Nested Quasi-Newton Optimization for Federated Learning Under Periodic Deterministic Communication Constraints

Lei Zhao, Wu-Sheng Lu, *Life Fellow, IEEE,* and Lin Cai, *Fellow, IEEE*

***Abstract*—Federated Learning (FL) enables decentralized model training while preserving data privacy, however, real-world deployments are often constrained by Periodic Deterministic Communication (PDC) schedules, where communication between clients and the central server occurs at fixed intervals due to bandwidth limitations, energy constraints, or regulatory restrictions. These rigid schedules introduce fundamental challenges, including delayed model updates, model drift, inefficient convergence, and heightened sensitivity to non-IID data distributions, which undermine FL performance in practical settings. To address these limitations, we propose Federated Nested Quasi-Newton Optimization (FedNQN), a novel framework that accelerates convergence and enhances FL robustness under PDC constraints. FedNQN integrates curvature-aware central acceleration with variance-controlled local adaptation, ensuring stable learning dynamics despite restricted communication. At the global level, second-order curvature information accelerates model updates, compensating for infrequent synchronization, while local updates leverage variance-controlled optimizations to mitigate drift and adapt to heterogeneous data distributions. This coordinated optimization strategy enhances convergence speed, improves model accuracy, and maintains computational efficiency, making FL more adaptable to real-world constraints. Extensive experiments on benchmark datasets validate FedNQN's effectiveness, demonstrating superior performance over state-of-the-art FL methods in terms of stability, scalability, and resilience to communication inefficiencies.**

***Index Terms*—Periodic Deterministic Communication, Nested Quasi-Newton Optimization, Curvature-Aware Global Acceleration, Variance-Controlled Local Adaptation**

I. Introduction

Federated learning (FL) has emerged as a transformative approach to collaborative machine learning, enabling decentralized clients to train shared models while preserving data privacy by avoiding direct data exchange [1]–[3]. This privacy-preserving paradigm is particularly valuable in data-sensitive domains such as mobile services, Internet-of-Things (IoT) deployments, smart grids, and healthcare monitoring [4], [5], where centralizing raw data is undesirable due to privacy, security, and reliability concerns. However, real-world FL deployments over edge devices face substantial systems-level constraints. In many practical settings, communication between clients and the central server is restricted to periodic or duty-cycled windows, driven by bandwidth limitations, energy-saving policies, protocol-imposed duty cycles, and scheduled access windows in wireless and IoT networks [6]–[9]. These periodic deterministic communication (PDC) constraints delay model updates, exacerbate model drift, and increase sensitivity to non-IID data, which can significantly degrade the performance of standard FL algorithms [10], [11]. In this work we focus on federated learning over large populations of edge devices operating under such block-based PDC schedules, and we develop Federated Nested Quasi-Newton Optimization (FedNQN), a curvature-aware framework designed to accelerate convergence and enhance robustness in this edge-device PDC regime.

Despite the promise of FL, most existing frameworks assume flexible and frequent client-server communication. This assumption breaks down in real-world deployments where participation is governed by fixed eligibility windows tied to device conditions such as charging status and Wi-Fi connectivity. These constraints define the regime of PDC, in which clients are available only at pre-scheduled intervals. Unlike asynchronous or stochastic participation, PDC imposes structured yet infrequent update schedules that introduce new challenges: global models must evolve with limited and temporally sparse feedback, while client models may drift significantly between communication windows. The effects of data heterogeneity become more pronounced under such constraints, as prolonged local training can lead to divergence and degrade the consistency of the aggregated model [9], [12]. These issues are further intensified in resource-limited environments where computational capabilities and local update quality vary across clients [13].

Existing FL algorithms are not designed to operate under such deterministic and intermittent communication regimes. Many rely on frequent gradient aggregation or adaptive synchronization heuristics that assume continuous client availability and low-cost communication [14]–[16]. These assumptions do not hold in PDC-constrained settings, where clients must update models within limited, non-negotiable windows defined by energy budgets, network access, or compliance requirements [17]. Under these conditions, standard optimization pipelines fail to provide timely feedback, resulting in slow convergence, reduced model quality, and poor adaptation to client heterogeneity. Addressing this disconnect requires a fundamentally different approach that directly incorporates periodic communication structures into the learning process while maintaining robustness to non-IID data and system variability.

In this work, we propose Federated Learning with Nested Quasi-Newton Optimization (FedNQN), a novel framework designed to address the practical challenges posed by PDC in federated systems. Unlike conventional FL methods that assume flexible synchronization or random client selection, FedNQN is explicitly constructed for environments with struc-

L. Zhao is with School of Management, Yale University, 165 Whitney Ave, New Haven, CT, 06511, United States. W-S. Lu, and L. Cai are with the Department of Electrical and Computer Engineering, University of Victoria, 3800 Finnerty Road, Victoria, BC, V8P 5C2, Canada.
*Corresponding author: Lin Cai (E-mail: cai@ece.uvic.ca).

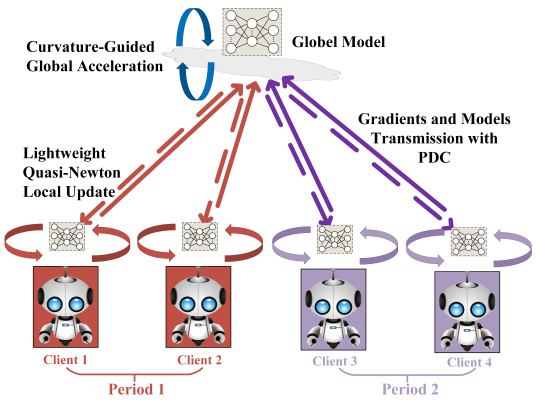


Fig. 1. Overview of FedNQN under periodic deterministic communication constraints.

tured participation schedules and heterogeneous data distributions. As illustrated in Fig. 1, FedNQN integrates lightweight quasi-Newton updates at clients with curvature-guided global refinement at the server, coordinated over fixed PDC intervals.

The core of FedNQN is a nested quasi-Newton optimization framework designed to address the unique challenges posed by PDC constraints, where client participation follows structured system-level schedules rather than random sampling. FedNQN integrates low-overhead inexact quasi-Newton updates at the client level with global curvature-informed acceleration at the server. Specifically, client-side updates leverage local curvature approximations for efficient adaptation under data and resource heterogeneity, while the server aggregates client gradients to construct a spectral inverse Hessian that guides global optimization. Our main contributions are as follows:

- We propose FedNQN, a federated learning framework tailored for PDC scenarios, where client participation is governed by structured system-level schedules instead of random sampling, reflecting practical constraints in real-world applications.
- We introduce a nested quasi-Newton optimization scheme that integrates inexact local quasi-Newton updates with central curvature-guided global refinement, enabling coordinated second-order acceleration at both local and global levels.
- We develop a structured curvature estimation strategy that constructs a spectral inverse Hessian approximation at the server using only aggregated client gradients, enabling global second-order guidance under deterministic participation. This global curvature is further leveraged to drive variance-reduced local updates, enhancing local training stability and coherence with the global objective.
- We conduct comprehensive experiments on multiple benchmarks under various non-IID and PDC conditions. Results show that FedNQN not only achieves faster convergence and higher test accuracy than state-of-the-art baselines, but also maintains competitive communication and computational efficiency. Despite slightly higher per-round costs, FedNQN reaches target performance in fewer rounds, resulting in lower total resource consumption and enhanced practicality for realistic FL deployments.

The rest of this paper is organized as follows. Section II reviews related work, identifying key challenges and gaps in FL under PDC constraints. Section III formalizes the federated optimization problem under PDC constraints, detailing the underlying challenges and defining key optimization objectives. Section IV introduces our proposed Nested Quasi-Newton optimization framework to address these challenges. Section V presents and analyzes experimental results, demonstrating the effectiveness of our approach. Section VI summarizes our contributions and explores potential directions for future research.

II. Related Work

FL has become a cornerstone for privacy-preserving decentralized training, but its deployment in real-world systems may be constrained by communication limitations, device heterogeneity, and non-IID data distributions [1], [3], [18], [19]. One major challenge is, statistical heterogeneity, i.e., local updates diverge under skewed data distributions, hindering convergence to the global optimum [2]. This issue is exacerbated in settings with communication limits as in PDC where clients can exchange information with the central server periodically, so, gradient inconsistencies and stale updates become more severe [8], [12].

The popular FedAvg algorithm [2] simply averages local SGD updates but suffers under heterogeneous conditions. Extensions like FedProx [3] introduce a proximal term to improve stability, while SCAFFOLD [13] employs control variates for variance reduction. Though effective to a degree, these first-order methods degrade when synchronization is infrequent or delayed. Similarly, MARINA [15] and other adaptive sampling methods [16], [20] reduce communication frequency but rely on stochastic participation, making them ill-suited for PDC scenarios, which require deterministic and fixed participation intervals.

To improve robustness under non-IID settings, a broad class of personalization-based methods has emerged. FedRep [21] separates shared encoders and local heads; FedAMP [22] uses attention-based aggregation; and FedBN [23] preserves local batch norm statistics. Later works such as FedPAC [24] meta-learn classifier combinations with aligned features; FedDBE [25] adversarially removes domain bias in representation space; FedALA [26] blends global and local models at initialization; and FedGH [27] trains global headers with graph-regularized objectives. These techniques enhance generalization and representation sharing, yet they assume flexible synchronization and do not account for deterministic schedules or curvature-informed optimization.

Another research line focuses on asynchronous updates and second-order methods. FedNTD [28] filters out extreme gradients under stale conditions; FedBABU [29] aggregates Bayesian posteriors; ASO-Fed [30] enables event-driven updates with adaptive learning rates; and CA^{2}FL [31] combines caching and periodic Hessian-informed step sizing. While these frameworks demonstrate the value of relaxed communication and adaptive aggregation, they do not introduce second-order updates at the local level, nor are they designed for PDC.

TABLE I
METHODOLOGICAL COMPARISON OF FEDERATED LEARNING ALGORITHMS

Method	Local Optimization	Global Optimization	Central Acceleration	Curvature Estimation	Variance Red.	Async
FedNQN	Inexact Quasi-Newton	Exact Quasi-Newton	✓	✓	✓	PDC
FedRep	SGD	Averaging	✗	✗	✗	✗
FedAMP	SGD	Weighted Averaging	✗	✗	✗	✗
FedALA	SGD	Adaptive Averaging	✗	✗	✗	✗
FedPAC	SGD	Meta Fusion	✗	✗	✓	✗
FedBABU	SGD	Bayesian Fusion	✗	✗	✗	✗
SCAFFOLD	SGD	Control Variate Avg.	✗	✗	✓	✗
FedDBE	SGD	Domain Averaging	✗	✗	✓	✗
FedNTD	SGD	Knowledge Distillation	✗	✗	✗	✗
ASO-Fed	SGD	Asynchronous Averaging	✗	✗	✗	✓
CA^2FL	SGD	Cached Async Averaging	✗	✗	✓	✓
FedGH	SGD	Graph Aggregation	✗	✗	✓	✓

Table I summarizes the key methodological characteristics of the aforementioned baselines.

However, real-world deployments of FL reflect the constraints imposed by PDC constraints. Prior studies have shown that client updates are typically triggered only when devices meet strict eligibility criteria such as being idle, charging, and connected to an unmetered network [6], [7]. These hardware and connectivity-driven conditions give rise to predictable but infrequent client participation, forming the basis of the PDC regime. Under such conditions, traditional federated optimization methods struggle to adapt, as they are not designed to account for the rigid structure of client participation.

Although existing methods have made significant progress in addressing statistical heterogeneity, personalization, and communication efficiency, they largely assume random or adaptive client availability. Consequently, these approaches are ill-suited to environments where participation schedules are fixed over extended communication periods. Their optimization frameworks do not explicitly account for the timing and structure of participation, which limits their effectiveness under deterministic communication constraints. This reveals a critical gap in the current literature, i.e., the lack of optimization strategies that are natively compatible with PDC constraints.

To address this shortcoming, we propose a Nested Quasi-Newton optimization framework designed specifically for FL under PDC constraints. At the global level, our method employs curvature-aware acceleration based on second-order approximations to refine model updates and offset the effects of infrequent client rescheduling. Locally, we incorporate variance-controlled optimization to mitigate client drift and enhance training efficiency under data heterogeneity. By embedding the periodic schedule directly into both global and local optimization dynamics, our approach enables stable convergence and robust performance in resource-constrained federated systems. FedNQN thus provides a principled and scalable solution for structured participation environments.

