Service Delay Minimization for Federated Learning Over Mobile Devices

Rui Chen, *Member, IEEE*, Dian Shi, *Student Member, IEEE*, Xiaoqi Qin, *Member, IEEE*, Dongjie Liu, Miao Pan, *Senior Member, IEEE*, and Shuguang Cui, *Fellow, IEEE*

Abstract— Federated learning (FL) over mobile devices has fostered numerous intriguing applications/services, many of which are delay-sensitive. In this paper, we propose a service delay efficient FL (SDEFL) scheme over mobile devices. Unlike traditional communication efficient FL, which regards wireless communications as the bottleneck, we find that under many situations, the local computing delay is comparable to the communication delay during the FL training process, given the development of high-speed wireless transmission techniques. Thus, the service delay in FL should be computing delay $+$ communication delay over training rounds. To minimize the service delay of FL, simply reducing local computing/communication delay independently is not enough. The delay trade-off between local computing and wireless communications must be considered. Besides, we empirically study the impacts of local computing control and compression strategies (i.e., the number of local updates, weight quantization, and gradient quantization) on computing, communication and service delays. Based on those trade-off observation and empirical studies, we develop an optimization scheme to minimize the service delay of FL over heterogeneous devices. We establish testbeds and conduct extensive emulations/experiments to verify our theoretical analysis. The results show that SDEFL reduces notable service delay with a small accuracy drop compared to peer designs.

Index Terms— Federated learning over mobile devices, weight quantization, gradient quantization, device heterogeneity.

I. INTRODUCTION

WITH ever increasing computing and communication capabilities, beyond learning inference, mobile devices (Google Pixel 4a with Adreno GPU, Mac with M1 chip and Wi-Fi 5, Nvidia Jetson devices with Wi-Fi 5, etc.) are promising to execute on-device training of some sophisticated deep learning (DL) models [1]. Meanwhile, federated learning (FL) has evolved as a prospective distributed learning system across numerous mobile devices. FL enables mobile devices to learn DL models locally and then exchange the model updates via FL aggregation without data privacy leakage [2]. FL over mobile devices can provide a variety of services, including Google's next word prediction [3], cardiac event prediction [4], mobile object detection [5], etc. Notably, many of those FL inspired services over mobile devices are delay-sensitive. FL service delays on mobile devices are primarily caused by two factors: local computing delays and wireless communication delays during training rounds. Because state-of-the-art DL models are typically overparameterized, it takes a long time to locally train those DL models on computational resource restricted mobile devices. In term of wireless communication delay, the size of gradients to exchange is relatively large (e.g., ResNet50 with ImageNet dataset has 98M parameters). This places a significant demand on wireless transmission and may result in huge aggregated communication delays during FL training. Such a service delay issue is more and more severe as the contemporary DL models become deeper and larger. As a result, it is vital to investigate how to develop a service delay efficient FL over mobile devices.

One of the most prominent ways for reducing computational complexity for on-device training is weight quantization [6], [7], [8]. Weight quantization, by shrinking the precision of model parameters, may effectively reduce the storage sizes and the computing delay of on-device training. Large-scale deep neural networks (DNNs) (e.g., ResNet and MobileNet) might, for example, be quantized to provide a foundation for fast on-device training/inference, as described in [7] and [6]. Those approaches, on the other hand, strive to increase learning accuracy and primarily focus on centralized learning problem with a single device. The weight quantization's impacts on computing delay or service delay in the FL with multiple

devices have not been fully explored yet. Since FL encompasses massive devices with heterogeneous computing capabilities and heterogeneous local datasets, it is worth further investigating the impacts of weight quantization on the service delay and FL performance.

