or critical points near random initialization, thereby guaranteeing continuous optimization progress.

Definition 2. Let $\mathcal{W}$ be a neighbor set of $\mathbf{w}^*$ (a minimum of F). We say there is no critical point for a function F, there exist constants $0 < \tau_1 < 1$ and $\tau_2 > 0$, if for any $\mathbf{w} \in \mathcal{W}$,

$$\tau_1^2 \cdot F(\mathbf{w}) \leq \|\nabla F(\mathbf{w})\|^2 \leq \tau_2^2 \cdot F(\mathbf{w}).$$

This condition provides a two-sided guarantee: the lower bound ensures that the gradient magnitude remains proportional to the loss value, preventing the optimizer from stalling at spurious critical points, while the upper bound prevents gradient explosion by ensuring that gradient norms scale reasonably with the loss. For QLIF networks specifically, this assumption is well-justified by several structural properties. First, the bounded quantum rotation angles $\theta \in [0, \pi)$ naturally satisfy the upper bound by preventing unbounded parameter growth. Second, the surrogate gradient mechanism (Eq. (27)) ensures non-zero gradients throughout training even when discrete spikes do not occur. Third, the spiking dynamics and reset mechanism create an inherently regularized loss landscape where gradient vanishing is rare. As long as the network produces classification errors ($F(\mathbf{w}) > 0$), spike timing mismatches generate informative gradient signals that drive optimization progress.

3) The semi-Lipschitz: Semi-Lipschitzness bounds the change in gradients when the parameters move in the optimization landscape, and is essential for controlling local-client gradient drift in FL. Classical smoothness forces gradients to behave at most linearly with the parameter difference, but semi-Lipschitzness is more flexible because it permits a small correction that depends on the loss value. This flexibility is important because the model's nonlinearities do not satisfy global Lipschitz gradient smoothness.

Definition 3. For a function F, we say it is (α, β)-semi-

Lipschitz, if for any $\mathbf{w}, \mathbf{w}'$,

$$\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')\|^2 \leq \beta^2 \, \|\mathbf{w} - \mathbf{w}'\|^2 + \alpha^2 \, \|\mathbf{w} - \mathbf{w}'\| \cdot \max\{F(\mathbf{w})^{1/2}, F(\mathbf{w}')^{1/2}\}.$$

B. Convergence analysis

To simplify the analysis, we assume full client participation in each communication round in the FedAvg setting.

a) Start from semi-smoothness: The (a, b)–semi-smoothness of F gives

$$\begin{aligned} F(\mathbf{w}_{t+1}) - F(\mathbf{w}^*) \;\leq\; & F(\mathbf{w}_t) - F(\mathbf{w}^*) \\ & + \langle \nabla F(\mathbf{w}_t), \, \mathbf{w}_{t+1} - \mathbf{w}_t \rangle \\ & + b \, \|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 + a \, \|\mathbf{w}_{t+1} - \mathbf{w}_t\| \, F(\mathbf{w}_t)^{1/2}, \end{aligned} \tag{28}$$

and we write $\Delta \mathbf{w}_t := \mathbf{w}_{t+1} - \mathbf{w}_t$ for short.

b) Express the round update and isolate the coupling term: Let define a data-proportional client weights $p_c := n_c/N$, the $\Delta \mathbf{w}_t$ is defined as a weighted average of local gradients

$$\begin{aligned} \Delta \mathbf{w}_t &= -\eta \sum_c p_c \sum_{k=1}^{K} \nabla F_c\big(\mathbf{w}_{c,k-1}^t\big) \\ &= -\tilde{\eta} \sum_{c,k} q_{c,k} \, g_{c,k-1}^t, \end{aligned} \tag{29}$$

where $q_{c,k} := \frac{p_c}{K}$, $\tilde{\eta} := K\eta$. By substituting Eq. (29) into the coupling term, we have

$$\big\langle \nabla F(\mathbf{w}_t), \Delta \mathbf{w}_t \big\rangle = -\tilde{\eta} \Big\langle \nabla F(\mathbf{w}_t), \sum_{c,k} q_{c,k} \, g_{c,k-1}^t \Big\rangle. \tag{30}$$

Noting that $\sum_k q_{c,k} = p_c$ and $\sum_c p_c \nabla F_c(\mathbf{w}_t) = \nabla F(\mathbf{w}_t)$, the second element in Eq. (30) can be written as

$$\begin{aligned} \sum_{c,k} q_{c,k} \, g_{c,k-1}^t &= \sum_{c,k} q_{c,k} \left(g_{c,k-1}^t - \nabla F_c(\mathbf{w}_t) \right) \\ &\quad + \sum_c \Big(\sum_k q_{c,k} \Big) \nabla F_c(\mathbf{w}_t) \\ &= E_t + \sum_c p_c \nabla F_c(\mathbf{w}_t) = E_t + \nabla F(\mathbf{w}_t), \end{aligned} \tag{31}$$

where

$$E_t := \sum_{c,k} q_{c,k} \big(g_{c,k-1}^t - \nabla F_c(\mathbf{w}_t) \big).$$

By plugging Eq. (31) into Eq. (30) and using bilinearity, we have

$$\begin{aligned} \big\langle \nabla F(\mathbf{w}_t), \Delta \mathbf{w}_t \big\rangle &= -\tilde{\eta} \Big(\|\nabla F(\mathbf{w}_t)\|^2 + \big\langle \nabla F(\mathbf{w}_t), E_t \big\rangle \Big) \\ &\leq -\frac{\tilde{\eta}}{2} \|\nabla F(\mathbf{w}_t)\|^2 + \frac{\tilde{\eta}}{2} \|E_t\|^2 \\ &\leq -\frac{\tilde{\eta}}{2} \|\nabla F(\mathbf{w}_t)\|^2 \\ &\quad + \frac{\tilde{\eta}}{2} \sum_{c,k} q_{c,k} \big\| g_{c,k-1}^t - \nabla F_c(\mathbf{w}_t) \big\|^2, \end{aligned} \tag{32}$$

where the second step follows from $-\langle x, z \rangle \leq \frac{1}{2}\|x\|^2 + \frac{1}{2}\|z\|^2$ with $x = \nabla F(\mathbf{w}_t)$ and $z = E_t$, and the last step follows from convexity of $\|\cdot\|^2$ (Jensen) for the normalized weights $\sum_{c,k} q_{c,k} = 1$. For each client c, apply the (α, β)–semi-Lipschitz condition and $2ab \leq a^2 + b^2$, we have

$$\begin{aligned} \big\| g_{c,k-1}^t - \nabla F_c(\mathbf{w}_t) \big\|^2 &\leq \beta^2 \|\mathbf{w}_{c,k-1}^t - \mathbf{w}_t\|^2 \\ &\quad + \alpha^2 \|\mathbf{w}_{c,k-1}^t - \mathbf{w}_t\| F_c(\mathbf{w}_t)^{1/2} \\ &\leq \big(\beta^2 + \tfrac{\alpha^2}{2}\big) \, \|\mathbf{w}_{c,k-1}^t - \mathbf{w}_t\|^2 \\ &\quad + \tfrac{\alpha^2}{2} \, F_c(\mathbf{w}_t). \end{aligned} \tag{33}$$

We define the weighted drift as follows:

$$\xi_p := \sum_{c,k} q_{c,k} \, \|\mathbf{w}_{c,k-1}^t - \mathbf{w}_t\|^2. \tag{34}$$