III. PROBLEM FORMULATION

FL enables decentralized clients to collaboratively train a global model while preserving data privacy. However, real-world deployments often face PDC constraints, where updates between clients and the central server occur at fixed intervals. These constraints disrupt synchronization, slow convergence, and exacerbate model drift, particularly in heterogeneous environments where clients possess non-IID data and varying computational resources. Since only a fixed subset of clients participates within each communication period, delayed synchronization arises as inactive clients must wait for their next scheduled round to contribute, leading to imbalanced updates. Additionally, statistical heterogeneity further amplifies model inconsistency, making convergence more challenging. Addressing these issues requires an optimization framework that effectively aligns local learning with global updates while mitigating the inefficiencies caused by limited communication.

A. Federated Learning Objective

Consider a distributed system in which a total of n training samples, denoted as $\{\boldsymbol{x_k}, y_k\}_{k=1}^{n}$, are distributed across a set of clients E. Each client $i \in E$ holds a local dataset P_i of size n_i, satisfying the relationship $n = \sum_{i=1}^{|E|} n_i$. The global learning objective is to minimize a weighted sum of local objective functions

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\textbf{minimize}} \quad \boldsymbol{f(w)} = \sum_{i=1}^{|E|} \frac{n_i}{n} f_i(\boldsymbol{w}), \tag{1}$$

where $f_i(\boldsymbol{w})$ represents the local loss function at client i, defined as

$$f_i(\boldsymbol{w}) = \frac{1}{n_i} \sum_{k\in P_i} \boldsymbol{F_k(w)}, \quad i = 1, \ldots, |E|, \tag{2}$$

where $\boldsymbol{F_k(w)}$ denotes the loss evaluated on the k-th local sample. While this objective provides a general formulation for FL, practical implementations face additional complexities due to communication constraints, non-uniform client participation, and heterogeneous data distributions.

B. Periodic Client Participation and Anchor Gradient

Unlike conventional FL frameworks where clients are randomly selected at each round, PDC imposes a structured participation scenario in which only a fixed subset of clients $C^r \subset E$ is participated within a periodic of communication rounds. This structured participation reflects real-world constraints, where network availability, energy limitations, and security policies dictate when clients can communicate with the central server.

At the beginning of each global round $r \in \{1, \ldots, R\}$, the central server transmits the current global model $\boldsymbol{w}^r$ to the participating clients in C^r. Each client $i \in C^r$ then computes its full local gradient using its entire local dataset and uploads it to the server. The server aggregates these client gradients into an anchor gradient as

$$\boldsymbol{g}(\boldsymbol{w}^r) = \sum_{i \in C^r} \frac{n_i}{n^r} \nabla f_i(\boldsymbol{w}^r), \tag{3}$$

where $n^r = \sum_{i \in C^r} n_i$ represents the total sample size of the participating clients.

This anchor gradient provides a representative summary of the participating clients' gradients during each period. Although client selection remains fixed within a communication period, the subset C^r varies across different periods. When analyzed at the level of full communication periods rather than individual rounds, this periodic variation effectively mimics random participation, ensuring that the aggregated gradient remains an unbiased estimator of the full global gradient, i.e.,

$$\mathbb{E}[\boldsymbol{g}(\boldsymbol{w}^r)] = \nabla f(\boldsymbol{w}^r), \tag{4}$$

where the expectation is taken over client participation across multiple communication periods.

C. Challenges in Periodic Deterministic Communication

Building on the structured participation model and anchor gradient, we now examine the practical challenges that arise under the PDC regime. While the anchor gradient provides an unbiased approximation of the global gradient across periods, the fixed client subset in each communication window introduces several optimization difficulties in practice.

Let τ denote the length in rounds of each communication period. We divide training into consecutive intervals indexed by $p = 0, 1, 2, \ldots$, where each period comprises rounds

$$r \in \{p\tau,\ p\tau + 1,\ \ldots,\ (p+1)\tau - 1\}. \tag{5}$$

Within each period p, a fixed subset of clients $C^p \subset E$ is selected and remains active throughout the entire period. The participation of each client $i \in E$ at round r is encoded by a binary indicator

$$a_{i,r} = \begin{cases} 1, & \text{if } i \in C^{\lfloor r/\tau \rfloor}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

which determines whether client i is permitted to upload updates in round r.

This deterministic participation pattern reflects constraints in real-world FL deployments, such as device operation schedule and availability, connectivity, or privacy constraints. However, it also introduces complications that degrade optimization efficiency. Clients that are inactive for extended periods may accumulate outdated local models, especially in settings where the global model evolves rapidly. Upon rejoining, such clients can produce updates that conflict with the current optimization direction, slowing convergence or introducing oscillations for the whole training process.

These difficulties are exacerbated by data heterogeneity. When local datasets are non-IID, the divergence in update directions is naturally more severe. The fixed-period exclusion of some clients intensifies this misalignment, as the gradients from outdated local models become poorly aligned with the global objective. Moreover, variations in local dataset sizes and computational capacity result in uneven contributions to the aggregated anchor gradient, further destabilizing learning dynamics.

To address these challenges, it is necessary to develop an optimization framework that is robust against delayed, inconsistent, and imbalanced client updates. In the following section, we introduce FedNQN, which integrates local quasi-Newton updates with curvature-aware acceleration at the server to restore stability and efficiency in training under PDC constraints.

IV. Nested Quasi-Newton Optimization for Federated Learning

To address the optimization challenges introduced by PDC constraints, we propose a Nested Quasi-Newton Optimization (FedNQN) framework that integrates curvature-aware updates at both the global server and participating clients. The goal is to maintain efficient convergence and robust learning under limited and structured client participation.

At the global level, the server performs curvature-guided updates by maintaining an inverse Hessian approximation, which accelerates optimization using second-order information. At the client level, local updates approximate this global curvature guidance while remaining computationally lightweight. Rather than computing full Hessian approximations, clients perform variance-controlled updates tailored to their local data distributions.

The nested design aligns global and local learning dynamics where global curvature updates refine the descent direction for all clients, while local variance-reduced steps stabilize contributions from heterogeneous data. This hierarchical coordination ensures model synchronization and convergence efficiency, even under infrequent client rescheduling. By combining second-order acceleration with communication-aware design, FedNQN offers an effective and scalable optimization framework for FL under real-world constraints.

A. Curvature-Guided Central Acceleration under Periodic Deterministic Communication Constraints

The Nested Quasi-Newton framework adopts a hierarchical optimization structure, where the global model is accelerated

at the server using curvature information, and clients locally approximate this trajectory through lightweight quasi-Newton updates. This structure is specifically designed to address challenges arising from PDC constraints, where clients only participate at fixed intervals. Under such constraints, maximizing the effectiveness of each global update becomes critical for maintaining efficient convergence.

To meet this challenge, the central server constructs curvature-aware updates by estimating second-order information from anchor gradients accumulated across communication rounds. These anchor gradients encode the effect of past updates on the global loss landscape, allowing the server to approximate curvature without explicit Hessian computation.

At each global round r, the server begins with a first-order Taylor expansion around the current model $\boldsymbol{w}^r$

$$\hat{f}(\hat{\boldsymbol{w}}^r) = f(\boldsymbol{w}^r) + \boldsymbol{\nabla f}(\boldsymbol{w}^r)^\top(\hat{\boldsymbol{w}}^r - \boldsymbol{w}^r), \tag{7}$$

and refines this estimate by incorporating second-order curvature

$$\hat{f}(\hat{\boldsymbol{w}}^r) + \frac{1}{2}(\hat{\boldsymbol{w}}^r - \boldsymbol{w}^r)^\top \boldsymbol{\nabla^2 f}(\boldsymbol{w}^r)(\hat{\boldsymbol{w}}^r - \boldsymbol{w}^r). \tag{8}$$

This curvature-adjusted approximation provides a more accurate model of the objective's geometry and enables reliable global descent steps, even when client feedback is delayed or sparse due to PDC.

To estimate the global curvature efficiently, the server avoids direct computation of the Hessian $\boldsymbol{\nabla^2 f}(\boldsymbol{w}^r)$ by analyzing variations in anchor gradients across successive global rounds. Specifically, it constructs a curvature pair using

$$\boldsymbol{\gamma}_r = \boldsymbol{g}(\boldsymbol{w}^r) - \boldsymbol{g}(\boldsymbol{w}^{r-1}) \approx \boldsymbol{\nabla^2 f}(\boldsymbol{w}^r)\boldsymbol{\delta}_r, \tag{9}$$

where $\boldsymbol{\delta}_r = \boldsymbol{w}^r - \boldsymbol{w}^{r-1}$ denotes the global model update. This curvature pair $(\boldsymbol{\delta}_r, \boldsymbol{\gamma}_r)$ enables implicit estimation of second-order information based solely on server-side observations.

This anchor-gradient-based approximation is particularly well-suited for FL under PDC, where direct and frequent client communication is infeasible. By accumulating model and gradient trajectories over time, the server maintains a global view of the optimization landscape and can construct curvature-aware updates even in the presence of delayed or sporadic feedback.

To apply curvature-guided acceleration, the server uses its inverse Hessian approximation $\boldsymbol{S}_r \approx \left(\boldsymbol{\nabla^2 f}(\boldsymbol{w}^r)\right)^{-1}$ to compute the global descent direction

$$\boldsymbol{d}_g^r = -\boldsymbol{S}_r \boldsymbol{g}(\boldsymbol{w}^r). \tag{10}$$

This direction defines a refined optimization trajectory that captures the curvature of the global objective. Based on this, the server computes the intermediate accelerated model

$$\hat{\boldsymbol{w}}^r = \boldsymbol{w}^r - \alpha_g^r \boldsymbol{d}_g^r, \tag{11}$$

where α_g^r is the global learning rate. This curvature-adjusted model $\hat{\boldsymbol{w}}^r$ is then used as the reference point for subsequent local updates and for aggregating client contributions in the global model update step. The next task is to update the curvature matrix $\boldsymbol{S}_r$ using the observed curvature pair, ensuring that future directions remain well-aligned with the evolving global landscape.

To refine the curvature information over time, the server updates its inverse Hessian approximation $\boldsymbol{S}_r$ using the classical BFGS formula as

$$\begin{aligned}\boldsymbol{S}_{r+1} =& \boldsymbol{S}_r + \left(1 + \frac{\boldsymbol{\gamma}_r^\top \boldsymbol{S}_r \boldsymbol{\gamma}_r}{\boldsymbol{\gamma}_r^\top \boldsymbol{\delta}_r}\right)\frac{\boldsymbol{\delta}_r \boldsymbol{\delta}_r^\top}{\boldsymbol{\gamma}_r^\top \boldsymbol{\delta}_r} \\ & - \frac{\boldsymbol{\delta}_r \boldsymbol{\gamma}_r^\top \boldsymbol{S}_r + \boldsymbol{S}_r \boldsymbol{\gamma}_r \boldsymbol{\delta}_r^\top}{\boldsymbol{\gamma}_r^\top \boldsymbol{\delta}_r},\end{aligned} \tag{12}$$

where $(\boldsymbol{\delta}_r, \boldsymbol{\gamma}_r)$ is the curvature pair constructed from consecutive global rounds. The update in (12) is derived by satisfying the secant condition $\boldsymbol{\gamma}_r = \boldsymbol{\nabla^2 f}(\boldsymbol{w}^r)\boldsymbol{\delta}_r$ while preserving symmetry and positive definiteness. Rather than updating the Hessian directly, the server maintains its inverse approximation $\boldsymbol{S}_r \approx \left(\boldsymbol{\nabla^2 f}(\boldsymbol{w}^r)\right)^{-1}$ and applies the Sherman-Morrison-Woodbury identity to ensure efficient and stable updates. This approach provides curvature-informed descent directions with minimal overhead and is widely used in second-order optimization methods [32].

While this update rule follows the classical BFGS structure, its application in the federated setting under PDC is a key innovation of our design. Unlike centralized optimization where curvature pairs are readily available, our method computes and accumulates curvature information exclusively at the server using anchor gradients and model differences, without requiring clients to transmit additional statistics. This adaptation allows the central server to perform second-order updates that reflect the global optimization landscape, even in the absence of synchronous or frequent client feedback.

By integrating this curvature-informed acceleration into each global round, the server establishes a refined optimization trajectory that guides the evolution of the global model. When clients reconnect under PDC, they initialize their local updates from this accelerated model $\hat{\boldsymbol{w}}^r$, thereby inheriting curvature-adaptive progress without incurring the overhead of second-order computations. This structure ensures that local optimization remains lightweight while globally coordinated, improving convergence and stability under constrained communication and heterogeneous data.

To characterize the effectiveness of the central curvature-guided update step under PDC, we present Theorem 1 as following.