For wireless transmission delay in FL, most existing works in wireless communities have mainly conducted the radio resource allocation under the FL convergence constraints [9], [10], [11], [12], while neglecting how to inherently reduce the communication payload and delay from the DL algorithms themselves. By contrast, in machine learning communities, assuming wireless transmissions are the bottleneck, there have been two mainstream strategies proposed to improve communication efficiency of FL: (i) gradient quantization [13], [14], which compresses the size of gradients to transmit and thus reduces communication bandwidth consumption per round in FL at the cost of precision; (ii) local computing control strategy, e.g., local stochastic gradient descent (SGD) [2]), which allows participating mobile devices to perform multiple training iterations locally before updating to the FL edge server, thereby avoiding communication after every local iteration and reducing the update frequencies. In parallel with those benefits, an aggressive gradient quantization strategy may result in significant distortion on the gradients, and thus require more communication rounds for compensation to achieve FL convergence [13]. Besides, increasing the number of local training iterations may lead to large discrepancies between local models, which consequently incurs an inferior error-convergence in FL. Intensive local training also introduces large computing delay in FL. Therefore, existing communication efficient FL designs [2], [13], [14], which prefer aggressive compression strategy/intensive local training to reduce the communication bandwidth consumption per round/the total number of communication rounds in FL, may not be optimal w.r.t. the service delay of FL over mobile devices. First, most of them assign the identical quantization strategy to all the participants, and don't consider the involved huge computing delay and heterogeneous computing capabilities of those devices. Such designs cannot mitigate the FL straggler problem and slow down the model aggregation. Second, they ignore the recent advance of high-speed wireless transmissions.

By embracing the high-speed wireless transmission era (e.g., 5G, Wi-Fi 5, Wi-Fi 6 or 6G), the wireless communication bottleneck can be relieved for FL over mobile devices. Furthermore, we observe that local computing delay for on-device training is comparable to wireless communication delay, if the transmission rate is sufficiently high. For example, transmitting a ResNet20 model via 100 Mbps Wi-Fi 5 links takes 90ms, which is comparable to the time consumption of executing one-step local training on a mobile device with a modest GPU, e.g., 86ms for Jetson Xavier with 1.3TFLOPs on Cifar10 dataset. Thus, in order to optimize the service delay of FL over mobile devices, simply minimizing local computing/wireless transmission delay independently is not good enough, and the delay trade-off between "working" (i.e., local computing) and "talking" (i.e., wireless communications) has to be considered.

Motivated by the aforementioned observations and challenges, we develop in this paper a **S**ervice **D**elay **E**fficient FL (SDEFL) over mobile devices, with the design goal of minimizing the FL's service delay (local computing delay + wireless communication delay during FL training process) without sacrificing the learning performance. The proposed SDEFL jointly considers stochastic weight quantization and local computing control for efficient on-device local training, and gradient quantization for efficient local model update communications. Briefly, we empirically study the impacts of different quantization and control strategies on the learning performance and the service delay in FL. Based on the empirical observations and theoretical convergence analysis, we formulate the service delay minimization to determine the optimal quantization strategies and iteration number of local training with considering heterogeneous computing and communication conditions of mobile devices. We further establish testbed and conduct experiments to evaluate the proposed scheme and verify our findings. Our salient contributions are summarized as follows.

- Inspired by the observation that computing and communication delays are comparable under many situations, we empirically study how local computing control (i.e., deciding the number of local training iterations), weight quantization strategy, and gradient quantization strategy affect the service delay of FL over mobile devices, respectively. Insight from those studies is that the local computing control and gradient quantization play very important roles to determine the service delay.

- Based on our empirical observation, we develop an SDEFL scheme over mobile devices by jointly selecting the number of local training iterations, gradient quantization levels, and weight quantization levels. Here, mobile devices are allowed to quantize the model weights locally to speed up the computing delay and then quantize the local-update gradients to reduce the communication delay. Besides, we provide the theoretical analysis of the convergence upper bound of the proposed SDEFL scheme with flexible quantization strategies.

- According to the derived theoretical convergence rate, we formulate the SDEFL problem as an integer nonlinear programming, where the number of local training iterations, weight quantization level, and gradient quantization level are decision variables. Geometric programming is exploited to develop feasible solutions.

- We set up testbeds, and conduct extensive simulations and experiments to verify the effectiveness of our proposed SDEFL scheme under various learning models, different data distributions across devices, and multiple wireless environmental settings.

The rest of this paper is organized as follows. The related work is provided in Section II In Section III, the system model and problem setting are introduced. The empirical studies are discussed in Sections IV. The SDEFL's formulation, convergence analysis and feasible solutions are illustrated in Sections V. In Section VI, the experimental and simulation results are presented and analyzed. The paper is concluded in Section VII.