Putting Eq. (33), Eq. (38), and $\sum_c p_c F_c(\mathbf{w}_t) = F(\mathbf{w}_t)$ to Eq. (32), we obtain

$$\begin{aligned} \big\langle \nabla F(\mathbf{w}_t), \Delta \mathbf{w}_t \big\rangle \leq & -\frac{\tilde{\eta}}{2} \|\nabla F(\mathbf{w}_t)\|^2 \\ & + \frac{\tilde{\eta}}{2} \Big(\big(\beta^2 + \tfrac{\alpha^2}{2}\big) \, \xi_p + \tfrac{\alpha^2}{2} \, F(\mathbf{w}_t) \Big). \end{aligned} \tag{35}$$

c) Bounding drift ξ_p:

$$\begin{aligned} \|\mathbf{w}_{c,k+1}^t - \mathbf{w}_t\|^2 &= \|\mathbf{w}_{c,k}^t - \mathbf{w}_t - \eta \, \nabla F_c\big(\mathbf{w}_{c,k}^t\big) \, \|^2 \\ &= \|\mathbf{w}_{c,k}^t - \mathbf{w}\|^2 \\ &\quad + 2 \, \langle \mathbf{w}_{c,k}^t - \mathbf{w}_t, -\eta \nabla F_c\big(\mathbf{w}_{c,k}^t\big) \rangle \\ &\quad + \eta^2 \|\nabla F_c\big(\mathbf{w}_{c,k}^t\big) \, \|^2. \end{aligned} \tag{36}$$

From the inequality

$$\begin{aligned} 2\langle x, -\eta y \rangle &\leq t \, \|x\|^2 + \frac{\eta^2}{t} \, \|y\|^2, \qquad \forall t > 0 \\ &\overset{t = \frac{1}{K-1}}{=} \frac{1}{K-1} \, \|x\|^2 + (K-1) \, \eta^2 \, \|y\|^2. \end{aligned} \tag{37}$$

By plugging $x = \mathbf{w}_{c,k}^t - \mathbf{w}_t$ and $y = \nabla F_c\Big(\mathbf{w}_{c,k}^t\Big)$ into Eq. (37), we obtain Eq. (36) as follows:

$$\begin{aligned} \|\mathbf{w}_{c,k+1}^t - \mathbf{w}_t\|^2 &\leq \Big(1 + \frac{1}{K-1}\Big) \|\mathbf{w}_{c,k}^t - \mathbf{w}_t\|^2 \\ &\quad + K \, \eta^2 \, \|\nabla F_c\big(\mathbf{w}_{c,k}^t\big) \, \|^2 \\ &\leq \Big(1 + \frac{2}{K}\Big) \|\mathbf{w}_{c,k}^t - \mathbf{w}_t\|^2 \\ &\quad + K \, \eta^2 \, \|\nabla F_c\big(\mathbf{w}_{c,k}^t\big) \, \|^2, \end{aligned} \tag{38}$$

where we used $1 + \frac{1}{K-1} \leq 1 + \frac{2}{K}$ for $K \geq 2$ for the second step.

From the convergence analysis result presented in the single-client descent case [72], local losses are non-increasing along the K steps, i.e., $F_c(\mathbf{w}_{c,k-1}^t) \leq F_c(\mathbf{w}_t)$ and the no-critical-point upper bound, Eq. (38) becomes

$$\begin{aligned} \|\mathbf{w}_{c,k+1}^t - \mathbf{w}_t\|^2 \leq & \Big(1 + \frac{2}{K}\Big) \|\mathbf{w}_{c,k}^t - \mathbf{w}_t\|^2 \\ & + K \, \eta^2 \, \tau_2^2 \, F_c(\mathbf{w}_t) \, . \end{aligned} \tag{39}$$

Unroll the recursion above, and we obtain

$$\begin{aligned} \|\mathbf{w}_{c,k}^t - \mathbf{w}_t\|^2 &\leq \Big(K \, \eta^2 \, \tau_2^2 \, F_c(\mathbf{w}_t) \Big) \sum_{i=1}^{K} \Big(1 + \frac{2}{K}\Big)^{i-1} \\ &\leq 4 \, K^2 \, \eta^2 \, \tau_2^2 \, F_c(\mathbf{w}_t) \, , \end{aligned} \tag{40}$$

where the geometric sequence is given by

$$\sum_{i=1}^{K} \Big(1 + \frac{2}{K}\Big)^{i-1} \le 4K. \tag{41}$$

Finally, the ξ_p is bounded by

$$\xi_p \le 4\,K^2\,\eta^2\,\tau_2^2\,F(\mathbf{w}_t)\,. \tag{42}$$

d) Bound the quadratic term $\|\Delta\mathbf{w}_t\|^2$: Using Eq. (29) and Jensen again, we have

$$\|\Delta\mathbf{w}_t\|^2 = \tilde{\eta}^2 \Big\| \sum_{c,k} q_{c,k}\, g_{c,k-1}^t \Big\|^2 \;\le\; \tilde{\eta}^2 \sum_{c,k} q_{c,k}\, \|g_{c,k-1}^t\|^2. \tag{43}$$

Apply the upper side of the no-critical-point condition for $\|g_{c,k-1}^t\|^2$ component in Eq. (43), we have

$$\|\Delta\mathbf{w}_t\|^2 \le \tilde{\eta}^2 (\tau_2')^2 \sum_{c,k} q_{c,k}\, F_c(\mathbf{w}_{c,k-1}^t), \tag{44}$$

As in [72] again, $F_c(\mathbf{w}_{c,k-1}^t) \le F_c(\mathbf{w}_t)$ for all $k \le K$. Hence,

$$\begin{aligned}\|\Delta\mathbf{w}_t\|^2 &\le \tilde{\eta}^2 (\tau_2')^2 \sum_{c,k} q_{c,k}\, F_c(\mathbf{w}_t) \\ &= \tilde{\eta}^2 (\tau_2')^2\, F(\mathbf{w}_t).\end{aligned} \tag{45}$$

e) Plug back into semi-smoothness and collect terms: By inserting Eq. (35), Eq. (42) and Eq. (45) into Eq. (28), and using the lower side of no-critical-point, i.e., $\|\nabla F(\mathbf{w}_t)\|^2 \ge \tau_1^2 F(\mathbf{w}_t)$, the contraction becomes

$$\begin{aligned}F(\mathbf{w}_{t+1}) - F(\mathbf{w}^*) \le F(\mathbf{w}_t) - F(\mathbf{w}^*) &- \frac{\tilde{\eta}}{2}\,\tau_1^2\,F(\mathbf{w}_t) \\ &+ \frac{\tilde{\eta}}{2}\Big(\big(\beta^2 + \tfrac{\alpha^2}{2}\big)\,\xi_p + \tfrac{\alpha^2}{2}\,F(\mathbf{w}_t)\Big) \\ &+ b\tilde{\eta}^2 {\tau_2'}^2\,F(\mathbf{w}_t) + a\,\tilde{\eta}\tau_2'\,F(\mathbf{w}_t).\end{aligned} \tag{46}$$

Let

$$\begin{aligned}A = \Big(\frac{2\tau_1^2 + \alpha^2}{4} \;-\; a\tau_2\Big)\tilde{\eta} \;-\; b\tau_2^2\tilde{\eta}^2 \\ -\;(2\beta^2 + \alpha^2)\,K^2{\tau_2'}^2\eta^2\tilde{\eta}, \qquad \tilde{\eta} = K\eta.\end{aligned} \tag{47}$$

We can write Eq. (46) as

$$\begin{aligned}F(\mathbf{w}_{t+1}) - F(\mathbf{w}^*) &\le (1-A)F(\mathbf{w}_t) - F(\mathbf{w}^*) \\ &\le (1-A)(F(\mathbf{w}_t) - F(\mathbf{w}^*)).\end{aligned} \tag{48}$$

f) Final convergence statement: For optimal convergence, we require $A > 0$. Furthermore, maximizing the value of A yields a faster convergence rate. We analyze how this is achieved by examining two key parameters.