Theorem 1 (Global Convergence of FedNQN under PDC Constraints): Suppose the global objective $f(\boldsymbol{w})$ is μ-strongly convex and L-smooth, and each local function $f_i(\boldsymbol{w})$ is μ_i-strongly convex and β_i-smooth. Define the aggregated curvature bounds as

$$\mu = \sum_{i=1}^{|E|} \frac{n_i}{n}\mu_i, \quad \beta = \sum_{i=1}^{|E|} \frac{n_i}{n}\beta_i. \tag{13}$$

Assume the global anchor gradient satisfies

$$\mathbb{E}[\boldsymbol{g}(\boldsymbol{w}^r)] = \nabla f(\boldsymbol{w}^r), \quad \mathbb{E}\left[\|\boldsymbol{g}(\boldsymbol{w}^r) - \nabla f(\boldsymbol{w}^r)\|^2\right] \le \sigma^2. \tag{14}$$

Let the server maintain a curvature approximation $\boldsymbol{S}_r$ such that

$$m\boldsymbol{I} \preceq \boldsymbol{S}_r \preceq M\boldsymbol{I} \tag{15}$$

TABLE II
DESCRIPTION OF NOTATIONS

NOTATION	DESCRIPTION
$\boldsymbol{f(w)}$	Global objective function.
$f_i(\boldsymbol{w})$	Local objective function for client i.
$\boldsymbol{w}^r$	Global model at round r.
$\hat{\boldsymbol{w}}^r$	Curvature-adjusted global model at round r.
$\hat{\boldsymbol{w}}^{\boldsymbol{r}}_{\boldsymbol{i,k}}$	Local model of client i at step k of round r.
C^r	Set of clients participating in round r.
E	Full set of clients in the system.
n_i	Number of local training samples at client i.
n^r	Total number of samples across all C^r.
α_l^i, α_g^r	Local and global learning rates.
$\boldsymbol{g}(\boldsymbol{w}^r)$	Anchor gradient at round r.
$\boldsymbol{d}_g^r$	Curvature-guided descent direction at the server.
$\boldsymbol{S}_r$	Inverse Hessian approximation at round r.
$\boldsymbol{\delta}_r$	Global model difference $\boldsymbol{w}^r - \boldsymbol{w}^{r-1}$.
$\boldsymbol{\gamma}_r$	Gradient difference $\boldsymbol{g}(\boldsymbol{w}^r) - \boldsymbol{g}(\boldsymbol{w}^{r-1})$.
K	Number of local steps per round.
τ	Communication period under PDC constraints.
$\boldsymbol{d}^r_{i,k-1}$	Local quasi-Newton update direction at client i.
$\boldsymbol{q}^r_{i,k-1}$	Adjusted local gradient at client i and step $k-1$.
$\rho^r_{i,k-1}$	Curvature coefficient at client i and step $k-1$.
n^j	Global sample count for feature j.
n_i^j	Local sample count for feature j at client i.
$\boldsymbol{\lambda}_i$	Local scaling vector for feature-aware correction.
$\boldsymbol{\Lambda}_i$	Diagonal matrix of local scaling factors.
a^j	Global feature-aware aggregation weight for feature j.
$\boldsymbol{A}_r$	Global diagonal aggregation matrix at round r.

for constants $0 < m \leq M$, and update the global model as

$$\boldsymbol{w}^{r+1} = \boldsymbol{w}^r - \alpha \boldsymbol{S}_r \boldsymbol{g}(\boldsymbol{w}^r), \tag{16}$$

with step size $\alpha \leq \min\left\{\frac{1}{L}, \frac{1}{M\mu}\right\}$.

Under PDC with fixed period τ, suppose the client participation schedule ensures adequate gradient consistency over time. Then the iterates $\{\boldsymbol{w}^r\}$ satisfy

$$\mathbb{E}[f(\boldsymbol{w}^r) - f^*] \leq (1 - \alpha\mu m)^r [f(\boldsymbol{w}^0) - f^*] + \frac{\alpha M \sigma^2}{2\mu}, \tag{17}$$

where $f^* = f(\boldsymbol{w}^*)$ is the global minimum. Moreover, if $\boldsymbol{S}_r$ converges to $(\nabla^2 f(\boldsymbol{w}^*))^{-1}$ as $r \to \infty$, then the updates achieve superlinear convergence once $\boldsymbol{w}^r$ enters a sufficiently small neighborhood of $\boldsymbol{w}^*$.

B. Curvature-Aware Variance-Controlled Local Acceleration

The local acceleration mechanism is an integral component of the nested quasi-Newton optimization framework, where local updates approximate the curvature-refined global descent step. This nested structure ensures that local training aligns with the global optimization trajectory, mitigating model drift and improving synchronization across clients. By incorporating variance-controlled gradient corrections with an approximated quasi-Newton optimization step, each client refines its updates using local curvature information while maintaining computational feasibility. This hierarchical coordination between local and global optimization enhances convergence efficiency, stabilizes learning in the presence of data heterogeneity, and preserves the curvature-aware acceleration strategy under PDC constraints.

1) Variance-Controlled Gradient Correction: Once the accelerated global model $\hat{\boldsymbol{w}}^{\boldsymbol{r}}$ is computed at the server, it is distributed to the participating clients $i \in C^r$, initializing local models as $\hat{\boldsymbol{w}}^{\boldsymbol{r}}_{\boldsymbol{i,0}} = \hat{\boldsymbol{w}}^{\boldsymbol{r}}$. Each client then performs K local updates using its dataset, incorporating variance-controlled corrections to mitigate gradient noise while maintaining computational efficiency. At each local step, client i samples a mini-batch and computes the stochastic gradient $\boldsymbol{g_i}(\hat{\boldsymbol{w}}^{\boldsymbol{r}}_{\boldsymbol{i,k-1}})$, which is then adjusted using the global curvature information to ensure alignment with the refined optimization trajectory established by the central acceleration step.

Local data heterogeneity introduces gradient variance, which can cause model drift and inconsistent updates across clients. To mitigate these issues and reinforce the nested optimization structure, each local gradient is corrected using the global anchor gradient

$$\boldsymbol{g_i}(\hat{\boldsymbol{w}}^{\boldsymbol{r}}_{\boldsymbol{i,k-1}}) - \boldsymbol{g_i}(\hat{\boldsymbol{w}}^r) + \boldsymbol{\nabla f}(\boldsymbol{w}^r), \tag{18}$$

where $\boldsymbol{\nabla f}(\boldsymbol{w}^r)$ acts as a global curvature-guided correction, ensuring local updates follow a direction consistent with the global acceleration step. This correction not only reduces variance but also maintains stability under PDC constraints, preserving the integrity of the hierarchical optimization structure.

We consider each sample $x \in \mathbb{R}^d$ with coordinates indexed by $j \in \{1, \ldots, d\}$. Feature j is defined as the j-th coordinate x^j of x after standard preprocessing, such as normalization or one-hot encoding. For client i with local dataset $\mathcal{D}_i$ of size n_i, the local count of feature j is

$$n_i^j = \sum_{x \in \mathcal{D}_i} \mathbf{1}\big[x^j \neq 0\big]. \tag{19}$$

This quantity records how often feature j appears in the samples stored at client i. For dense normalized features, $x^j \neq 0$ holds for almost all samples and thus $n_i^j \approx n_i$. For sparse or one-hot encoded representations, n_i^j reflects the empirical occurrence frequency of the corresponding coordinate. When appropriate, an equivalent thresholded form $\mathbf{1}\big[|x^j| > \varepsilon\big]$ can be used to define presence.

Let

$$n = \sum_i n_i, \qquad n^j = \sum_i n_i^j \tag{20}$$

denote the global sample size and the global count of feature j. The occurrence rate of feature j at client i is n_i^j/n_i, and the corresponding global occurrence rate is n^j/n. The feature scaling factor assigned to client i on coordinate j is

$$\lambda_i^j = \frac{n^j/n}{n_i^j/n_i}, \qquad \boldsymbol{\Lambda}_i = \mathrm{diag}(\boldsymbol{\lambda}_i). \tag{21}$$

If feature j is over-represented at client i relative to the global distribution, then n_i^j/n_i exceeds n^j/n and $\lambda_i^j < 1$, which down-scales its contribution in the local update. If feature j is under-represented, then $\lambda_i^j > 1$ and its influence is up-scaled. When local and global occurrence rates match, $\lambda_i^j = 1$ and no correction is applied. In this way, the feature-based reweighting corrects client drift that arises from systematic feature-distribution mismatch instead of modifying the model architecture or accessing raw data.

The protocol is communication-efficient and privacy-preserving. Each client i computes its local counts $\{n_i^j\}_{j=1}^d$ and its local size n_i, and sends only these aggregated statistics to the server. The server aggregates to obtain $\{n^j\}_{j=1}^d$ and n, then broadcasts the global occurrence rates $\{n^j/n\}_{j=1}^d$ to all clients. Each client reconstructs $\{\lambda_i^j\}_{j=1}^d$ locally and applies the diagonal scaling matrix $\mathbf{\Lambda}_i$ to its feature-scaled gradients or to the model update direction as described in Section IV-B. No individual samples or per-example feature vectors leave the clients.

To further balance contributions from clients with different data volumes, the local learning rate at client i is normalized as

$$\alpha_l^i = \frac{\alpha_l}{n_i}, \tag{22}$$

where α_l is a base step size that serves as the initial local rate before any internal adjustments by the chosen local optimizer. This normalization prevents clients with small datasets from producing disproportionately large updates and complements the feature-level scaling provided by $\mathbf{\Lambda}_i$.

To explicitly integrate variance-controlled updates within the global curvature-aware framework, the update dynamic $\boldsymbol{\delta}_{i,k-1}^r$ is defined as

$$\boldsymbol{\delta}_{i,k-1}^r = -\alpha_l^i[\mathbf{\Lambda}_i(\boldsymbol{g}_i(\hat{\boldsymbol{w}}_{i,k-1}^r) - \boldsymbol{g}_i(\hat{\boldsymbol{w}}^r)) + \boldsymbol{g}(\boldsymbol{w}^r)], \tag{23}$$

where the term $\boldsymbol{g}_i(\hat{\boldsymbol{w}}_{i,k-1}^r) - \boldsymbol{g}_i(\hat{\boldsymbol{w}}^r)$ captures deviations from the global reference, while $\mathbf{\Lambda}_i$ ensures corrections are weighted appropriately based on local and global feature distributions. The inclusion of $\boldsymbol{g}(\boldsymbol{w}^r)$ serves as a global anchor, preventing local models from diverging excessively due to heterogeneous data distributions. This formulation maintains a balance between local adaptability and global consistency, mitigating the effects of statistical heterogeneity.

The above variance-controlled correction mechanism plays a critical role in reinforcing the nested quasi-Newton optimization structure by allowing local updates to approximate the curvature-guided global acceleration step. By maintaining local-global consistency, the framework achieves faster convergence, reduced communication overhead, and greater stability under PDC constraints, making it well-suited for FL environments with resource limitations.

2) Curvature-Aware Local Gradient Adjustment: To align local updates with global optimization, each client refines its local model using curvature-aware adjustments that approximate the global acceleration step. This ensures that despite the constraints of PDC schedules, local updates remain synchronized with the global optimization trajectory. Given the computed update dynamic, the intermediate local model $\boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}}^{\boldsymbol{r}}$ is updated as

$$\boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}}^{\boldsymbol{r}} = \hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k-1}}^{\boldsymbol{r}} + \boldsymbol{\delta}_{\boldsymbol{i},\boldsymbol{k-1}}^{\boldsymbol{r}}. \tag{24}$$

Variance-controlled gradient correction stabilizes local updates while allowing adaptation to non-IID data distributions.