II. RELATED WORK

There are several research efforts made on the FL resource utilization optimization in wireless networks in both computing and communications [9], [11], [15], [16]. Particularly, Shi et al. in [15] proposed a client scheduling scheme for FL training under targeted training delay budget. Others like Fu et al. in [9] and Vu et al. in [11] studied the tradeoffs between FL training delay and energy and conducted radio resource allocation for specific (non-optimal) design parameters, which is set based on human heuristics. Luo et al. in [16] proposed cost-efficient design and study how to determine the design parameters (i.e., H). However, among the previous works, the targeted learning models are either relatively simple (i.e., with convex assumptions) or shallow networks, which is inconsistent with the state-of-art DL models, and neglect to reduce the payloads from learning algorithms. A few pioneer works, such as Li et al. in [17] and Shi et al. in [18] have made efforts on energy-efficient FL design on how to determine the learning parameters (e.g., sparsity ratio of model updates). However, they do not consider the mismatch between the huge computing burden on resource limited mobile devices.

Various works in machine learning literature have been developed to design efficient quantization schemes to facilitate on-device learning and communication efficient distributed learning. For on-device learning, "LQ-Net" in Zhang et al. [7] quantized model weights such that the inner products can be computed efficiently on devices. Nevertheless, most of the weight quantization schemes only consider the case of a single device. A few works like Fu et al. [19] proposed a variance-reduced weight quantization scheme in the distributed learning setting to improve the model accuracy but lacks of the discussion about its impact of how to accelerate the service delay. Different from those existing works, we study and implemented the quantization schemes on real-world mobile devices to verify their actual speedup w.r.t computing/service delay for on-device training.

For distributed learning with gradient quantization, recent theoretical works [14], [20], [21] introduced variance-reduced methods and error-feedback schemes to improve the model prediction accuracy and their corresponding convergence analysis assume identical quantization strategies across all the mobile devices, which is not appropriate in the FL setting. Moreover, the effect of precision scaling on both model accuracy and service delay is an open problem that we aim to address in this work. Note that our proposed scheme can be easily incorporated with the resource allocation to further improve the delay efficiency.

III. FEDERATED LEARNING WITH WEIGHT AND GRADIENT QUANTIZATIONS

A. FL Framework With Weight and Gradient Quantization

We consider a wireless FL system consisting of one mobile edge server (e.g., base station or gNodeB) and a set $\mathcal{N} = \{1, 2, \cdots, N\}$ of mobile devices, shown in Fig 1. Each device n has its private dataset $\mathcal{D}_n$ with $D_n = |\mathcal{D}_n|$ training data samples. All devices collaboratively train a global DNN model
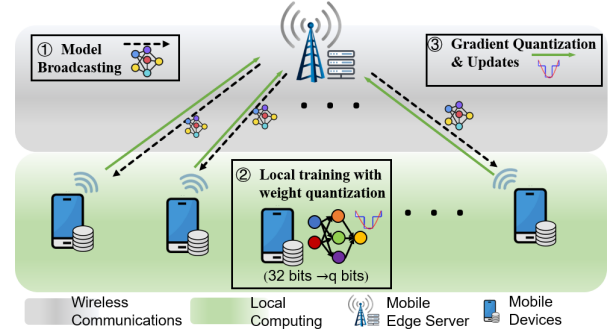

Fig. 1. Federated edge learning system with quantization.

under the coordination of an mobile edge server.[1] The goal is to find the optimal model $\boldsymbol{w} \in \mathbb{R}^d$ to minimize a global empirical risk $F(\boldsymbol{w})$ as follows

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} F(\boldsymbol{w}) = \sum_{n=1}^{N} p_n F_n(\boldsymbol{w}), \qquad (1)$$

where d denotes the total number of model parameters and p_n is the weight of device n such that $\sum_{n=1}^{N} p_n = 1$. $F_n(\boldsymbol{w}) = \frac{1}{D_n} \sum_{i \in \mathcal{D}_n} f_n^i(\boldsymbol{w}; x_n^i, y_n^i)$ is the local training loss of device n, where (x_n^i, y_n^i) is the i-th training sample in $\mathcal{D}_n$. Thus, we have $p_n = D_n / \sum_{n=1}^{N} D_n$.

To achieve high efficient FL over mobile devices, in this paper, we propose the FL with Weight and Gradient Quantization (FL-QuWG) algorithm by considering multi-facet techniques, including local computing control (i.e., H), gradient quantization (i.e., q_g), and on-device weight quantization (i.e., q_w) during local model updates. Local computing control allows mobile devices to perform multiple local training iterations between every two sequential global aggregations, which reduces the total number of FL communication rounds. Gradient quantization enables devices to reduce the transmission bits of the updated gradients. Weight quantization allows mobile devices to store the model parameters and conduct local training with low-precision values to save the memory access and computing delay.

Despite that weight quantizators and gradient quantizators are applied at different FL training stages, their mathematical definitions are the same. Here, we consider an unbiased stochastic quantization scheme, defined as follows.