Impact of the learning rate η (with fixed K): For *small* η (thus small $\tilde{\eta}$), the linear gain dominates, $A \approx \tilde{\eta}\big(\frac{2\tau_1^2+\alpha^2}{4} - a\tau_2\big)$, so increasing η speeds convergence. If η is *too large*, the negative curvature and drift penalties $-b\tau_2^2\tilde{\eta}^2$ and $-(2\beta^2+\alpha^2)K^2{\tau_2'}^2\eta^2\tilde{\eta}$ reduce A and can make it nonpositive.

Impact of the number of local epochs K: With η fixed, the drift term grows like

$$(2\beta^2 + \alpha^2)\,K^2{\tau_2'}^2\eta^2\tilde{\eta} \;=\; (2\beta^2 + \alpha^2)\,{\tau_2'}^2\,K^3\eta^3,$$

so increasing K can sharply decrease A unless η is reduced. If we scale the stepsize as $\eta = c/K$ (so $\tilde{\eta} = c$), then

$$A(c) = c\Big(\tfrac{2\tau_1^2+\alpha^2}{4} - a\tau_2\Big) \;-\; b\tau_2^2c^2 \;-\; (2\beta^2+\alpha^2){\tau_2'}^2c^3, \tag{49}$$

which is independent of K. Thus, with $\eta = c/K$, increasing K mainly reduces communication (fewer rounds per amount of local compute) while leaving the per–round convergence rate determined by the single scalar $c = \tilde{\eta}$ and performing more local computations.

V. Simulation results

To comprehensively evaluate the proposed framework, we conduct experiments on three datasets: MNIST [73], Fashion-MNIST [74], and KMNIST [75]. MNIST serves as the primary benchmark for handwritten digit recognition. FashionMNIST offers a more complex challenge with grayscale images of ten fashion categories, representing higher feature complexity. KMNIST consists of cursive Japanese characters, providing a dataset that is both sparse and structurally diverse. For all three datasets, we employ the standard split of 60,000 training samples and 10,000 test samples. Inputs are normalized to the range [0, 1] without data augmentation. To simulate statistical heterogeneity in the federated setting, data is partitioned across clients using a label-proportion Dirichlet distribution with a concentration parameter α_{dir}. Unless otherwise noted, we use $\alpha_{dir} = 0.5$ for non-iid settings and uniform random partitioning for iid baselines for 25 clients.

We compare the proposed QLIF model against two distinct baselines to validate both its quantum advantage and its competitiveness with classical deep learning. To ensure a fair comparison, the ANN baseline shares the same architecture as our QLIF network. The proposed FL-QLIF network consists of an input layer with 784 neurons (corresponding to the flattened 28×28 input images), a hidden layer with 1024 QLIF neurons, and an output layer with 10 neurons. The spiking threshold is set to 0.3, and we utilize Poisson rate encoding with a simulation window of 25 time steps. The ANN baseline is a classical feedforward network with an identical 784-1024-10 architecture. The QNN baseline represents the quantum state-of-the-art. To maintain comparability, we utilize 4 qubits, which is a common and established configuration for QNNs in the literature [76]–[78]. In this architecture, the input images are flattened, downsampled, and embedded into quantum states using angle encoding, which effectively represents the input layer of the QLIF counterpart. Subsequently, we employ 2 layers of VQC to represent the hidden layer and the final output layer, respectively.

All experiments are conducted using a batch size of 128, and the global model is trained for a total of 60 iterations. Optimization is performed using the Adam optimizer with an initial learning rate of $\eta = 0.005$ and momentum coefficients $\beta_1 = 0.9, \beta_2 = 0.999$.

All simulations were implemented using Python 3.11, PyTorch, and PennyLane. The federated learning environment was simulated on a workstation equipped with an Intel Core i7-14700, 64 GB of RAM, and NVIDIA T400 4GB.

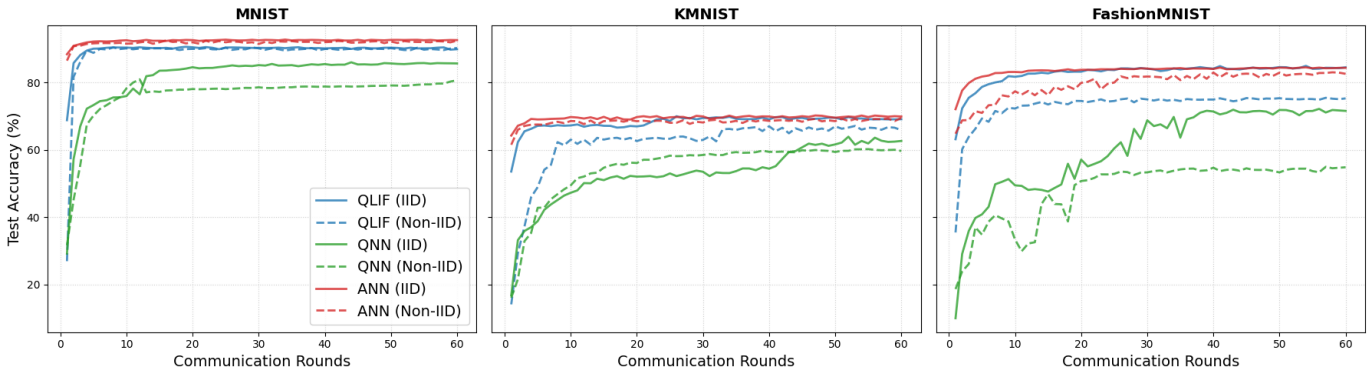

Fig. 3: Performance comparison of the proposed local QLIF model with QNN and ANN models on iid and non-iid data schemes ($K = 1$).

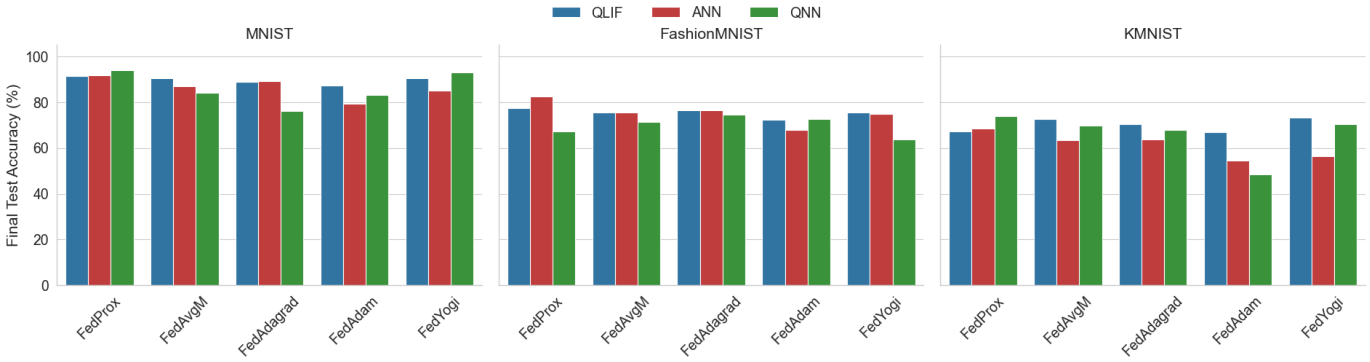

Fig. 4: The performance comparisons of QLIF to ANN/QNN on different FL variants.