After computing the intermediate update $\boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}}^{\boldsymbol{r}}$, each client evaluates the local curvature by measuring the gradient change between consecutive updates. This gradient variation provides an estimate of the local Hessian dynamics, which is essential for refining update directions. The gradient change is given by

$$\boldsymbol{\gamma}_{i,k-1}^r = \boldsymbol{g}_{\boldsymbol{i}}(\boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}}^{\boldsymbol{r}}) - \boldsymbol{g}_{\boldsymbol{i}}(\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k-1}}^{\boldsymbol{r}}), \tag{25}$$

which captures how the local gradient evolves in response to parameter updates, effectively incorporating second-order information into the optimization process. Using this curvature-sensitive adjustment, the local update direction is computed as

$$\boldsymbol{d}_{\boldsymbol{i},\boldsymbol{k}}^{\boldsymbol{r}} = -\boldsymbol{S}_{i,k}^r \boldsymbol{g}_{\boldsymbol{i}}(\boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}}^{\boldsymbol{r}}),$$

where $\boldsymbol{S}_{i,k}^r$ serves as an approximation of the inverse Hessian, leveraging historical curvature information to refine the update step, formulated as

$$\begin{aligned}\boldsymbol{S}_{i,k}^r = \boldsymbol{S}_{i,k-1}^r &+ \left(1 + \frac{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{S}_{i,k-1}^r\boldsymbol{\gamma}_{i,k-1}^r}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r}\right)\frac{\boldsymbol{\delta}_{i,k-1}^r\boldsymbol{\delta}_{i,k-1}^{r\top}}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r} \\ &- \frac{\boldsymbol{S}_{i,k-1}^r\boldsymbol{\gamma}_{i,k-1}^r\boldsymbol{\delta}_{i,k-1}^{r\top} + \boldsymbol{\delta}_{i,k-1}^r\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{S}_{i,k-1}^r}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r},\end{aligned} \tag{26}$$

which can be expressed in a simplified form as

$$\begin{aligned}\boldsymbol{S}_{i,k}^r = &\left(\boldsymbol{I} - \frac{\boldsymbol{\delta}_{i,k-1}^r\boldsymbol{\gamma}_{i,k-1}^{r\top}}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r}\right)\boldsymbol{S}_{i,k-1}^r\left(\boldsymbol{I} - \frac{\boldsymbol{\gamma}_{i,k-1}^r\boldsymbol{\delta}_{i,k-1}^{r\top}}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r}\right) \\ &+ \frac{\boldsymbol{\delta}_{i,k-1}^r\boldsymbol{\delta}_{i,k-1}^{r\top}}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r}.\end{aligned} \tag{27}$$

By incorporating this approximation, each client adapts its update direction dynamically, ensuring that the step size and orientation are adjusted based on local curvature characteristics. This method enhances convergence stability while maintaining computational efficiency in resource-limited FL environments.

FedNQN operates under block-based PDC-style participation. In each communication block, only a subset of edge devices is active, and this subset may change across blocks or experience occasional missed uploads. The server maintains its quasi-Newton state using consecutive global model updates, so the curvature information and the global search direction are defined at the aggregate level and do not depend on fixed client schedules. The limited-memory update naturally refreshes as participation patterns change, which allows FedNQN to remain stable under realistic PDC-style irregularity.

3) Efficient Curvature Approximation and Adaptive Local Step Optimization: Given the computational limitations of individual clients, maintaining or updating the full inverse Hessian approximation $\boldsymbol{S}_{i,k-1}^r$ can be prohibitively expensive in terms of both storage and computation. To alleviate this burden, the curvature-aware local acceleration adopts a low-cost approximation by initializing the inverse Hessian as an identity matrix, then, we have

$$\begin{aligned}\boldsymbol{S}_{i,k}^r = &\left(\boldsymbol{I} - \frac{\boldsymbol{\delta}_{i,k-1}^r\boldsymbol{\gamma}_{i,k-1}^{r\top}}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r}\right)\left(\boldsymbol{I} - \frac{\boldsymbol{\gamma}_{i,k-1}^r\boldsymbol{\delta}_{i,k-1}^{r\top}}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r}\right) \\ &+ \frac{\boldsymbol{\delta}_{i,k-1}^r\boldsymbol{\delta}_{i,k-1}^{r\top}}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r},\end{aligned} \tag{28}$$

Algorithm 1 Nested Quasi-Newton Federated Learning

Input: Initial global model $\boldsymbol{w}^0$, step sizes α_g^r, α_l^i, period τ, local steps K
Initialize global curvature approximation $\boldsymbol{S}_0 = \boldsymbol{I}$
for global round $r = 0, 1, 2, \ldots$ **do**
  Server executes:
  Conduct central acceleration via (10) and (11)
  Select clients C^r under PDC and broadcast $\hat{\boldsymbol{w}}^r$ and anchor gradient $\boldsymbol{g}(\boldsymbol{w}^r)$
  for each client $i \in C^r$ **in parallel do**
    Initialize $\hat{\boldsymbol{w}}_{i,0}^r = \hat{\boldsymbol{w}}^r$, $\boldsymbol{S}_{i,0}^r = \boldsymbol{I}$
    for local step $k = 1, \ldots, K$ **do**
      Compute $\boldsymbol{\delta}_{i,k-1}^r$ and $\boldsymbol{y}_{i,k}^r$ via (23) and (24)
      Estimate gradient change $\boldsymbol{\gamma}_{i,k-1}^r$ via (25)
      Compute curvature coefficient $\rho_{i,k-1}^r$ via (29)
      Compute gradient alignment scalar $t_{i,k-1}^r$ via (31)
      Construct adjusted local gradient $\boldsymbol{q}_{i,k-1}^r$ via (32)
      Compute descent direction $\boldsymbol{d}_{i,k}^r$ via (33)
      Update $\hat{\boldsymbol{w}}_{i,k}^r$ using $\boldsymbol{d}_{i,k}^r$ via (34)
    end for
    Return accumulated update $\boldsymbol{u}_i^r = \sum_{k=1}^K \alpha_l^i \boldsymbol{d}_{i,k-1}^r$
  end for
  Server aggregates:
  Compute feature-aware scaling matrix $\boldsymbol{A}_r$ via (38)
  Update global model $\boldsymbol{w}^{r+1}$ via (39)
  Update global curvature approximation $\boldsymbol{S}_{r+1}$ via (12)
end for

which significantly reduces the computational overhead by avoiding explicit Hessian inverse approximation, instead updating the inverse Hessian approximation incrementally using only local gradient differences $\boldsymbol{\gamma}_{i,k-1}^r$ and update steps $\boldsymbol{\delta}_{i,k-1}^r$. By maintaining a compact and efficient structure, this method enables local updates to benefit from curvature information without excessive memory or processing costs. Furthermore, this approximation strikes a balance between computational efficiency and optimization performance, ensuring that curvature-guided updates remain effective while keeping the method scalable for FL under PDC constraints.

To effectively incorporate curvature information into the optimization process, we leverage the observation that $\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r$ encodes the curvature along the update direction $\boldsymbol{\delta}_{i,k-1}^r$. Based on this, we define the reciprocal curvature scaling factor

$$\rho_{i,k-1}^r = \frac{1}{\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{\delta}_{i,k-1}^r}, \tag{29}$$

which is motivated by the first-order Taylor expansion

$$\boldsymbol{\gamma}_{i,k-1}^r \approx \nabla^2 f_i(\tilde{\boldsymbol{y}})\boldsymbol{\delta}_{i,k-1}^r \tag{30}$$

evaluated at one intermediate point $\tilde{\boldsymbol{y}}$, which suggests that $\rho_{i,k-1}^r$ approximates the inverse local curvature along the update direction. By using $\rho_{i,k-1}^r$ as a scaling factor, we adjust the update step to reflect curvature information, thereby improving the adaptation of local updates to the loss landscape.

Next, we quantify how much the gradient has changed in the direction of the most recent parameter update, which provides additional curvature-sensitive information captured by the scalar quantity

$$t_{i,k-1}^r = \boldsymbol{\delta}_{i,k-1}^{r\top}\boldsymbol{g}(\boldsymbol{y}_{i,k}^r), \tag{31}$$

which is essential for constructing rank-one curvature corrections, as it encodes the alignment between the new gradient and the previous update direction. By incorporating $t_{i,k-1}^r$ into the curvature estimation, we ensure that the updated approximation remains adaptive to newly observed curvature variations. Using $\rho_{i,k-1}^r$ and $t_{i,k-1}^r$, we compute an adjusted gradient-like quantity

$$\boldsymbol{q}_{i,k-1}^r = \boldsymbol{g}(\boldsymbol{y}_{i,k}^r) - \rho_{i,k-1}^r t_{i,k-1}^r \boldsymbol{\gamma}_{i,k-1}^r. \tag{32}$$

This adjustment removes a rank-one correction term $\rho_{i,k-1}^r t_{i,k-1}^r \boldsymbol{\gamma}_{i,k-1}^r$ which injects a second-order curvature refinement into the gradient update. $\rho_{i,k-1}^r$ modulates the correction based on local curvature, while $t_{i,k-1}^r$ quantifies the alignment of new gradient information with the prior update direction. This combination enables the framework to retain adaptive second-order updates while remaining computationally efficient in FL environments.

Once the adjusted gradient $\boldsymbol{q}_{i,k-1}^r$ is computed, it is used to refine the descent direction for the accelerated local update by incorporating both gradient information and curvature-sensitive scaling. The local acceleration direction is defined as

$$\boldsymbol{d}_{i,k}^r = \rho_{i,k-1}^r(\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{q}_{i,k-1}^r - t_{i,k-1}^r)\boldsymbol{\delta}_{i,k-1}^r - \boldsymbol{q}_{i,k-1}^r, \tag{33}$$

where the term $\boldsymbol{\gamma}_{i,k-1}^{r\top}\boldsymbol{q}_{i,k-1}^r$ provides an additional curvature-aware scaling factor, capturing the interaction between the newly adjusted gradient and the historical curvature. By subtracting $t_{i,k-1}^r$, this update balances the influence of the previous local step against the newly corrected gradient, ensuring that the update direction remains adaptive to the evolving local curvature. The local model update is then performed as

$$\hat{\boldsymbol{w}}_{i,k+1}^r = \hat{\boldsymbol{w}}_{i,k}^r + \alpha_l^i \boldsymbol{d}_{i,k}^r, \tag{34}$$

where α_l^i is the local step size for client i. By updating the local model along $\boldsymbol{d}_{i,k}^r$, the optimization process integrates curvature-aware adjustments that account for both past parameter changes $\boldsymbol{\delta}_{i,k-1}^r$ and gradient variations $\boldsymbol{\gamma}_{i,k-1}^r$. This dual incorporation refines the local descent direction, enhancing stability and computational efficiency while ensuring alignment with the global curvature-guided optimization trajectory. After K local iterations, the final local model for client i is obtained by aggregating all curvature-informed updates

$$\hat{\boldsymbol{w}}_{\boldsymbol{i,K}}^{\boldsymbol{r}} = \hat{\boldsymbol{w}}^r + \sum_{k=1}^K \alpha_l^i \boldsymbol{d}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}}, \tag{35}$$

ensuring that each client's model integrates a sequence of curvature-aware refinements, maintaining alignment with the global optimization process. By iteratively adjusting the update direction based on second-order information, the method enhances convergence speed while preserving robustness against data heterogeneity and communication constraints.

To demonstrate that local training under the variance-reduced quasi-Newton step remains globally aligned, we establish the following geometric convergence guarantee for local updates as shown in Theorem 2.

Theorem 2 (Local Convergence of Inexact Quasi-Newton Updates): Assume that each client's local objective $f_i(w)$ is μ_i-strongly convex and β_i-smooth. At global round r, let client i initialize its local model as $\hat{w}_{i,0}^r = \hat{w}^r$ and perform K local inexact quasi-Newton updates using the variance-reduced gradient estimator

$$g_i(w) - g_i(w^r) + g(w^r), \tag{36}$$

anchored by the global gradient $g(w^r)$. Let the local step size satisfy $\alpha_l \leq \frac{\mu_i}{5\beta_i^2 e}$ for a bounded constant $e > 0$ that relates the local curvature at $\hat{w}_{i,k}^r$ to the central iterate w^r, then the local model satisfies the following geometric contraction:

$$\mathbb{E}\left[\left\|\hat{w}_{i,K}^r - w^*\right\|^2\right] \leq \exp\left(-\frac{\mu_i^2 K}{5\beta_i^2 e}\right)\left\|w^r - w^*\right\|^2, \tag{37}$$

where w^* is the global optimum. This result shows that the local updates are aligned with the global optimization direction, and the model remains close to w^* with exponentially decaying error in K. The contraction ensures that the anchor gradient $g(w^r)$ remains a stable estimator, thereby supporting efficient central updates under PDC.