Definition 1 (Unbiased Quantization Scheme [8]): A randomized mapping $Q : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is an *unbiased quantization scheme* if there exists δ such that $\mathbb{E}[Q(\boldsymbol{w})] = \boldsymbol{w}$, $\mathbb{E}\left[\|Q(\boldsymbol{w}) - \boldsymbol{w}\|_2^2\right] \leq \delta \|\boldsymbol{w}\|_2^2, \forall \boldsymbol{w} \in \mathbb{R}^d$.

In general, each element in $\boldsymbol{w} \in \mathbb{R}^d$ will be independently quantized between $\arg\min \boldsymbol{w}$ and $\arg\max \boldsymbol{w}$ with the data precision of $q \leq 32$ bits. Let q_w and q_g be the quantization strategy of weight quantization and gradient quantization, respectively. Then δ can be defined as $\delta = (1 + \sqrt{2d - 1})/(2(2^q - 1))$ [8]. We note that a smaller q leads

---

[1]For simplicity, this paper only considers synchronous FL settings. We left the asynchronous FL setting as future work.

Algorithm 1 FL With Weight and Gradient Quantization

Input: initial point $\boldsymbol{w}^0 = \boldsymbol{w}_n^0$, learning rate η, number of training iterations K
Output: $\boldsymbol{w}^K$

1: **for** $k = 0, \cdots, K-1$ **do**
2:   **for** mobile device $n \in \mathcal{N}$ in parallel **do**
3:     Computing stochastic gradient
     $\triangledown \widetilde{f}_n\left(\boldsymbol{w}_n^k\right) = \triangledown f_n\left(\boldsymbol{w}_n^k, \widetilde{x}_n, \widetilde{y}_n\right)$ for $\widetilde{x}_n, \widetilde{y}_n \in \mathcal{D}_n$
4:     Update the model parameters
     $\boldsymbol{w}_n^{k+1} \leftarrow Q_{w,n}\left(\boldsymbol{w}_n^k - \eta_n \triangledown \widetilde{f}_n\left(\boldsymbol{w}_n^k\right)\right)$
5:     **if** $((k+1) \bmod H) = 0$ **then**
6:       Send $Q_{g,n}(\boldsymbol{w}_n^{k+1-H} - \boldsymbol{w}_n^{k+1})$ to the FL server
7:       Receive $\boldsymbol{\Delta}_Q^{k+1-H}$ from the FL sever
8:       Set $\boldsymbol{w}_n^{k+1} \leftarrow \boldsymbol{w}_n^{k+1-H} - \boldsymbol{\Delta}_Q^{k+1-H}$
9:     **end if**
10:   **end for**
11:   The FL server aggregates
   $\boldsymbol{\Delta}_Q^{k+1-H} \leftarrow \sum_{n=1}^{N} p_n Q_{g,n}(\boldsymbol{w}_n^{k+1-H} - \boldsymbol{w}_n^{k+1})$
12: **end for**

---

to a higher variance (i.e., a larger δ).[2] We denote q_w and q_g be the quantization strategy to represent the weights and gradients, respectively. The coefficients δ_g and δ_w for gradient quantization and weight quantization schemes, respectively.

The pseudocode of the FL-QuWG algorithm is presented in Alg. 1. Mathematically, all mobile devices start at the same initial point $\boldsymbol{w}_0$. Each device n first quantizes the global model $\boldsymbol{w}$ with $q_{w,n}$ bits ($q_{w,n} \leq 32$) locally and runs H local training iterations with quantized weights and their local datasets (Line 3-4). After completing local on-device training, each device n compresses the updated gradients with $q_{g,n}$ bits ($q_{g,n} \leq 32$) and transmits the quantized gradients to the edge server (Line 6). The edge server then aggregates the gradient information (Line 11) and broadcasts the updated global gradient to the mobile devices (Line 7). Device then update the global model simultaneously using aggregated gradient information and start the local training for next round (Line 8). The above procedure iterates until FL terminates. Let K be the total training iterations, and then the total number of FL rounds is K/H.

B. The Service Delay of FL

Given the training procedure in Alg. 1, the service delay of FL involves computing delay for local training (Line 3-4) and wireless communication delay (Line 6) in each round.[3] We assume the communication/computing delay of a mobile

---

[2]We observe that commonly used quantization schemes stochastic rotated quantization scheme [20] share similar trend and hence we consider the stochastic quantization scheme as an example in this paper to analyze the effect of quantization due to the space limit.