A. *Performance comparison*

Fig. 3 illustrates the test accuracy convergence of the proposed FL-QLIF framework compared to the QNN and ANN baselines across MNIST, KMNIST, and FashionMNIST datasets under both iid and non-iid ($\alpha_{dir} = 0.5$) settings. The proposed QLIF model demonstrates significantly superior robustness and convergence speed compared to the standard QNN baseline. Across all three datasets, the QNN (green lines) exhibits slower learning curves and lower final accuracy. This performance gap is particularly pronounced in non-iid environments (dashed green lines), where the QNN suffers from severe instability and degradation. For instance, on the complex FashionMNIST dataset, the non-iid QNN struggles to surpass 60% accuracy, whereas the non-iid QLIF maintains a significantly higher accuracy of 75%. This suggests that the variational quantum circuits in standard QNNs struggle to generalize when client data is statistically heterogeneous. In contrast, FL-QLIF maintains high stability, effectively mitigating the client drift that hampers the baseline QNN.

Remarkably, the FL-QLIF model achieves convergence speeds and final accuracy levels that are highly comparable to the classical ANN baseline. In the MNIST and KMNIST experiments, the performance gap between QLIF and ANN is negligible. On MNIST, the QLIF model converges rapidly to around 91%, closely trailing the ANN's around 92.5% peak. This demonstrates that the quantum-inspired neuron model successfully captures the expressivity required for these classification tasks while offering the energy efficiency benefits detailed in Section V-C. The impact of statistical heterogeneity varies by dataset complexity. For MNIST and KMNIST, both QLIF and ANN exhibit strong resilience, maintaining nearly identical performance in iid and non-iid settings. However, the challenge of heterogeneity becomes evident in the FashionMNIST dataset. Here, both the ANN and QLIF models experience a noticeable drop in accuracy under non-iid conditions (dashed lines) compared to their iid counterparts (solid lines). Specifically, the QLIF accuracy drops from 84% (iid) to 75% (non-iid), mirroring the ANN's decline from 84% to 81%. Despite this "suffering" caused by the complex distribution shift, QLIF continues to track the ANN's performance closely, proving its viability for statistically heterogeneous federated environments where standard quantum models fail.

B. *Robustness to federated optimization algorithms*

While the proposed QLIF neuron demonstrates strong performance under standard aggregation (FedAvg), practical federated environments often present significant optimization challenges. Specifically, the non-differentiable nature of spiking dynamics combined with high statistical heterogeneity can lead to unstable gradient estimates. To evaluate the robustness of our framework, we conduct experiments on three datasets (MNIST, FashionMNIST, KMNIST) under a challenging non-iid scenario ($\alpha_{dir} = 0.5$). We compare the performance across five distinct federated optimization strategies: FedProx,

FedAvgM, FedAdagrad, FedAdam, and FedYogi. The experiments are conducted with $\mathcal{C} = 25$ clients and $K = 3$ local epochs. The specific hyperparameters for the advanced variants are set as follows: for FedProx, the proximal coefficient is $\mu = 0.1$; for FedAvgM, server momentum is $\beta = 0.9$. For the adaptive methods (FedAdagrad, FedAdam), we utilize a server learning rate of $\eta_g = 0.01$ with stability constant $\epsilon = 10^{-3}$. For FedYogi specifically, we adopt decay rates of $\beta_1 = 0.9$ and $\beta_2 = 0.99$ to control the effective learning rate growth.

Fig. 4 presents the comparative results of the three models across these optimization strategies. Several key observations can be drawn from the analysis. In general, the FL-QLIF model demonstrates remarkable stability and robustness across the different optimizers. It consistently outperforms the baselines on FedAvgM, FedAdam, and FedYogi, particularly in the more complex KMNIST and FashionMNIST tasks. For instance, in KMNIST under FedYogi, FL-QLIF achieves 73.25%, significantly surpassing the ANN (56.51%) and the QNN (70.38%). Similarly, on FashionMNIST with FedAvgM, FL-QLIF reaches 75.64%, edging out both the ANN (75.58%) and the QNN (71.24%). Even in cases where it is not the strict top performer, such as with FedProx on MNIST (91.35%), FL-QLIF maintains highly competitive accuracy relative to the best-performing ANN (91.71%) and QNN (93.95%) variants, validating its resilience to diverse optimization landscapes.

Furthermore, the QNN baseline benefits significantly from these advanced optimization strategies. Compared to its performance under standard FedAvg (as discussed in Section V-A), the QNN shows noticeable improvements in most scenarios here. For example, while QNN performance often lags in simpler settings, it achieves a surprising 93.95% on MNIST with FedProx and 93.10% with FedYogi, surpassing the ANN in both cases. This suggests that advanced aggregators like FedProx and FedYogi help mitigate the gradient variance issues that typically hamper variational quantum circuits in non-iid settings.

Finally, the FedAdam algorithm reveals a distinct divergence in model adaptability. The classical ANN struggles significantly, evidenced by a sharp drop in accuracy across all datasets. Specifically, it falls to 54.65% on KMNIST and 79.31% on MNIST. Conversely, the QNN maintains relatively good performance and reaches 83.09% on MNIST. Most notably, FL-QLIF outperforms all other models in this challenging regime. It achieves 66.87% on KMNIST compared to 54.65% for the ANN and 48.45% for the QNN. Although most models generally perform worse with FedAdam compared to other variants, QLIF remains the most resilient choice. This highlights its superior adaptability to adaptive learning rates in scenarios where classical networks may fail.

C. Energy efficiency evaluation

A primary motivation for adopting neuromorphic-based computing in federated learning is the potential for significant energy savings. To quantify this advantage, we estimate the theoretical energy consumption of the proposed QLIF model compared to the classical ANN baseline. According to [79], the computational cost of ANNs is dominated by multiply-accumulate (MAC) operations, which occur densely across all connections. In contrast, SNNs operate in an event-driven manner where computation—specifically accumulate (AC) operations—is triggered only when a spike occurs. For a fully connected layer l with N_{in}^l inputs and N_{out}^l outputs, the number of operations per inference is defined as follows:

- The ANN performs a MAC operation for every weight regardless of input sparsity

$$N_{MAC}^l = N_{in}^l \times N_{out}^l. \tag{50}$$

- The spiking-based model distributes computation over T time steps. An operation occurs only if a presynaptic neuron fires. Let R^l denote the average spike rate (sparsity) of the input to layer l

$$N_{AC}^l = N_{in}^l \times N_{out}^l \times T \times R^l. \tag{51}$$

TABLE III: Measured average spike activity (R) per layer

Dataset	**Input**	**Hidden**	**Output**
MNIST	0.13	0.04	0.07
FashionMNIST	0.29	0.09	0.05
KMNIST	0.18	0.06	0.05

To determine the sparsity factor R, we performed inference on the global test sets of all three datasets (MNIST, FashionMNIST, KMNIST) using the trained global model. We measured the R for each layer, defined as the ratio of total spikes emitted to the total possible spikes over the simulation window $T = 25$. The empirically measured spike rates are summarized in Table III. It is important to note that while we report the output layer's spike rate to analyze network behavior, only the sparsity of the input and hidden layers contributes to the final energy calculations, as these drive the synaptic operations in the network.