C. Feature-Aware Aggregation in Nested Quasi-Newton Optimization

At the global level, the central server aggregates local updates while accounting for variations in feature presence across different clients. When a feature appears in all participating clients, its updates are averaged directly. However, if a feature is present in only a subset of clients, its contribution to the global update is adjusted to prevent underrepresentation. To achieve this, the weight assigned to each feature is determined based on the number of clients containing nonzero values for that feature. Given ω^j as the number of such clients containing nonzero values for feature j, the corresponding aggregation parameter is defined as $a^j = \frac{|C^r|}{\omega^j}$ where $|C^r|$ is the number of participating clients in the r-th round. This adaptive weighting ensures that features appearing in fewer clients are not diminished in the global model, preserving overall feature diversity. To integrate this adjustment, a scaling diagonal matrix is defined as

$$\boldsymbol{A}_r = \text{diag}(\{a^j\}_{j=1,\ldots,d}), \tag{38}$$

where each diagonal entry scales the updates associated with its corresponding feature, and there are d features in total. This mechanism prevents the model from being biased toward features that appear frequently across clients, ensuring a more balanced aggregation. With these adjustments, the global model update is performed using the curvature-aware aggregation mechanism

$$\boldsymbol{w}^{r+1} = \hat{\boldsymbol{w}}^r - \frac{\alpha_g^r}{|C^r|}\boldsymbol{A_r}\sum_{i\in C^r}\frac{n_i}{n^r}\sum_{k=1}^{K}\alpha_l^i \boldsymbol{d_{i,k-1}^r}, \tag{39}$$

where n^r denotes the total number of samples across participating clients in round r.

The proposed approach addresses key challenges in FL by integrating second-order curvature information into both local and global updates, refining optimization steps in a way that adapts to varying data distributions across clients. The final aggregation step ensures that the influence of different features is properly weighted, maintaining diversity in the global model while mitigating the effects of non-IID data.

By leveraging this nested optimization structure, the framework maintains synchronization between global and local updates while reducing communication overhead and computational burden. The periodic deterministic scheduling constraints, which limit client participation per interval, are explicitly incorporated into the optimization process, ensuring that learning remains robust despite constrained communication.

V. Experiments

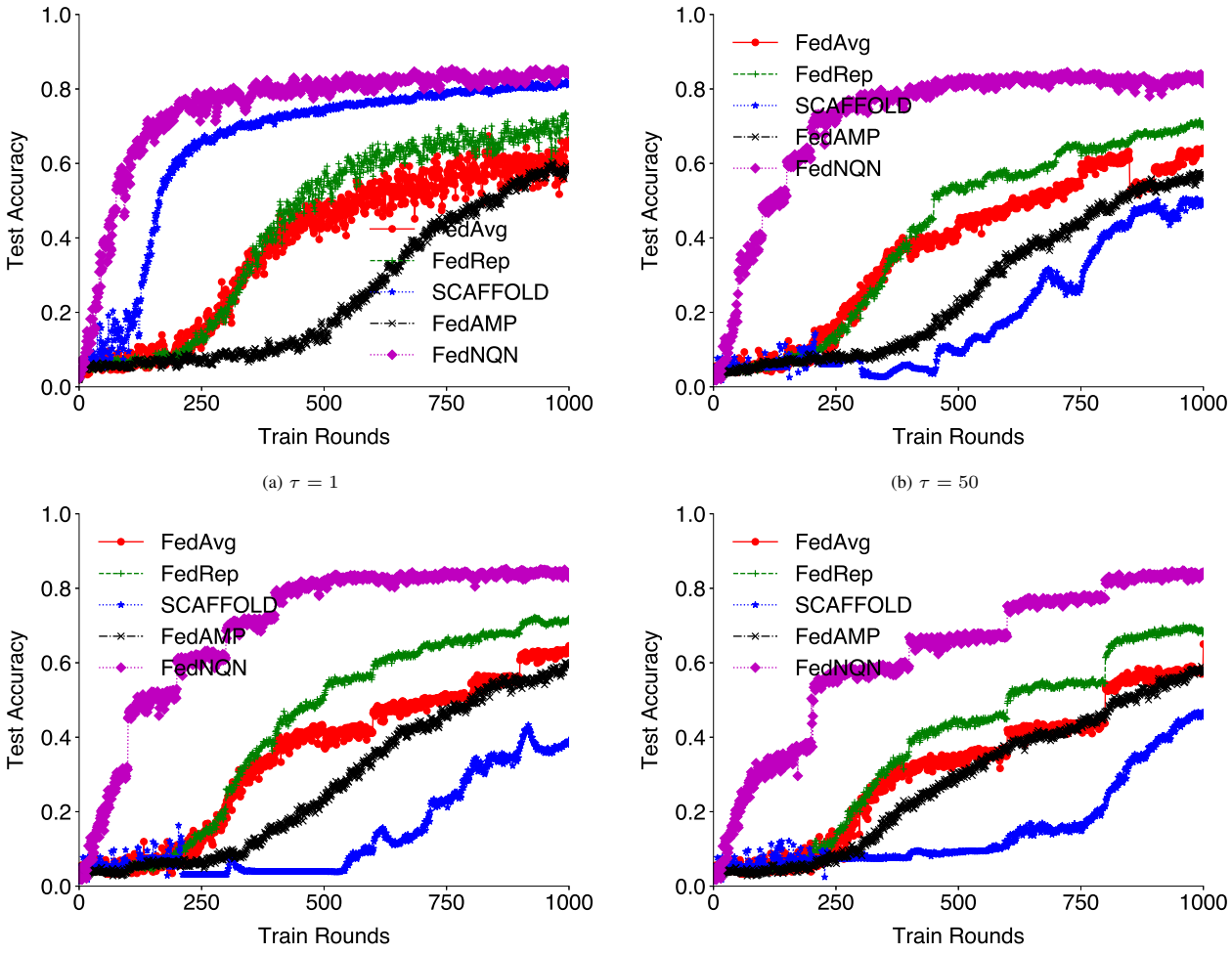
We evaluate the proposed FedNQN designed to integrate global curvature-aware acceleration with client-level variance-controlled quasi-Newton updates across four widely-used federated learning benchmarks, i.e., MNIST, CIFAR-10, FEMNIST, and CIFAR-100. These datasets span diverse input domains, network sizes, and heterogeneity profiles, enabling a comprehensive assessment under both statistical and system-level challenges.

We benchmark FedNQN against a diverse set of state-of-the-art FL algorithms spanning multiple design paradigms. These include first-order methods such as FedAvg [2] and SCAFFOLD [13]; personalized or local adaptation frameworks including FedRep [21], FedAMP [22], FedNTD [28], and FedBABU [29]; and curvature- or communication-aware methods such as FedPAC [24], FedDBE [25], FedALA [26], FedGH [27], CA^{2}FL [31], and ASO-Fed [30]. The following subsections detail experimental settings, model architectures, and performance results under varying levels of data heterogeneity and communication constraints.

All experiments are trained with stochastic gradient descent with momentum 0.10 and mini batch size 10. When $\tau = 1$, each client performs one local epoch per communication round. When $\tau = 50$, each client performs fifty local epochs per round under the same batch size and momentum. The initial local learning rate is set to $\alpha_l = 0.005$ and the initial global learning rate to $\alpha_g = 0.10$ on all datasets in the main comparison. The choice $\alpha_g = 0.10$ enables fast progress of the aggregated model under stochasticity induced by client sampling and data heterogeneity, while $\alpha_l = 0.005$ is conservative enough to mitigate excessive client drift during local updates. Momentum 0.10 provides mild smoothing of noisy gradients, and the mini batch size 10 balances gradient variance with per round computational cost.

A. Evaluation on Federated Benchmarks FEMNIST

We now assess the robustness and scalability of FedNQN on more complex federated benchmarks, beginning with FEMNIST. This dataset introduces significant task complexity by including 62 character classes, covering digits as well as



(a) $\tau = 1$ (b) $\tau = 50$

(a) $\tau = 100$ (b) $\tau = 200$

Fig. 2. Performance Comparison with test accuracy on FEMNIST under increasing periodic deterministic client rescheduling τ.

uppercase and lowercase letters. More importantly, FEMNIST reflects realistic FL conditions by grouping samples based on individual writers. This induces highly non-IID distributions and unequal data volumes across clients, making FEMNIST considerably more challenging than other benchmark datasets. Therefore, it serves as a comprehensive benchmark to evaluate both many-class classification accuracy and robustness under federated heterogeneity.

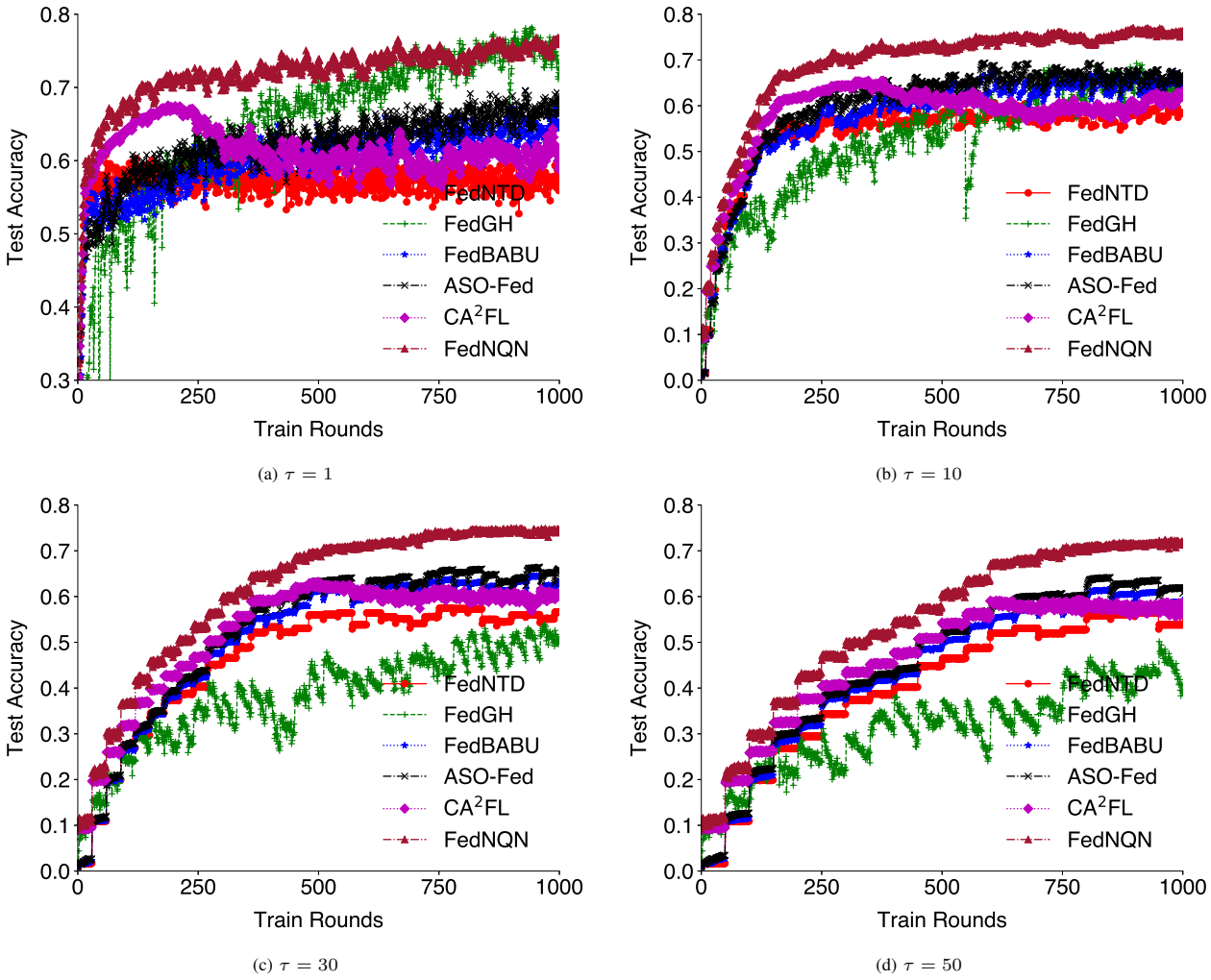
We compare FedNQN against four representative baselines, i.e., FedAvg, a widely used benchmark in FL, FedRep, which adopts a representation-based approach for personalized learning, SCAFFOLD, which mitigates client drift through control variates, and FedAMP, an adaptive personalization method that dynamically adjusts local models based on peer updates.

When client rescheduling is performed at every round, i.e., $\tau = 1$, which is similar to randomized client selection in standard FL, FedNQN achieves clear advantages in both convergence speed and test accuracy. It surpasses 90% test accuracy within 200 communication rounds, while FedAvg, FedRep, and FedAMP require between 250 and 300 rounds to reach similar performance. Among the baselines, SCAFFOLD performs best when there is no PDC constraints, reaching around 80% test accuracy within 1000 rounds. However, its performance deteriorates significantly once PDC constraints are imposed, as shown in the following experiments. These results highlight the early benefits of incorporating curvature information in FedNQN.