[3]We neglect the costs of the model broadcasting (Line 7) and the model aggregation (Line 11), which are taken by the powerful FL server and whose delay is much smaller than that of model updates. In typical wireless transmission environments (e.g., cellular network), downlink bandwidth is much larger than uplink bandwidth [22]. Therefore, the delay of global model broadcasting using downlink is much shorter than that of local model updates using uplink transmission at mobile devices and can be neglected.

device for every round during FL training is the same, but varies among devices due to device heterogeneity. Therefore, we can neglect the indices for the FL communication rounds. Let T_n denote the per-round service delay of device n to perform on-device training and model updates. We have

$$T_n \triangleq T^{cp}(q_{w,n}, H) + T_n^{cm}(q_{g,n}), \qquad (2)$$

where $T^{cp}(q_{w,n}, H)$ denotes the computing delay related to weight quantization $q_{w,n}$ and local computing control H, and $T^{cm}(q_{g,n})$ represents the communication delay per FL round with gradient quantization $q_{g,n}$. Given straggler issues in synchronized FL systems, the per-round service delay T is determined by the slowest participant, i.e., $T = \max_{n \in \mathcal{N}}\{T_n\}$. Suppose the FL model converges in K iterations using Alg. 1, and the corresponding service delay is

$$T_{tot} \triangleq \frac{K}{H} \cdot \max_{n \in \mathcal{N}} \quad \{T_n\}. \qquad (3)$$

Our objective is to minimize the service delay of FL over mobile devices while guaranteeing FL convergence. We employ the expected gradient norm as an indicator of convergence [23] due to the non-convexity of $F(\cdot)$, where the algorithm achieves an ϵ-suboptimal solution, if $\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\triangledown F(\boldsymbol{w}^k)\right\|_2^2\right]\right] \leq \epsilon$. When ϵ is arbitrarily small, this condition can guarantee the algorithm converges to a stationary point after K training iterations.

Thus, when the FL-QuWG achieves an ϵ-suboptimal solution in K training iterations, the service delay T_{tot} is

$$\min \ T_{tot} \qquad (4a)$$

$$\text{s.t., } \mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\triangledown F(\boldsymbol{w}^k)\right\|_2^2\right]\right] \leq \epsilon. \qquad (4b)$$

Next, we conduct a series of empirical studies to analyze how service delay in (3) and FL convergence in (4b) are affected by different strategies (i.e., H, $q_{w,n}$, $q_{g,n}$).

IV. EMPIRICAL UNDERSTANDING OF DIFFERENT STRATEGIES' IMPACTS ON FL'S SERVICE DELAY

In this section, we empirically study and present the impacts of the local computing control and compression techniques (i.e., gradient quantization and weight quantization) on both learning model accuracy and service delay.

Our experiments use the ResNet20 model on the CIFAR-10 dataset. The computing delay is acquired on two representative mobile devices, Jetson Xavier and Nvidia RTX 8000. Note that the RTX (16.5 TFLOPS) has more computing capabilities than Xavier (1.3 TFLOPS). We set a laptop computer as an FL server and measure the transmission delay of ResNet20 and MobileNetV2 in the Wi-Fi environment. The Wi-Fi uplink transmission rate achieves 102 Mbps on average. More details of the experimental setup can be found in Section VI.

Let $\rho = T^{cp}(32, 1)/T^{cm}(32)$ be the ratio of the computing delay per local iteration and the communication delay in full precision. ρ indicates whether the FL task is computing dominant or communication dominant for mobile devices given their computing capabilities and wireless conditions.
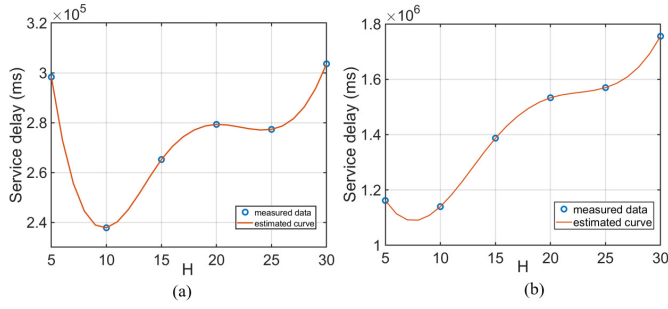
Fig. 2. The service delay vs local computing control using ResNet20 when $q_w = 32$ and $q_g = 32$. The model stops training when the accuracy achieves 98.8%. (a) RTX8000 ($\rho = 0.14$); (b) Xavier ($\rho = 0.61$).
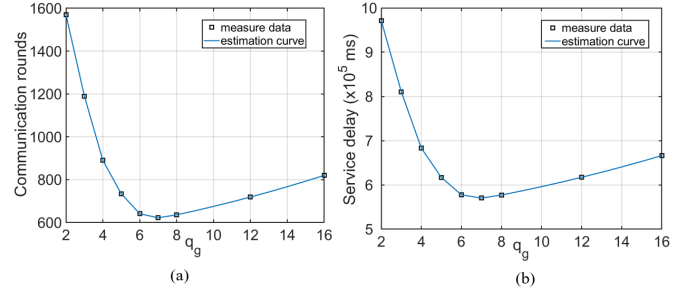