We adopt energy cost values for 45nm CMOS technology with 32-bit integer arithmetic [80]: $E_{MAC} = 3.2$ pJ and $E_{AC} = 0.1$ pJ. As a result, the ANN baseline consumes a constant energy regardless of the dataset, as its topology and operation count remain fixed ($N_{total} \approx 8.13 \times 10^5$ MACs)

$$E_{ANN} = \sum_l (N_{MAC}^l) \times E_{MAC} \approx 2.6\mu J. \tag{52}$$

In contrast, the energy of the FL-QLIF model varies across datasets due to differences in feature sparsity

$$E_{QLIF} = \sum_l (N_{AC}^l) \times E_{AC}. \tag{53}$$

For MNIST, the QLIF model consumes approximately $0.26\mu J$, representing a **10× reduction** in energy compared to the ANN. Similarly, for KMNIST, the model achieves a **7.2**× savings with an energy consumption of roughly **0.36**μJ. Even on the more complex FashionMNIST dataset, where spike activity is naturally higher, the QLIF model maintains a low consumption of **0.58**μJ, delivering a **4.4**× efficiency gain. Besides, it is worth noting that while the event-driven sparsity of the proposed framework significantly reduces the synaptic energy footprint compared to standard ANNs, it is important to discuss the computational complexity introduced by the QLIF

TABLE IV: Scalability analysis: Final test accuracy (%) under varying participation ratios (r) and number of clients (C)

Dataset	Model	$\mathbf{C = 50}$				$\mathbf{C = 100}$			
		$\mathbf{r = 0.1}$	$\mathbf{r = 0.25}$	$\mathbf{r = 0.5}$	$\mathbf{r = 1.0}$	$\mathbf{r = 0.1}$	$\mathbf{r = 0.25}$	$\mathbf{r = 0.5}$	$\mathbf{r = 1.0}$
MNIST	**QLIF**	**87.86**	**87.69**	**89.04**	**90.12**	**85.40**	**89.32**	**89.36**	**90.66**
	ANN	86.71	87.01	89.56	92.56	87.58	88.92	89.43	91.66
	QNN	39.94	64.98	66.03	71.26	56.82	65.10	67.92	72.12
KMNIST	**QLIF**	**60.24**	**62.42**	**63.52**	**65.24**	**59.66**	**63.44**	**63.73**	**64.82**
	ANN	60.42	61.76	61.69	67.24	60.13	64.98	64.67	67.49
	QNN	45.50	50.38	52.93	57.89	42.05	54.79	58.53	60.15
FashionMNIST	**QLIF**	**68.60**	**72.25**	**77.36**	**79.23**	**69.80**	**72.85**	**75.61**	**78.68**
	ANN	70.63	71.69	75.08	80.34	70.83	72.95	74.95	79.54
	QNN	41.45	48.09	50.63	53.67	40.25	46.63	50.19	54.25

neuron dynamics themselves. Simulating the LIF neuron using quantum rotation gates requires evaluating non-linear transcendental functions, such as $\arcsin$, $\sin$, and exponential decays. While these operations introduce a complicated computational overhead at the individual neuron level, this cost is effectively marginalized by the network's structural scaling. Algorithmic complexity dictates that neuron state updates scale linearly with the number of neurons ($\mathcal{O}(N)$), whereas synaptic operations scale quadratically with the number of connections ($\mathcal{O}(N^2)$). In our evaluated 784-1024-10 architecture, synaptic connections outnumber neurons, i.e., approximately 8.13×10^5 MACs versus 1034 active QLIF neurons. Consequently, the total computational cost remains overwhelmingly dominated by the massive reduction in dense synaptic MAC operations. These results highlight that the proposed quantum-inspired framework successfully leverages data sparsity to minimize computational cost on edge hardware.

D. Scalability

To validate the practical viability of FL-QLIF in large-scale edge networks, we evaluate its scalability and resilience to partial client participation. In real-world IoT scenarios, edge devices frequently drop out due to connectivity issues or battery constraints, making full participation (100% of clients) unrealistic. We investigate the model's performance by varying the participation ratio $r \in \{0.1, 0.25, 0.5\}$ across a total pool of $\mathcal{C} = \{50, 100\}$ clients on the MNIST, FashionMNIST, and KMNIST datasets. To ensure a rigorous evaluation under realistic non-iid conditions, we set the $\alpha_{dir} = 0.5$ and fix the number of local update epochs to $K = 3$ for all scalability experiments. Note that these distributions are assumed to be stationary throughout training. The performance on full participation ($r = 1$) is also evaluated for benchmarking purposes.

The experimental results summarized in Table IV demonstrate the practical scalability and resilience of the proposed FL-QLIF framework in large-scale edge networks. A critical observation is that the FL-QLIF model maintains a highly competitive performance relative to the classical ANN baseline across all datasets and participation scenarios. For instance, on the MNIST dataset with $C = 50$ and a participation ratio of $r = 0.5$, the FL-QLIF model achieves an accuracy of 89.04%, closely trailing the ANN's accuracy of 89.56%. Similarly, in the more complex FashionMNIST task with $\mathcal{C} = 100$ clients, the quantum-inspired model reaches 72.85% accuracy at $r = 0.25$, effectively matching the ANN benchmark of roughly 72.95%. This confirms that the QLIF neuron successfully captures the necessary feature representations required for complex classification tasks without the extensive computational overhead associated with full classical networks.

Furthermore, the proposed framework exhibits remarkable robustness to partial client participation. While the ANN baseline tends to suffer more pronounced performance degradation as the participation ratio drops, the FL-QLIF model preserves its stability. In the KMNIST experiments ($C = 100$), as the participation ratio decreases to $r = 0.1$, the FL-QLIF model maintains a consistent accuracy profile, achieving 59.66% compared to the ANN's 60.13%. Notably, the performance gap between the low-participation regime ($r = 0.1$) and full participation ($r = 1.0$) is often narrower for QLIF than for the baselines, suggesting that the quantum-inspired aggregation mechanism effectively mitigates the noise introduced by random client sampling. In stark contrast, the standard QNN baseline struggles significantly in these heterogeneous environments, failing to surpass 54% accuracy on FashionMNIST ($C = 50$) even at higher participation rates. These results collectively highlight that FL-QLIF offers a scalable and resilient solution for edge intelligence, delivering high accuracy even when network resources limit client availability.

E. Impact of learning rate and local epochs on convergence

We now compare our theoretical analysis with empirical results. For simplicity and clarity of analysis, we conduct these specific experiments using the MNIST dataset. Fig. 6 (a) illustrates the impact of the number of local epochs K with fixed $\eta = 5 \times 10^{-3}$. In this case, increasing K aggravates the drift penalty, which grows as $(2\beta^2+\alpha^2)\tau_2'^2 K^3\eta^3$, thereby reducing A and slowing down per-round progress. Empirically, $K = 1$ and $K = 3$ achieve rapid convergence and stable final accuracy, whereas $K = 5$ and especially $K = 10$

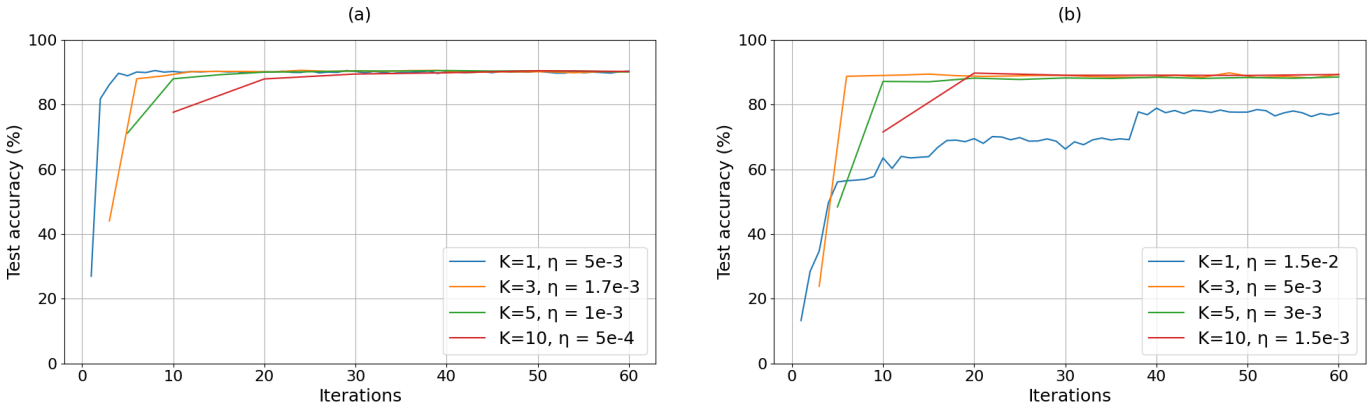

Fig. 5: Convergence under the scaling rule $\eta = c/K$: (a) Experiment 1 ($c = 5 \times 10^{-3}$); (b) Experiment 2 ($c = 1.5 \times 10^{-2}$)