As the PDC interval increases to $\tau = 50$ and $\tau = 100$, the performance gap widens notably. With $\tau = 50$, FedNQN achieves over 85% accuracy within 300 rounds and converges near 92%. In contrast, FedAvg and FedRep remain below 65% and 60%, respectively, while FedAMP and SCAFFOLD fail to surpass 50%. At $\tau = 100$, FedNQN remains the only method to maintain steady improvement, reaching 80% accuracy by round 1000. All other methods remain well below 60%, highlighting their vulnerability to PDC constraints. Even under the extreme condition of $\tau = 200$, FedNQN continues to improve and ultimately achieves 70% accuracy, outperforming all baselines by a margin of 15% to 25%.

B. Evaluation on Federated Benchmarks CIFAR-100

We now evaluate the performance of FedNQN on the CIFAR-100 dataset. Fig. 3 presents a comparative analysis of FedNTD, FedGH, FedBABU, ASO-Fed, CA^{2}FL, and the proposed FedNQN, under four different periodic deterministic client rescheduling, i.e., $\tau \in \{1, 10, 30, 50\}$.



(a) $\tau = 1$ (b) $\tau = 10$

(c) $\tau = 30$ (d) $\tau = 50$

Fig. 3. Performance Comparison with test accuracy on CIFAR-100 under increasing periodic deterministic client rescheduling τ.

In the communication case with $\tau = 1$, FedNQN achieves 60% test accuracy in under 100 communication rounds, outperforming the baselines which require significantly more rounds to reach similar performance. Specifically, CA^{2}FL and ASO-Fed require approximately 150 and 200 rounds respectively, while FedNTD and FedGH exceed 250 rounds. As the PDC interval τ increases, the performance advantage of FedNQN becomes more pronounced. At $\tau = 10$, FedNQN surpasses 65% accuracy within 150 rounds and eventually converges around 73%. In contrast, CA^{2}FL remains near 68%, and other baselines such as FedBABU and FedGH fail to exceed 60%.

Under infrequent client rescheduling with $\tau = 30$ and $\tau = 50$, the benefits of FedNQN become even more evident. While first-order methods generally remain below 60% accuracy, FedNQN consistently achieves between 70% and 72%, maintaining a clear margin of 6% to 12% over the closest baselines.

These results from FEMNIST and CIFAR-100 collectively highlight two important advantages of FedNQN. First, by leveraging second-order curvature information, FedNQN consistently reduces the number of communication rounds needed to reach a target accuracy, typically by 30% to 50% on FEMNIST and 30% to 40% on CIFAR-100, compared to strong baselines. This reduction in communication rounds leads to notable savings in bandwidth and energy consumption, especially under PDC settings. Second, while FedNQN introduces a moderate computational overhead of approximately 10% per round due to curvature estimation as shown in Table III, this cost is offset by its accelerated convergence and higher test accuracy. The robustness of FedNQN to stale updates and infrequent client rescheduling, observed across both datasets, makes it particularly well-suited for FL environments with PDC constraints.

C. Computational Cost Analysis

Table III reports the average per-round training time in seconds for all baseline methods and the proposed FedNQN across four federated image classification benchmarks. All experiments were conducted on a Linux workstation equipped with an Intel Core i9-14900KF CPU with 24 physical cores and 32 logical threads, running at up to 5.3 GHz, and 62 GB of RAM.

We use lightweight convolutional architectures tailored to each dataset. For MNIST and CIFAR-10, we adopt a compact

CNN with two convolutional layers of kernel size 5, each followed by a ReLU activation and 2×2 max-pooling. These layers output 32 and 64 channels, respectively. The resulting features are passed to a fully connected layer with 512 hidden units. For FEMNIST, we use a higher-resolution model designed for 128×128 RGB inputs. It consists of two convolutional layers with kernel size 5 and padding 2, each followed by ReLU activation and 2×2 max-pooling. These layers output 32 and 64 channels, producing feature maps of size 32×32, which are flattened into a 65536-dimensional vector and processed by a fully connected layer with 256 units. For CIFAR-100, we reuse the same convolutional architecture as in CIFAR-10, where successive pooling reduces the spatial resolution from 32×32 to 8×8, followed by a fully connected layer with 512 hidden units.

FedAvg completes each training round in 13.96 seconds on MNIST, 18.84 seconds on CIFAR-10, 104.32 seconds on FEMNIST, and 4.51 seconds on CIFAR-100. Other baseline methods incur additional overhead due to the use of control variates, local adaptation layers, or curvature-aware mechanisms. FedNQN requires 26.52 seconds per round on MNIST, 31.56 seconds on CIFAR-10, 136.73 seconds on FEMNIST, and 8.61 seconds on CIFAR-100. This corresponds to approximately $2\times$ the cost of FedAvg on MNIST and CIFAR-10, $1.3\times$ on FEMNIST, and $1.9\times$ on CIFAR-100. Despite the increased per-round cost, these results indicate that the nested quasi-Newton updates in FedNQN scale efficiently with growing data heterogeneity and model complexity, maintaining competitive computational efficiency across all benchmarks.

The cost difference between FEMNIST and CIFAR-100 stems from variations in local sample counts and client population. FEMNIST uses 20 clients, with 10 participating in each round. Each client holds approximately 3100 samples, resulting in around 31000 samples processed per round. In contrast, CIFAR-100 involves 200 clients, with 20 participating per round and about 250 samples per client, yielding roughly 5000 samples per round. These differences contribute to FEMNIST's higher computational burden. Across datasets, lightweight second-order and variance-reduction methods such as SCAFFOLD and FedDBE typically introduce an overhead of 10% to 40% relative to FedAvg. FedNQN incurs a higher overhead of approximately 70% to 90%, yet remains substantially more efficient than heavier methods such as FedPAC, ASO-Fed, and CA^{2}FL, which incur several times the computational cost of FedAvg.

TABLE III
AVERAGE PER-ROUND TRAINING TIME IN SECONDS ACROSS METHODS AND DATASETS

Method	MNIST	CIFAR-10	FEMNIST	CIFAR-100
FedAvg	13.96	18.84	104.32	4.51
SCAFFOLD	15.30	20.07	112.12	5.04
FedRep	18.13	24.66	150.96	5.75
FedAMP	21.53	26.20	136.21	7.60
FedPAC	74.23	44.17	304.21	48.80
FedDBE	15.34	20.43	111.00	4.74
FedALA	20.71	26.86	170.87	6.65
FedGH	18.95	26.21	148.88	5.83
FedNTD	18.88	26.20	150.31	6.09
FedBABU	24.30	31.58	181.92	7.70
ASO-Fed	55.40	33.89	206.08	17.60
CA^{2}FL	52.21	25.94	217.03	10.20
FedNQN	26.52	31.56	136.73	8.61

TABLE IV
PER-ROUND COMMUNICATION COST PER CLIENT.

Method	Upload (Client $\rightarrow$ Server)	Download (Server $\rightarrow$ Client)
FedAvg	8.41 MB	8.41 MB
SCAFFOLD	16.82 MB	16.82 MB
FedRep	0.20 MB	0.20 MB
FedAMP	16.82 MB	16.82 MB
FedNTD	4.00 MB	4.00 MB
FedBABU	8.41 MB	8.41 MB
FedPAC	16.82 MB	8.41 MB
FedDBE	12.00 MB	8.41 MB
FedALA	16.82 MB	16.82 MB
FedGH	16.82 MB	8.41 MB
CA^{2}FL	12.41 MB	8.41 MB
ASO-Fed	8.41 MB	8.41 MB
FedNQN	16.82 MB	16.82 MB

D. Communication Cost Comparison

To provide a complete assessment of the communication efficiency of FedNQN, we analyze the per-round communication cost for all evaluated methods using the same model architecture employed in our experiments. Specifically, we adopt the compact CNN, which consists of two convolutional layers followed by two fully connected layers. This architecture contains approximately $2,202,952$ trainable parameters, i.e., 8.41 MB in float32 precision.

In FedNQN, each client receives both the global model parameters and an anchor gradient from the server at the beginning of each round. During the upload phase, the client transmits both its accumulated local model update and the corresponding local gradient. These elements are essential for enabling server-side curvature estimation and client-side variance reduction. Consequently, the per-round communication cost per client in FedNQN amounts to 16.82 MB in both upload and download directions.

Table IV summarizes the communication cost per round across all compared methods. While first-order baselines such as FedAvg only exchanges model weights, several state-of-the-art approaches introduce significant additional communication overhead. For example, SCAFFOLD and FedAMP transmit control variates, effectively doubling the per-round cost. FedPAC and FedGH also require the exchange of auxiliary correction terms or both gradients and models. In contrast, personalized methods such as FedRep and FedNTD reduce communication by only exchanging partial model components, e.g., the classifier head, though at the cost of generality or performance consistency.

Although FedNQN incurs a higher per-round communication volume than first-order methods, it converges in significantly fewer rounds, leading to a competitive and lower total communication cost to reach a target accuracy. Furthermore, all second-order computations in FedNQN are confined to the

central server, and no matrix-valued curvature data is transmitted, preserving communication simplicity and scalability. Thus, the round-based comparison used in our evaluation fairly reflects both communication efficiency and practical applicability.

E. Client and Server Cost Decomposition

TABLE V
PER-ROUND COMPUTATION COST DECOMPOSITION (SECONDS). (C): CLIENT-SIDE COST, (S): SERVER-SIDE COST, (T): TOTAL COST, (S/T): SERVER SHARE OF TOTAL.

Method	MNIST	CIFAR-10	FEMNIST	CIFAR-100
FedAvg (C)	13.26	17.90	99.10	4.28
FedAvg (T)	13.96	18.84	104.32	4.51
FedNQN (C)	13.26	17.90	99.10	4.28
FedNQN (S)	13.26	13.66	37.63	4.33
FedNQN (T)	26.52	31.56	136.73	8.61
FedNQN (S/T)	50.0%	43.3%	27.5%	50.3%

FedNQN is designed so that each client performs a memoryless local update with the same workload as FedAvg. In each communication round, the client side procedure of FedNQN matches FedAvg with similar computing intensity. The additional curvature aware computation introduced by FedNQN is executed centrally at the server. As a result, the per round client cost is essentially unchanged relative to FedAvg, while the server assumes the extra overhead.

Table V reports an illustrative wall clock split per communication round. FedNQN exhibits a server share between 27.5% and about 50% across datasets. This confirms that the extra work is concentrated on the server rather than on resource constrained clients. Furthermore, our experiments show that FedNQN reaches the target accuracy in roughly 30% to 40% fewer rounds than FedAvg. This implies a modest increase in total system wide computation and communication, but a clear benefit on the client side, i.e., since per round client computation is held at the FedAvg level, the reduction in rounds directly yields a similar reduction in client compute, uplink volume, and downlink volume.

This allocation is aligned with the PDC setting, where client resources and communication opportunities are the primary bottleneck, while the server can absorb heavier updates. Fewer rounds also reduce synchronization events, mitigate straggler effects, and shorten the wall clock time to reach a given accuracy, even if the aggregate system cost increases moderately due to stronger server side processing.

F. Unified Baselines Comparison in Cross-Dataset Robustness

To ensure that performance gains are not driven by inconsistent baselines or dataset-specific choices, we adopt a unified evaluation protocol for all methods on the main benchmarks. All results in Table VI are reported at the 200-th communication round under a common setting. For each dataset, all algorithms use the same model architecture and the same client data partitions. Unless otherwise specified, the batch size is fixed to 10, the momentum coefficient to 0.10, and each method runs one local epoch per round when $\tau = 1$ and fifty local epochs per round when $\tau = 50$. The initial local learning rate is set to $\alpha_l = 0.005$ and the initial global learning rate to $\alpha_g = 0.10$. Methods that include internal step-size adaptation or correction mechanisms keep their standard schedules without additional retuning. This harmonized protocol ensures that the comparisons isolate algorithmic behavior rather than favoring specific methods through tailored hyperparameters or training budgets.

TABLE VI
TEST ACCURACY AT THE 200-TH COMMUNICATION ROUND UNDER THE UNIFIED BASELINES.