Fig. 3. Impact of gradient quantization level using ResNet20 when $q_w = 32$ and $H = 10$. The model stops training when the accuracy achieves 98.8%: (a) The number of communication rounds vs gradient quantization level; (b) The service delay vs gradient quantization level.

A. Impacts of Local Computing Control on the Service Delay

Local computing control strategies reduce the total number of communication rounds and communication costs by pushing more local computing burdens on the local devices. It has an important influence on model convergence and service delay. Fig. 2 depicts the performance (i.e., service delay to reach the target accuracy) on different H and ρ. It is easily observed that selecting a large number of H is not always effective. The service delay first degrades and then quickly increases once H is larger than a certain threshold. With today's wireless transmission rate, the transmission delay of mobile devices is comparable with its computing delay, that is, $\rho = 0.14$ and $\rho = 0.61$ on average for RTX and Xavier, respectively. In this case, increasing H can only reduce the communication time but not necessarily save the service delay. Moreover, the optimal local iteration is related to the ratio ρ. For example, when ρ decreases (e.g., low communication delay or high computing delay), the optimal strategy of minimizing the service delay is to conduct fewer local training iterations and to update the model more frequently.

B. Impacts of Gradient Quantization on the Service Delay

Similar to the local computing control strategy, gradient quantization reduces the data transmission bits to save communication delay. Unlike the broad discussion on local computing control strategy, the impact of different gradient quantization levels on the service delay is less investigated.

Gradient quantization has a critical influence on both the overall training iterations (K) and communication delay per round (T^{cm}). For T^{cm}, it scales linearly with the quantization level. The extreme case of quantizing data into ternary levels $\{-1, 0, 1\}$ can ideally reduce $32\times$ communication delay per round than full-precision ones. The impact on K is more complicated than its on T^{cm}. We show the impact of the gradient quantization on the communication rounds in Fig. 3 (a) and the corresponding service delay in Fig. 3 (b). In this subsection, for simplicity, we assign the same gradient quantization level to all the devices. It shows that the extreme reduction in precision (i.e., q_g less than 4 bits) requires more communication rounds to recover the model performance, and thus, the service delay significantly increases. As q_g increases, both the communication rounds and service delay decrease sharply. Surprisingly, we discover that moderate quantization

levels (e.g., 5-8 bits) converge faster than the higher-precision counterparts (e.g., 16 bits). The benefit of using moderate quantization levels can be explained by the recent findings on the gradient diversity in distributed learning settings. Gradient diversity measures the difference between the local model gradients and aggregated gradients, and a large gradient diversity encourages active exploration of parameter space [24]. As such, the noise of moderate quantization properly enlarges the diversity that helps the model converge in lower training loss than the full precision. Hence, by choosing quantization levels judiciously, we can guarantee that model converges fast and accurately.

C. Impacts of Weight Quantization on the Service Delay

Weight quantization reduces data precision to save the data size for memory access of mobile devices. It allows low precision operations to speed up the local computing delay on mobile devices. We investigate the speed-up performance of local training with low precision over the full precision counterparts.[4] We perform FL training with different combinations of $(H, q_g, q_w = 16)$ and $(H, q_g, q_w = 32)$. The comparison results are shown in Table I. The "round ratio" denotes the ratio of "Communication rounds for local training with $(H, q_g, q_w = 16)$" and "Communication rounds for local training with $(H, q_g, q_w = 32)$." This "round ratio" indicates that 16-bit is sufficient precision for most network training. More importantly, the models with the weight quantization of 16 bits can save up to $1.6\times$ computing delay with almost the same rate as the full precision case.