TABLE V: Quantitative results of scaling the learning rate with local epochs

K	η	**Accuracy (%)**	**Conv. Round**	**Behavior**
Experiment 1: $c = 5{\times}10^{-3}$				
1	$5{\times}10^{-3}$	90.3	6-8	Stable, rapid convergence
3	$1.7{\times}10^{-3}$	90.1	6-9	Stable, rapid convergence
5	$1{\times}10^{-3}$	90.0	10-15	Smooth, consistent
10	$5{\times}10^{-4}$	90.2	10-20	Stable, slower start
Experiment 2: $c = 1.5{\times}10^{-2}$				
1	$1.5{\times}10^{-2}$	77.4	–	Unstable, oscillatory updates
3	$5{\times}10^{-3}$	89.3	9-12	Stable, fast convergence
5	$3{\times}10^{-3}$	88.5	10-15	Slower but stable
10	$1.5{\times}10^{-3}$	89.3	10-20	Consistent overall

converge more slowly and plateau at slightly lower accuracy. This confirms the theoretical prediction that large K values, if combined with a fixed η, adversely affect convergence due to the accumulation of drift.

Fig. 6 (b) shows the effect of varying the learning rate η for fixed $K = 3$. When η is small, the linear gain term dominates ($A \approx \tilde{\eta} \cdot \text{const}$), so increasing η accelerates convergence. This behavior is evident for $\eta = 5 \times 10^{-4}$, which converges slowly but stably, while moderate values such as $\eta = 10^{-3}$ and $\eta = 5 \times 10^{-3}$ achieve the fastest improvement and reach around 90% accuracy within 10 rounds. As predicted, when η becomes too large, the quadratic and cubic penalty terms ($-b\tau_2^2\tilde{\eta}^2$ and $-(2\beta^2 + \alpha^2)K^2\tau_2'^2\eta^2\tilde{\eta}$) reduce the effective gain, causing the training to slow down and fluctuate. This effect appears clearly for $\eta = 10^{-2}$ and $\eta = 5 \times 10^{-2}$, where convergence becomes unstable and the final accuracy is lower.

To further verify the theoretical prediction that the per-round convergence rate remains constant when scaling the stepsize as $\eta = c/K$, we conducted an experiment where the learning rate and the number of local epochs K were jointly varied while maintaining approximately constant $c = K\eta$. Starting from the well-performing configurations (K=1, η=5×10^{-3}) and (K=3, η=5×10^{-3}), we scaled η inversely with respect to K for larger local epoch counts. The corresponding results are shown in Fig. 5(a) and Fig. 5(b). Furthermore, the detailed quantitative outcomes are summarized in Table II, which lists the accuracy, convergence rounds, and qualitative behavior for each configuration.

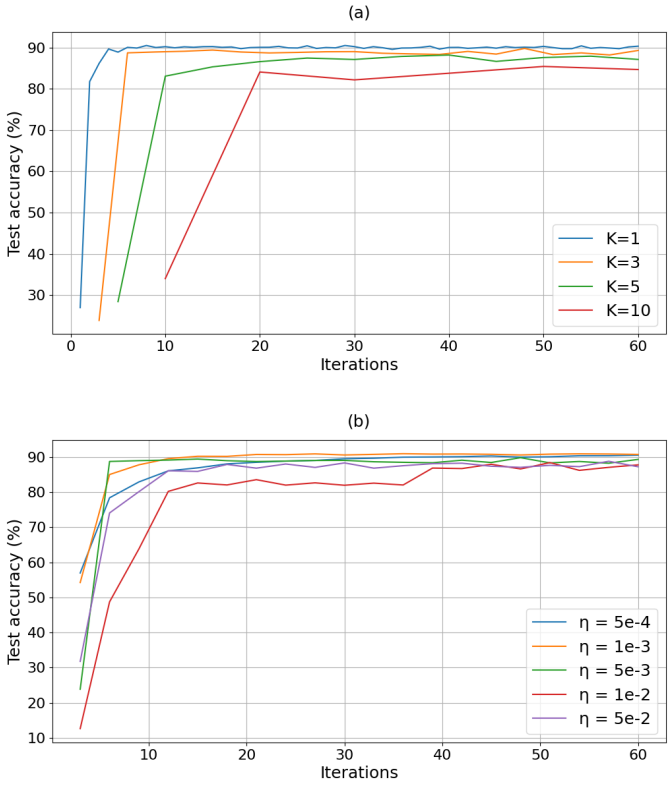

Fig. 6: Performance comparison of (a) different local updates between each global communication round, (b) different learning rate.

In the first experiment, the four configurations generally follow the expected trend. The curves for $K = 1, 3,$ and 5 overlap after roughly 10 communication rounds, demonstrating nearly identical convergence speed and final accuracy ($\approx 90\%$). The $K = 10$ case, however, converges slightly more slowly, indicating that excessively large K begins to weaken the assumption of constant per-round convergence. This observation is consistent with the theory on the effect of local epochs K. This experiment is quite consistent with the theoretical result. In this regime, increasing K reduces the communication frequency without degrading learning performance.

In the other experiment, the overall trend still supports the theory, but with minor deviations. The configuration (K=1, η=1.5×10^{-2}) shows unstable updates and lower accuracy (approximate 75%), indicating that the cubic penalty terms in $A(c)$ dominate when η becomes too large. Conversely, the other three settings (K=3, 5, 10) converge almost identically and stably.

Overall, the experiments validate the two key predictions: (i) the learning rate exhibits a trade-off between faster linear gain at small η and instability from curvature and drift penalties at large η, and (ii) increasing the number of local epochs sharply reduces the per-round convergence rate when η is fixed, as predicted by the cubic growth of the drift term. Furthermore, the results empirically verify the third prediction: scaling the learning rate inversely with K maintains a nearly constant convergence rate across different local update counts. However, both K and η must remain within moderate ranges; excessively large values of either can violate the above conditions, leading to slower or unstable convergence.

VI. Conclusion

In this paper, we presented FL-QLIF, a unified framework that synergizes the privacy-preserving capabilities of FL with the energy efficiency of SNNs and the representational power of QC. We introduced an improved QLIF neuron model that leverages unitary rotation gates to capture biologically plausible integration and leakage dynamics within a single computational step. Empirically, we demonstrated that FL-QLIF is robust to statistical heterogeneity and highly scalable, with experimental results verifying the stability bounds predicted by our theoretical analysis. In direct comparisons, our model consistently outperformed QNN baselines in non-iid scenarios and maintained accuracy competitive with full-precision ANNs. Crucially, this performance is achieved with a fraction of the energy cost. Specifically, it demonstrates up to a $10\times$ reduction in synaptic energy consumption for sparse data tasks. These results, grounded in theoretical proof of convergence under semi-smooth conditions, suggest that quantum-inspired neuromorphic computing is a powerful enabler for next-generation edge AI, offering a path toward ubiquitous, low-power, and privacy-secure distributed learning.