Method	FEMNIST $\tau = 1$	FEMNIST $\tau = 50$	CIFAR-100 $\tau = 1$	CIFAR-100 $\tau = 50$
FedAvg	0.1326	0.1074	0.2141	0.2230
SCAFFOLD	0.6045	0.0769	0.2025	0.1772
FedRep	0.0818	0.0935	0.1871	0.1948
FedAMP	0.0571	0.0813	0.5988	0.1794
FedGH	0.6247	0.5685	0.5646	0.2643
FedNTD	0.6404	0.1883	0.5433	0.2944
FedBABU	0.6085	0.1870	0.5459	0.3053
ASO-Fed	0.3591	0.5573	0.5713	0.3181
CA^{2}FL	0.2346	0.1002	0.6733	0.3748
FedNQN	0.7393	0.6975	0.7058	0.4155

Under this unified setup, FedNQN consistently achieves the best performance across both datasets and both communication regimes. On FEMNIST with $\tau = 1$, FedNQN attains a test accuracy of 0.7393, which exceeds the strongest baseline FedNTD at 0.6404, corresponding to an absolute gain of 0.0989 and a relative improvement of about 15.4%. For FEMNIST with $\tau = 50$, FedNQN reaches 0.6975 compared to FedGH at 0.5685, yielding an absolute gain of 0.1290 and a relative improvement of about 22.7%. On CIFAR-100 with $\tau = 1$, FedNQN obtains 0.7058 against CA^{2}FL at 0.6733, with an absolute gain of 0.0325 and a relative improvement of about 4.8%. With $\tau = 50$ on CIFAR-100, FedNQN achieves 0.4155 against CA^{2}FL at 0.3748, with an absolute gain of 0.0407 and a relative improvement of about 10.9%.

The cross-dataset trend is consistent. Baselines that are sensitive to client drift, such as FedAvg and several personalization-oriented methods, degrade substantially when τ increases; for instance, FedAvg drops to 0.1074 on FEMNIST and 0.2230 on CIFAR-100 at $\tau = 50$. Curvature-aware or variance-reduction methods such as FedGH, CA^{2}FL, and ASO-Fed exhibit improved robustness yet are still outperformed by FedNQN in every reported setting. The stability of FedNQN under both small and large τ, together with its superiority on both FEMNIST and CIFAR-100 under the unified protocol, demonstrates that its advantages are not tied to a particular dataset or communication regime and instead arise from its ability to remain effective under increased local computation and heterogeneous client distributions.

VI. CONCLUSION

This work proposed a Nested Quasi-Newton optimization framework for FL under PDC constraints. The framework introduces a hierarchical second-order optimization strategy

that enhances both global and local learning dynamics. At the server side, curvature-guided global acceleration leverages inverse Hessian approximations to refine global model updates, while on the client side, lightweight quasi-Newton updates incorporate local curvature information to approximate the global descent direction. This nested design ensures that local progress remains aligned with the global objective, effectively mitigating the challenges of delayed client rescheduling induced by PDC constraints. Extensive experiments across diverse benchmarks demonstrate that FedNQN achieves faster convergence and higher accuracy than the state-of-the-art baselines, while maintaining favorable computational efficiency. These results highlight the potential of hierarchical second-order optimization in advancing robust and efficient FL, particularly in structured and resource-constrained communication settings.

APPENDIX A

This appendix provides a unified convergence analysis of the proposed FedNQN framework under PDC. We analyze the local curvature-aware updates and the global curvature-guided optimization steps separately. It is shown that the inexact quasi-Newton local updates exhibit geometric loss reduction and align with the global optimization trajectory.

A. Local Training Convergence under Periodic Client Participation

We first analyze the local training dynamics under PDC. In each round, clients perform multiple local optimization steps using curvature-aware quasi-Newton updates before communicating with the server. Under standard smoothness and strong convexity assumptions, the local model contracts geometrically toward a curvature-adjusted global reference point.

1) Local Variance Reduction: To align the local update direction with the global descent direction under PDC constraints, we define the following corrected local objective for client i at round r

$$\tilde{f}_i(\boldsymbol{w}) = f_i(\boldsymbol{w}) - \left[\nabla f_i(\boldsymbol{w}^r) - \nabla f(\boldsymbol{w}^r)\right]^\top \boldsymbol{w}, \tag{40}$$

where $\boldsymbol{w}^r$ is the global model at the start of round r, and $\nabla f(\boldsymbol{w}^r)$ is the anchor gradient broadcast by the server. This construction ensures gradient consistency at the anchor point

$$\nabla \tilde{f}_i(\boldsymbol{w}^r) = \nabla f(\boldsymbol{w}^r). \tag{41}$$

In practice, clients use stochastic gradients. A variance-reduced estimator for $\nabla \tilde{f}_i(\boldsymbol{w})$ is given by

$$\boldsymbol{g}_i(\boldsymbol{w}) - \boldsymbol{g}_i(\boldsymbol{w}^r) + \boldsymbol{g}(\boldsymbol{w}^r), \tag{42}$$

where $\boldsymbol{g}_i(\cdot)$ and $\boldsymbol{g}(\cdot)$ denote local and global stochastic gradients, respectively. This estimator satisfies the unbiasedness property

$$\mathbb{E}\left[\boldsymbol{g}_i(\boldsymbol{w}) - \boldsymbol{g}_i(\boldsymbol{w}^r) + \boldsymbol{g}(\boldsymbol{w}^r)\right] = \nabla f(\boldsymbol{w}). \tag{43}$$

Since $\boldsymbol{g}(\boldsymbol{w}^r)$ is fixed during local training, the variance of the estimator reduces to that of the local difference

$$\mathrm{Var}\left(\boldsymbol{g}_i(\boldsymbol{w}) - \boldsymbol{g}_i(\boldsymbol{w}^r) + \boldsymbol{g}(\boldsymbol{w}^r)\right) \le \mathbb{E}\left[\|\boldsymbol{g}_i(\boldsymbol{w}) - \boldsymbol{g}_i(\boldsymbol{w}^r)\|^2\right]. \tag{44}$$

Using the β_i-smoothness of f_i, we further have

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{g}_i(\boldsymbol{w}) - \boldsymbol{g}_i(\boldsymbol{w}^r)\|^2\right] &\le \|\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{w}^r)\|^2 \\ &\le \beta_i^2 \|\boldsymbol{w} - \boldsymbol{w}^r\|^2,\end{aligned} \tag{45}$$

which confirms that the variance of the corrected estimator is tightly controlled by the distance between the local model $\boldsymbol{w}$ and the anchor point $\boldsymbol{w}^r$.

2) One-Step Progress Under Smoothness and Strong Convexity: Assume that the local objective f_i is μ_i-strongly convex and β_i-smooth. Let $\hat{\boldsymbol{w}}_{i,k}^r$ denote the local model of client i at local step k during round r, initialized from the global model $\boldsymbol{w}^r$, i.e., $\hat{\boldsymbol{w}}_{i,0}^r = \boldsymbol{w}^r$. The local update rule with step size α_l is given by

$$\hat{\boldsymbol{w}}_{i,k}^r = \hat{\boldsymbol{w}}_{i,k-1}^r - \alpha_l \left(\boldsymbol{g}_i(\hat{\boldsymbol{w}}_{i,k-1}^r) - \boldsymbol{g}_i(\boldsymbol{w}^r) + \boldsymbol{g}(\boldsymbol{w}^r)\right). \tag{46}$$

Let $\boldsymbol{w}^*$ be the global optimum. The squared distance to $\boldsymbol{w}^*$ after one local update satisfies

$$\begin{aligned}&\mathbb{E}_{k-1}\left[\|\hat{\boldsymbol{w}}_{i,k}^r - \boldsymbol{w}^*\|^2\right] = \|\hat{\boldsymbol{w}}_{i,k-1}^r - \boldsymbol{w}^*\|^2 \\ &- 2\alpha_l \left(\hat{\boldsymbol{w}}_{i,k-1}^r - \boldsymbol{w}^*\right)^\top \mathbb{E}_{k-1}\left[\boldsymbol{g}_i(\hat{\boldsymbol{w}}_{i,k-1}^r) - \boldsymbol{g}_i(\boldsymbol{w}^r) + \boldsymbol{g}(\boldsymbol{w}^r)\right] \\ &+ \alpha_l^2 \mathbb{E}_{k-1}\left[\left\|\boldsymbol{g}_i(\hat{\boldsymbol{w}}_{i,k-1}^r) - \boldsymbol{g}_i(\boldsymbol{w}^r) + \boldsymbol{g}(\boldsymbol{w}^r)\right\|^2\right],\end{aligned} \tag{47}$$

where $\mathbb{E}_{k-1}[\cdot]$ denotes conditional expectation given prior iterates. By the unbiasedness of the gradient estimator

$$\mathbb{E}_{k-1}\left[\boldsymbol{g}_i(\hat{\boldsymbol{w}}_{i,k-1}^r) - \boldsymbol{g}_i(\boldsymbol{w}^r) + \boldsymbol{g}(\boldsymbol{w}^r)\right] = \nabla f_i(\hat{\boldsymbol{w}}_{i,k-1}^r), \tag{48}$$

and the strong convexity of f_i implies

$$\left(\hat{\boldsymbol{w}}_{i,k-1}^r - \boldsymbol{w}^*\right)^\top \nabla f_i(\hat{\boldsymbol{w}}_{i,k-1}^r) \ge \mu_i \|\hat{\boldsymbol{w}}_{i,k-1}^r - \boldsymbol{w}^*\|^2. \tag{49}$$

For the variance term, we apply a norm inequality

$$\begin{aligned}&\mathbb{E}_{k-1}\left[\left\|\boldsymbol{g}_i(\hat{\boldsymbol{w}}_{i,k-1}^r) - \boldsymbol{g}_i(\boldsymbol{w}^r) + \boldsymbol{g}(\boldsymbol{w}^r)\right\|^2\right] \\ &\le 2\mathbb{E}_{k-1}\left[\|\boldsymbol{g}_i(\hat{\boldsymbol{w}}_{i,k-1}^r) - \boldsymbol{g}_i(\boldsymbol{w}^r)\|^2\right] + 2\|\boldsymbol{g}(\boldsymbol{w}^r)\|^2 \\ &\le 2\beta_i^2 \|\hat{\boldsymbol{w}}_{i,k-1}^r - \boldsymbol{w}^r\|^2 + 2\|\nabla f(\boldsymbol{w}^r)\|^2.\end{aligned} \tag{50}$$

To relate $\|\boldsymbol{w}^r - \boldsymbol{w}^*\|^2$ to $\|\hat{\boldsymbol{w}}_{i,k-1}^r - \boldsymbol{w}^*\|^2$, assume a constant $e \ge 1$ such that

$$\|\boldsymbol{w}^r - \boldsymbol{w}^*\|^2 \le e\|\hat{\boldsymbol{w}}_{i,k-1}^r - \boldsymbol{w}^*\|^2. \tag{51}$$

Substituting all bounds yields the contraction inequality

$$\begin{aligned}&\mathbb{E}_{k-1}\left[\|\hat{\boldsymbol{w}}_{i,k}^r - \boldsymbol{w}^*\|^2\right] \\ &\le \left(1 - 2\alpha_l\mu_i + 5\alpha_l^2\beta_i^2 e\right)\|\hat{\boldsymbol{w}}_{i,k-1}^r - \boldsymbol{w}^*\|^2.\end{aligned} \tag{52}$$

Choosing α_l such that $1 - 2\alpha_l\mu_i + 5\alpha_l^2\beta_i^2 e < 1$ guarantees geometric improvement at each local step.

3) Geometric Decay and Local Convergence: By recursively applying the one-step contraction derived earlier, we characterize the convergence behavior of local updates over K steps. To ensure contraction, we impose the following step size condition $\alpha_l \le \frac{\mu_i}{5\beta_i^2 e}$, which guarantees that the geometric contraction factor is bounded by

$$1 - 2\alpha_l\mu_i + 5\alpha_l^2\beta_i^2 e \le \exp\left(-\frac{\mu_i^2}{5\beta_i^2 e}\right) < 1. \tag{53}$$

Consequently, the local model error decays geometrically with K local steps

$$\mathbb{E}\left[\|\hat{\boldsymbol{w}}_{i,K}^r - \boldsymbol{w}^*\|^2\right] \leq \exp\left(-\frac{\mu_i^2 K}{5\beta_i^2 e}\right)\|\boldsymbol{w}^r - \boldsymbol{w}^*\|^2, \quad (54)$$

which confirms that the variance-reduced local quasi-Newton steps achieve linear convergence in expectation. The decay rate is explicitly governed by the local strong convexity constant μ_i, smoothness constant β_i, the number of steps K, and the overshoot parameter e. To reduce the squared error by at least a factor of $1/e$, it suffices to choose the number of local steps as $K \geq \frac{5\beta_i^2 e}{\mu_i^2}$.

Moreover, since the global iterate $\boldsymbol{w}^r$ is itself improved through quasi-Newton updates at the central server, its distance to the global optimum $\boldsymbol{w}^*$ shrinks over rounds. As a result, local iterates remain increasingly close to the true solution, promoting stability and enabling communication-efficient training under the PDC regime.