TABLE I
IMPACT OF WEIGHT QUANTIZATION

(H, q_g)	$(10, 3)$	$(10, 8)$	$(10, 16)$
round radio	0.99	0.98	0.99
speed up of the service delay	1.64x	1.59x	1.5x
(H, q_g)	$(25, 3)$	$(25, 8)$	$(25, 16)$
round radio	0.99	0.97	0.98
speed up of the service delay	1.68x	1.65x	1.61x

---

[4]Jetson Xavier NX features a Volta GPU with 48 Tensor Cores that can accelerate large matrix operations in 16-bits format. Since Tensor Cores only supports FP16 training, we show the speed-up of the service delay when $q_w = 16$.
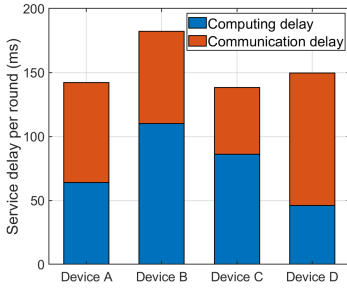
Fig. 4. Service delay vs hetergeneous devices.

D. Impacts of Device Heterogeneity on the Service Delay

We choose different working modes[5] of Xaviers to represent the device heterogeneity w.r.t. the local computing capabilities. Fig. 4 reports the service delay of a subset of participating devices when they perform FL training of ResNet20. Hence, we find that: (1) stragglers greatly slow down model updates in practical FL; and (2) optimized quantization strategies should adapt to heterogeneous computing and communication conditions. Here, optimizing the communication delay can greatly benefit Device D, but it has limited improvement for the service delay of Device B, since it is bottlenecked by the computation delay.

In summary, the empirical results above show that the service delay of FL over mobile devices is affected by multiple factors such as the number of local iterations, weight quantization level, gradient quantization level, devices' heterogeneous computing capabilities, different wireless transmission rates of mobile devices for local model updates. In particular, to effectively reduce the service delay, only reducing the computing/communication delay is not good enough, and the trade-off between computing delay and communication delay has to be taken into consideration. Those empirical observations above urge us to jointly consider local computing control, weight quantization strategy, gradient quantization strategy, and device heterogeneity, characterize computing, communications and FL convergence, and develop a optimized solution to minimize the service delay.

V. MINIMIZING THE SERVICE DELAY OF FL OVER MOBILE DEVICES

The experimental observations above reveal that local computing control H, gradient quantization levels, q_g, and weight quantization levels, q_w play a vital role in the service delay of FL over mobile devices. In this section, we start with discussion on the computing and communication delay model, followed by an approximate analytical relationship between the convergence constraint and the control variables (q_w, q_g, and H). Then, we develop the heterogeneity-aware algorithm to tune q_w, q_g, and H to balance the trade-off between computing and communication delay of mobile devices in FL training under model convergence.

A. Computation and Transmission Modeling

1) Computation Model: GPUs are the most commonly used accelerators for DNN computations. We consider the GPUs instead of CPUs in this work for two reasons. First, CPUs cannot support relatively large and complicated model training tasks. Second, GPUs are more efficient than CPUs for on-device DNN training and are increasingly integrated into today's mobile devices. Hence, the computing delay of GPU-based training is comparable to the communication delay of transmitting FL models in high-speed networks (See the example we use in Section III). The local computing of device n involves data fetching from memory and operating arithmetic in core processors. In Section IV-C, we have observed empirically that weight quantization does speedup the on-device training. Hence, we modify the GPU model in [25] and propose the following delay function to capture the relation among T^{cp}, H and q_w,

$$T_n^{cp}(q_{w,n}, H) \triangleq H\left(\alpha_1(q_{w,n})t_n^{core} + \frac{\alpha_2(q_{w,n})\theta_n^{mem}}{f_n^{mem}}\right) + t_n^0,$$
(5)

where t_n^{core} is the delay coefficient of one training iteration performed in full precision ($q_w = 32$), determined by the specific DNN structure (e.g., layer configuration and batch size); $\alpha_1(q_{w,n})$ denotes the accelerator factor when the model parameter is represented by $q_{w,n}$ bits. θ_n^{mem} and f_{mem} denote the number of cycles for device n to fetch data and memory frequency, respectively; $\alpha_2(q_{w,n})$ denotes the scaling factor. t_n^0 represents the other component unrelated to training task.