Future work will focus on deploying this framework on physical neuromorphic hardware to empirically validate the predicted energy benefits, characterize the overhead of fixed-point approximations of the QLIF neuron dynamics, and explore the practical constraints of hardware-level deployment in large-scale asynchronous IoT networks. In this context, a key priority will be developing adjustable thresholds that adapt to device power settings, helping to find the best balance between saving energy and maintaining high accuracy. Furthermore, while our initial evaluation establishes the framework's viability on standard benchmarks, the QLIF neuron model can be extended into more advanced architectures to efficiently process more complex, high-dimensional datasets. Finally, extending FL-QLIF to non-stationary federated environments with concept drift remains an important future direction.

References

[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Jun. 2019.

[2] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Gener. Comput. Syst.*, vol. 74, pp. 76–85, Sep. 2017.

[3] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, Apr. 2020.

[4] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, Jan. 2020.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, FL, USA, Apr. 2017, pp. 1273–1282.

[6] B. Hazarika, K. Singh, A. Paul, and T. Q. Duong, "Hybrid machine learning approach for resource allocation of digital twin in uav-aided internet-of-vehicles networks," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 2923–2939, Nov. 2023.

[7] C. Che, X. Li, C. Chen, X. He, and Z. Zheng, "A decentralized federated learning framework via committee mechanism with convergence guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4783–4800, Aug. 2022.

[8] E. T. Martínez Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2983–3013, Sep. 2023.

[9] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2021.

[10] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2983–3013, Sep. 2023.

[11] Y. Sun, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Semi-decentralized federated edge learning for fast convergence on non-iid data," in *Proc. IEEE Wirel. Commun. Netw. Conf. (WCNC)*, TX, USA, Apr. 2022, pp. 1898–1903.

[12] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, Apr. 2020.

[13] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv:1909.06335*, 2019.

[14] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv:2003.00295*, 2020.

[15] H. Yang, K.-Y. Lam, L. Xiao, Z. Xiong, H. Hu, D. Niyato, and H. Vincent Poor, "Lead federated neuromorphic learning for wireless edge artificial intelligence," *Nat. Commun.*, vol. 13, no. 1, p. 4269, Jul. 2022.

[16] L. B. Kristensen, M. Degroote, P. Wittek, A. Aspuru-Guzik, and N. T. Zinner, "An artificial spiking quantum neuron," *npj Quantum Inf.*, vol. 7, no. 1, p. 59, Apr. 2021.

[17] D. Marković and J. Grollier, "Quantum neuromorphic computing," *Appl. Phys. Lett.*, vol. 117, no. 15, Oct. 2020.

[18] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.

[19] E. Hunsberger and C. Eliasmith, "Training spiking deep networks for neuromorphic hardware," *arXiv:1611.05141*, 2016.

[20] A. W. Harrow and A. Montanaro, "Quantum computational supremacy," *Nature*, vol. 549, no. 7671, pp. 203–209, Sep. 2017.

[21] S. C. Prabhashana, D. Van Huynh, H. Jung, B. Canberk, S. L. Cotton, and T. Q. Duong, "Quantum deep reinforcement learning for urllc satellite-air-ground integrated networks with digital twin applications," *IEEE Internet Things J.*, Nov. 2025.

[22] D. Brand and F. Petruccione, "A quantum leaky integrate-and-fire spiking neuron and network," *npj Quantum Infor.*, vol. 10, no. 1, pp. 1–8, Dec. 2024.

[23] Z. Cai, J. Pang, Y. Li, Y. Huang, and Z. Xie, "A comprehensive survey of federated open-world learning," *IEEE Trans. Netw. Sci. Eng.*, Jun. 2025.

[24] Y. Mu, N. Garg, and T. Ratnarajah, "Federated learning in massive mimo 6g networks: Convergence analysis and communication-efficient design," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 4220–4234, Aug. 2022.

[25] J. Yang, W. Jiang, and L. Nie, "Hypernetworks-based hierarchical federated learning on hybrid non-iid datasets for digital twin in industrial iot," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 2, pp. 1413–1423, Oct. 2023.

[26] X. Zhou, W. Liang, J. Ma, Z. Yan, and K. I.-K. Wang, "2d federated learning for personalized human activity recognition in cyber-physical-social systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 3934–3944, Jan. 2022.

[27] H. Ma, K. Yang, and Y. Jiao, "Cellular traffic prediction via byzantine-robust asynchronous federated learning," *IEEE Trans. Netw. Sci. Eng.*, Feb. 2025.

[28] S. Pala, K. Singh, C.-P. Li, O. A. Dobre, and T. Q. Duong, "Joint beamforming design and sensing in satellite and ris-enhanced terrestrial networks: A federated learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 11, no. 5, pp. 3397–3411, Jan. 2025.

[29] M. R. Uddin, S. Shaon, R. Rahman, D. C. Nguyen, O. A. Dobre, and D. Niyato, "Quantum federated learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 28, pp. 3942–3975, Dec. 2026.

[30] C. Ren, R. Yan, H. Zhu, H. Yu, M. Xu, Y. Shen, Y. Xu, M. Xiao, Z. Y. Dong, M. Skoglund *et al.*, "Toward quantum federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, Sep. 2025.

[31] B. Hazarika, K. Singh, O. A. Dobre, C.-P. Li, and T. Q. Duong, "Quantum-enhanced federated learning for metaverse-empowered vehicular networks," *IEEE Trans. Commun.*, vol. 73, no. 6, pp. 4168–4183, Nov. 2024.

[32] M. Xu, D. Niyato, Z. Yang, Z. Xiong, J. Kang, D. I. Kim, and X. Shen, "Privacy-preserving intelligent resource allocation for federated edge learning in quantum internet," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 142–157, Nov. 2022.

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

[34] N. Innan, M. A.-Z. Khan, A. Marchisio, M. Shafique, and M. Bennai, "Fedqnn: Federated learning using quantum neural networks," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*, Yokohama, Japan, Sep. 2024, pp. 1–9.

[35] Q. Xia and Q. Li, "Quantumfed: A federated learning framework for collaborative quantum training," in *Proc. IEEE Glob. commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[36] R. Rahman, S. Shaham, and D. C. Nguyen, "Toward personalized quantum federated learning for anomaly detection," *IEEE Trans. Netw. Sci. Eng.*, vol. 13, pp. 3335–3350, 2025.

[37] D. Gurung and S. R. Pokhrel, "Performance analysis and design of a weighted personalized quantum federated learning," *IEEE Trans. Artif. Intell.*, Aug. 2025.

[38] R. Huang, X. Tan, and Q. Xu, "Quantum federated learning with decentralized data," *IEEE J. Sel. Topics Quantum Electron.*, vol. 28, no. 4: Mach. Learn. in Photon. Commun. and Meas. Syst., pp. 1–10, Apr. 2022.

[39] Z. Qu, Y. Li, B. Liu, D. Gupta, and P. Tiwari, "Dtqfl: A digital twin-assisted quantum federated learning algorithm for intelligent diagnosis in 5g mobile network," *IEEE J. Biomed. Health Inform.*, Aug. 2023.

[40] N. Innan, A. Marchisio, M. Bennai, and M. Shafique, "Qfnn-ffd: Quantum federated neural network for financial fraud detection," in *Proc. IEEE Int. Conf. Quantum Softw. (QSW)*, Helsinki, Finland, Jul. 2025, pp. 41–47.

[41] C. Ren, Z. Y. Dong, M. Skoglund, Y. Gao, T. Wang, and R. Zhang, "Enhancing dynamic security assessment in smart grids through quantum federated learning," *IEEE Trans. Autom. Sci. Eng.*, Nov. 2024.