B. Global Convergence of Aggregated Quasi-Newton Updates

We now analyze the global convergence behavior of FedNQN by focusing on the server-side updates. Building on the local convergence guarantees established in Appendix A-A, we show that the aggregated quasi-Newton step, formed from anchor gradients and global curvature estimates, leads to consistent descent toward the global optimum.

At each communication round r, the server performs the following global update

$$\boldsymbol{w}^{r+1} = \boldsymbol{w}^r - \alpha \boldsymbol{S}_r \boldsymbol{g}(\boldsymbol{w}^r), \quad (55)$$

where $\boldsymbol{g}(\boldsymbol{w}^r)$ is the anchor gradient and $\boldsymbol{S}_r$ is the inverse Hessian approximation maintained at the server.

We assume the global objective $f(\boldsymbol{w})$ is μ-strongly convex and L-smooth. The anchor gradient is unbiased with bounded variance

$$\mathbb{E}[\boldsymbol{g}(\boldsymbol{w}^r)] = \nabla f(\boldsymbol{w}^r), \quad \mathbb{E}\left[\|\boldsymbol{g}(\boldsymbol{w}^r) - \nabla f(\boldsymbol{w}^r)\|^2\right] \leq \sigma^2. \quad (56)$$

Moreover, we assume the curvature approximations are uniformly bounded

$$m\boldsymbol{I} \preceq \boldsymbol{S}_r \preceq M\boldsymbol{I}, \quad \text{for all } r, \quad (57)$$

for constants $0 < m \leq M$. By the L-smoothness of f, we apply the standard descent lemma

$$f(\boldsymbol{w}^{r+1}) \leq f(\boldsymbol{w}^r) + \nabla f(\boldsymbol{w}^r)^\top(\boldsymbol{w}^{r+1} - \boldsymbol{w}^r) + \frac{L}{2}\|\boldsymbol{w}^{r+1} - \boldsymbol{w}^r\|^2. \quad (58)$$

Substituting the update rule yields

$$f(\boldsymbol{w}^{r+1}) \leq f(\boldsymbol{w}^r) - \alpha\nabla f(\boldsymbol{w}^r)^\top \boldsymbol{S}_r \boldsymbol{g}(\boldsymbol{w}^r) + \frac{\alpha^2 L}{2}\|\boldsymbol{S}_r \boldsymbol{g}(\boldsymbol{w}^r)\|^2. \quad (59)$$

Taking expectation over the randomness of $\boldsymbol{g}(\boldsymbol{w}^r)$, we obtain

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{w}^{r+1})] \\
&\leq f(\boldsymbol{w}^r) - \alpha\nabla f(\boldsymbol{w}^r)^\top \boldsymbol{S}_r \mathbb{E}[\boldsymbol{g}(\boldsymbol{w}^r)] + \frac{\alpha^2 L}{2}\mathbb{E}[\|\boldsymbol{S}_r \boldsymbol{g}(\boldsymbol{w}^r)\|^2] \\
&= f(\boldsymbol{w}^r) - \alpha\nabla f(\boldsymbol{w}^r)^\top \boldsymbol{S}_r \nabla f(\boldsymbol{w}^r) + \frac{\alpha^2 L}{2}\mathbb{E}[\|\boldsymbol{S}_r \boldsymbol{g}(\boldsymbol{w}^r)\|^2],
\end{aligned} \quad (60)$$

where we used the unbiasedness of the anchor gradient.

To bound the second moment term, we decompose the squared norm as

$$\begin{aligned}
&\mathbb{E}[\|\boldsymbol{S}_r \boldsymbol{g}(\boldsymbol{w}^r)\|^2] = \mathbb{E}[\|\boldsymbol{S}_r(\boldsymbol{g}(\boldsymbol{w}^r) - \nabla f(\boldsymbol{w}^r)) + \boldsymbol{S}_r \nabla f(\boldsymbol{w}^r)\|^2] \\
&\leq 2\mathbb{E}[\|\boldsymbol{S}_r(\boldsymbol{g}(\boldsymbol{w}^r) - \nabla f(\boldsymbol{w}^r))\|^2] + 2\|\boldsymbol{S}_r \nabla f(\boldsymbol{w}^r)\|^2 \\
&\leq 2M^2\sigma^2 + 2M\nabla f(\boldsymbol{w}^r)^\top \boldsymbol{S}_r \nabla f(\boldsymbol{w}^r),
\end{aligned} \quad (61)$$

where the last step uses the spectral bound $\|\boldsymbol{S}_r\| \leq M$ and Jensen's inequality. Substituting this into the earlier bound, we obtain

$$\begin{aligned}
\mathbb{E}[f(\boldsymbol{w}^{r+1})] &\leq f(\boldsymbol{w}^r) - \alpha\nabla f(\boldsymbol{w}^r)^\top \boldsymbol{S}_r \nabla f(\boldsymbol{w}^r) \\
&\quad + \alpha^2 LM\nabla f(\boldsymbol{w}^r)^\top \boldsymbol{S}_r \nabla f(\boldsymbol{w}^r) + \alpha^2 LM^2\sigma^2 \\
&= f(\boldsymbol{w}^r) - \alpha(1 - \alpha LM)\nabla f(\boldsymbol{w}^r)^\top \boldsymbol{S}_r \nabla f(\boldsymbol{w}^r) + \alpha^2 LM^2\sigma^2.
\end{aligned} \quad (62)$$

Using the lower spectral bound $\boldsymbol{S}_r \succeq m\boldsymbol{I}$, we have

$$\nabla f(\boldsymbol{w}^r)^\top \boldsymbol{S}_r \nabla f(\boldsymbol{w}^r) \geq m\|\nabla f(\boldsymbol{w}^r)\|^2, \quad (63)$$

which yields

$$\mathbb{E}[f(\boldsymbol{w}^{r+1})] \leq f(\boldsymbol{w}^r) - \alpha m(1 - \alpha LM)\|\nabla f(\boldsymbol{w}^r)\|^2 + \alpha^2 LM^2\sigma^2. \quad (64)$$

By the strong convexity of f, the gradient norm satisfies

$$\|\nabla f(\boldsymbol{w}^r)\|^2 \geq 2\mu(f(\boldsymbol{w}^r) - f^*). \quad (65)$$

Substituting this gives

$$\mathbb{E}[f(\boldsymbol{w}^{r+1}) - f^*] \leq (1 - 2\alpha\mu m(1 - \alpha LM))(f(\boldsymbol{w}^r) - f^*) + \alpha^2 LM^2\sigma^2. \quad (66)$$

Choosing the step size $\alpha \leq \min\left\{\frac{1}{L}, \frac{1}{M\mu}\right\}$ ensures that $1 - \alpha LM \geq \frac{1}{2}$ and simplifies the contraction factor

$$\mathbb{E}[f(\boldsymbol{w}^{r+1}) - f^*] \leq (1 - \alpha\mu m)(f(\boldsymbol{w}^r) - f^*) + \alpha^2 LM^2\sigma^2. \quad (67)$$

Unrolling the recursion over r iterations and summing the geometric series for the residual variance yields

$$\mathbb{E}[f(\boldsymbol{w}^r) - f^*] \leq (1 - \alpha\mu m)^r(f(\boldsymbol{w}^0) - f^*) + \frac{\alpha M\sigma^2}{2\mu}, \quad (68)$$

which completes the convergence proof.

The global update in FedNQN combines curvature-aware descent with variance-reduced anchor gradients. Thanks to the spectral bounds on the inverse Hessian approximation and the controlled variance of the aggregated gradients, each global step aligns well with the true curvature of the objective. As the model iterates approach the optimum, the quasi-Newton matrix $\boldsymbol{S}_r$ increasingly approximates the true inverse Hessian, allowing the algorithm to transition from linear to superlinear convergence, thus accelerating the global optimization process even under PDC constraints.

REFERENCES

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

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

[3] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[4] L. Zhao, L. Cai, and W.-S. Lu, "Federated learning for data trading portfolio allocation with autonomous economic agents," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[5] S. M. Pesenti, P. Millossovich, and A. Tsanakas, "Differential quantile-based sensitivity in discontinuous models," *European Journal of Operational Research*, 2024.

[6] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[7] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *Proceedings of machine learning and systems*, vol. 1, pp. 374–388, 2019.

[8] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[9] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[10] Y. Wang, Z. Su, Y. Pan, T. H. Luan, R. Li, and S. Yu, "Social-aware clustered federated learning with customized privacy preservation," *IEEE/ACM Transactions on Networking*, 2024.

[11] Z.-L. Chang, S. Hosseinalipour, M. Chiang, and C. G. Brinton, "Asynchronous multi-model dynamic federated learning over wireless networks: Theory, modeling, and optimization," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[12] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[13] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

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

[15] E. Gorbunov, K. P. Burlachenko, Z. Li, and P. Richtárik, "Marina: Faster non-convex distributed learning with compression," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3788–3798.

[16] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," *arXiv preprint arXiv:2101.11203*, 2021.

[17] L. Zhao, L. Cai, and W.-S. Lu, "Transform-domain federated learning for edge-enabled iot intelligence," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6205–6220, 2022.

[18] A. M. Elbir, S. Coleri, A. K. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "A hybrid architecture for federated and centralized learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 3, pp. 1529–1542, 2022.

[19] L. Zhao, L. Cai, and W.-S. Lu, "Accelerating federated learning for edge intelligence using conjugated central acceleration with inexact global line search," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[20] X. Gu, K. Huang, J. Zhang, and L. Huang, "Fast federated learning in the presence of arbitrary device unavailability," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[21] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International conference on machine learning*. PMLR, 2021, pp. 2089–2099.

[22] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 9, 2021, pp. 7865–7873.

[23] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[24] J. Xu, X. Tong, and S.-L. Huang, "Personalized federated learning with feature alignment and classifier collaboration," *arXiv preprint arXiv:2306.11867*, 2023.

[25] J. Zhang, Y. Hua, J. Cao, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Eliminating domain bias for federated learning in representation space," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[26] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Fedala: Adaptive local aggregation for personalized federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11 237–11 244.

[27] L. Yi, G. Wang, X. Liu, Z. Shi, and H. Yu, "Fedgh: Heterogeneous federated learning with generalized global header," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 8686–8696.

[28] G. Lee, M. Jeong, Y. Shin, S. Bae, and S.-Y. Yun, "Preservation of the global knowledge by not-true distillation in federated learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 461–38 474, 2022.

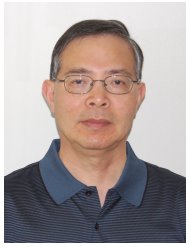
[29] J. Oh, S. Kim, and S.-Y. Yun, "Fedbabu: Towards enhanced representation for federated image classification," *arXiv preprint arXiv:2106.06042*, 2021.

[30] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-iid data," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 15–24.

[31] Y. Wang, Y. Cao, J. Wu, R. Chen, and J. Chen, "Tackling the data heterogeneity in asynchronous federated learning with cached update calibration," in *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023.

[32] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*, 2nd ed. Springer, 2021.

Lei Zhao (S'17-M'23) received the B.S. and M.A.Sc. degrees in computer science and technology from Xidian University, Xi'an, China, in 2015 and 2018, respectively, and earned his Ph.D. in Electrical and Computer Engineering from the University of Victoria in 2023. He is currently a Postdoctoral Research Associate at Yale School of Management. His research focuses on machine learning and optimization in finance.



Wu-Sheng Lu (F'99-LF'12) received the B.Sc. degree in Mathematics from Fudan University, Shanghai, China, in 1964, the M.S. degree in electrical engineering, and the Ph.D. degree in control science from the University of Minnesota, Minneapolis, USA, in 1983 and 1984, respectively. Since 1987, he has been with the University of Victoria, Victoria, B.C., Canada, and is now Professor Emeritus. He is the co-author with A. Antoniou of Two-Dimensional Digital Filters (Marcel Dekker, 1992) and Practical Optimization: Algorithms and Engineering Applications (2nd ed., Springer, 2021), and with E. K. P. Chong and S. H. Zak of An Introduction to Optimization (5th ed., Wiley, 2023).

Lin Cai (S'00-M'06-SM'10-F'20) has been with the Department of Electrical & Computer Engineering at the University of Victoria since 2005, and she is currently a Professor. She is an NSERC E.W.R. Steacie Memorial Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and an IEEE Fellow. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and the Internet of Things. She has been elected to serve the board of the IEEE Vehicular Technology Society, 2019 - 2024, and as its VP in Mobile Radio. She has been a Board Member of IEEE Women in Engineering (2022-24) and IEEE Communications Society (2024-2026). She has served as an Associate Editor-in-Chief for IEEE Transactions on Vehicular Technology, and as a Distinguished Lecturer of the IEEE VTS Society and the IEEE Communications Society.