[42] Z. Qu, Z. Chen, S. Dehdashti, and P. Tiwari, "Qfsm: a novel quantum federated learning algorithm for speech emotion recognition with minimal gated unit in 5g iov," *IEEE Trans. Intell. Veh.*, Oct. 2024.

[43] N. Skatchkovsky, H. Jang, and O. Simeone, "Spiking neural networks—part iii: Neuromorphic communications," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1746–1750, Jun. 2021.

[44] N. Skatchkovsky *et al.*, "Federated neuromorphic learning of spiking neural networks for low-power edge intelligence," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Virtual Barcelona., May 2020, pp. 8524–8528.

[45] Y. Venkatesha, Y. Kim, L. Tassiulas, and P. Panda, "Federated learning with spiking neural networks," *IEEE Trans. Signal Process.*, vol. 69, pp. 6183–6194, Oct. 2021.

[46] J. Ji, C.-T. Lam, K. Wang, and B. K. Ng, "Amned: An efficient framework for spiking neuron coding in aircomp federated learning," *IEEE Access*, Aug. 2025.

[47] K. Xie, Z. Zhang, B. Li, J. Kang, D. Niyato, S. Xie, and Y. Wu, "Efficient federated learning with spike neural networks for traffic sign recognition," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9980–9992, May 2022.

[48] M. M. H. Galib and M. Younis, "Lightweight spiking federated learning-based detector for cognitive vehicular networking," in *Proc. IEEE Int. Conf. Commun.*, Montreal, Canada, Jun. 2025, pp. 3569–3574.

[49] C. Shang, D. T. Hoang, M. Hao, D. Niyato, and J. Yu, "Energy-efficient decentralized federated learning for uav swarm with spiking neural networks and leader election mechanism," *IEEE Commun. Lett.*, Aug. 2024.

[50] M. Zhang, B. Li, H. Liu, and C. Zhao, "Federated learning for radar gesture recognition based on spike timing-dependent plasticity," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 2, pp. 2379–2393, Jan. 2024.

[51] A. Ajayan and A. P. James, "Edge to quantum: hybrid quantum-spiking neural network image classifier," *Neuromorphic Comput. Eng.*, vol. 1, no. 2, p. 024001, Sep. 2021.

[52] Y. Sun, Y. Zeng, and T. Zhang, "Quantum superposition inspired spiking neural network," *Iscience*, vol. 24, no. 8, Aug. 2021.

[53] Z. Xu, K. Shen, P. Cai, T. Yang, Y. Hu, S. Chen, Y. Zhu, Z. Wu, Y. Dai, J. Wang *et al.*, "Parallel proportional fusion of a spiking quantum neural network for optimizing image classification," *Applied Intell.*, vol. 54, no. 22, pp. 11 876–11 891, Sep. 2024.

[54] D. Konar, V. Aggarwal, A. D. Sarma, S. Bhandary, S. Bhattacharyya, and A. Cangi, "Deep spiking quantum neural network for noisy image classification," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*, Queensland, Australia, Jun. 2023, pp. 1–10.

[55] Y. Chen, C. Wang, H. Guo, X. Gao, and J. Wu, "Accelerating spiking neural networks using quantum algorithm with high success probability and high calculation accuracy," *Neurocomputing*, vol. 493, pp. 435–444, Jul. 2022.

[56] L. Lyu, C. Pang, and J. Wang, "A quantum model of biological neurons," *Neurocomputing*, vol. 598, p. 128223, Sep. 2024.

[57] D. Konar, A. D. Sarma, S. Bhandary, S. Bhattacharyya, A. Cangi, and V. Aggarwal, "A shallow hybrid classical–quantum spiking feedforward neural network for noise-robust image classification," *Appl. Soft Comput.*, vol. 136, p. 110099, Mar. 2023.

[58] M. K. Saggi, A. S. Bhatia, and S. Kais, "Quantum integration of spiking leaky integrate-and-fire neurons and variational circuits for enhanced multiclass classification," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, vol. 1, New Mexico, USA, Aug. 2025, pp. 2419–2425.

[59] D. Brand and R. K. Jha, "Quantum neuromorphic classification of eeg brain signals," in *Proc. IEEE Int. Conf. Quantum Artif. Intell. (QAI)*, Naples, Italy, Nov. 2025, pp. 92–97.

[60] N. Innan, A. Marchisio, and M. Shafique, "Fl-qdsnns: Federated learning with quantum dynamic spiking neural networks," in *Proc. IEEE Int. Conf. Quantum Artif. Intell. (QAI)*, Naples, Italy, Nov. 2025, pp. 113–119.

[61] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay, "Opportunities for neuromorphic computing algorithms and applications," *Nat. Comput. Sci.*, vol. 2, no. 1, pp. 10–19, Jan. 2022.

[62] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proc. IEEE*, vol. 111, no. 9, pp. 1016–1054, Sep. 2023.

[63] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, p. 500, Aug. 1952.

[64] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Rapid feedforward computation by temporal encoding and learning with spiking neurons," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1539–1552, Feb. 2013.

[65] X. Wang, X. Lin, and X. Dang, "Supervised learning in spiking neural networks: A review of algorithms and evaluations," *IEEE Trans. Neural Netw.*, vol. 125, pp. 258–280, May 2020.

[66] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1-4, pp. 17–37, Oct. 2002.

[67] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," *Adv. Neural Inf. Process. Syst.*, vol. 31, Dec. 2018.

[68] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 51–63, Nov. 2019.

[69] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian, "Deep residual learning in spiking neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 21 056–21 069, Dec. 2021.

[70] D. Brand, I. Sinayskiy, and F. Petruccione, "Markovian noise modelling and parameter extraction framework for quantum devices," *Sci. Rep.*, vol. 14, no. 1, p. 4769, Feb. 2024.

[71] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. Int. Conf. Mach. Learn.*, California, USA, Jun. 2019, pp. 242–252.

[72] X. Li, Z. Song, R. Tao, and G. Zhang, "A convergence theory for federated average: Beyond smoothness," in *Proc. IEEE Int. Conf. Big Data*, Osaka, Japan, Dec. 2022, pp. 1292–1297.

[73] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Oct. 2012.

[74] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv:1708.07747*, 2017.

[75] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, "Deep learning for classical japanese literature," *arXiv:1812.01718*, 2018.

[76] N. Q. Thoong, A. A. Cheema, B. Canberk, D. T. Tran, O. A. Dobre, and T. Q. Duong, "Channel estimation for reconfigurable intelligent surface-aided 6g noma systems: a quantum machine learning approach," *IEEE Trans. Netw. Sci. Eng.*, Sep. 2025.

[77] V.-L. Nguyen, L.-H. Nguyen, R.-H. Hwang, B. Canberk, and T. Q. Duong, "Quantum machine learning for 6g network intelligence and adversarial threats," *IEEE Commun. Stand. Mag.*, May 2025.

[78] B.-N. D. Tran, M. Fahim, B. D. McNiven, M. Guizani, H. Shin, and T. Q. Duong, "Quantum lstm model for estimation of energy expenditure in human aging using wearable iot healthcare technology," *IEEE Internet Things J.*, Apr. 2025.

[79] Z. Pan, Y. Chua, J. Wu, M. Zhang, H. Li, and E. Ambikairajah, "An efficient and perceptually motivated auditory neural encoding and decoding algorithm for spiking neural networks," *Front. Neurosci.*, vol. 13, p. 1420, Jan. 2020.

[80] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *Proc. IEEE Int. Solid-state Circuits Conf. Dig. Tech. Papers (ISSCC)*, CA, USA, Feb. 2014, pp. 10–14.