Here we define function $\alpha_1(q_{w,n})$ in accordance with NVIDIA documentation [26] as

$$\alpha_1(q_{w,n}) = (1-m) + \frac{m}{(32/q_{w,n})},$$
(6)

where m denotes the fraction of matrix multiplication and convolution operations that can be accelerated by Tensor Core given a DNN configuration and $(32/q_{w,n})$ represents the ideal speed-up for those operations when quantization level $q_{w,n}$ is selected [27]. Then the speedup of using Tensor Core is $\frac{1}{\alpha_1(q_{w,n})}$ [26]. In term of memory access time, we assume that scaling factor $\alpha_2(q_{w,n})$ are linear functions of data bitwidth $q_{w,n}$. This is reasonable since the dataflow (i.e., how data moves in the memory hierarchy) of memory access is fixed given a DNN configuration and hence the number of bits accessed scales linearly with the corresponding bitwidths [28].

To validate the results in Eqn. (5), we implement two DL models, i.e., ResNet20 and MobileNetv2, on a Xavier NX and RTX with Volta GPU. Table II shows a comparison between the estimation in Eqn. (5) and the actual computing delay measured by Jetson stats [29]. The result shows that the estimated delay is close to the actual computing delay.

2) Communication Model: Let S denote the total number of bits transmitted from devices to the edge server. Given the gradient quantization $q_{g,n}$, we have

$$S(q_{g,n}) = s_1 dq_{g,n} + s_0,$$
(7)

TABLE II
ESTIMATION OF ONE PASS COMPUTING DELAY
ON RESNET20 AND MOBILNETV2

Model	ResNet20		Mobilnetv2	
	measured	estimated	measured	estimated
Xavier NX	74.6 (ms)	77.3 (ms)	375 (ms)	401.3 (ms)
RTX 8000	14 (ms)	15.2 (ms)	131 (ms)	144.6 (ms)

where s_0 and s_1 are coefficients determined by additional communication overhead involved in wireless transmission [30]. We consider the transmission rate with orthogonal frequency-division multiple access (OFDMA) scheme with total bandwidth W. The expected transmission rate[6] of mobile device n is given by,

$$r_n(\boldsymbol{\lambda}_n) = \lambda_n W \mathbb{E}_{h_n} \left[\log_2 \left(1 + \frac{P_n^{tran}|h_n|^2}{N_0} \right) \right], \quad (8)$$

where λ_n represents the resource allocation ratio that satisfied $0 \le \lambda_n \le 1$ and $\sum_{n=1}^N \lambda_n = 1$; The expectation is taken over channel fading h_n between mobile device n and the edge server, N_0 is white Gaussian noise, and P_n^{tran} denotes the transmitting power. Thus, the communication delay for transmitting the quantized gradients of the quantized model from device n to the edge server is calculated as

$$T_n^{cm}(q_{g,n}, \boldsymbol{\lambda}_n) \triangleq \frac{S(q_{g,n})}{r_n(\boldsymbol{\lambda}_n)}. \quad (9)$$

B. Relationship Between Convergence and Different Control Variables

In this subsection, we present the convergence analysis of our proposed FL with local computing control, weight quantization and gradient quantization, and the derived convergence result is used to approximate the effect of different strategies (H, q_w, and q_g) to the number of training iterations in constraint (11b).

We make the following assumptions on the non-convex function $F_n, \forall n$, in (1), which are commonly used for non-convex analysis of SGD [31], [32].

Assumption 1: All the loss functions $F_n, \forall n$, are differentiable and their gradients are L-Lipschitz continuous: for all x and $y \in \mathbb{R}^d$, $\|\nabla F_n(x) - \nabla F_n(y)\|_2 \le L \|x - y\|_2, \forall n \in \mathcal{N}$.

Assumption 2: Assume that $\tilde{f}_n$ is randomly sampled from device n's local loss functions. The stochastic gradient is unbiased estimator and its variance with a mini-batch of size M: $\mathbb{E} \left\| \nabla \tilde{f}_n(\boldsymbol{w}^k) - \nabla F_n(\boldsymbol{w}^k) \right\|_2^2 \le \frac{\sigma^2}{M}$, and its second moment is $\mathbb{E} \left\| \nabla \tilde{f}_n(\boldsymbol{w}^k) \right\|_2^2 \le \tau^2, \forall n \in \mathcal{N}$.

Assumption 3: For heterogeneous data distribution it satisfies: $\mathbb{E} \left\| \nabla F_n(\boldsymbol{w}) - \nabla F(\boldsymbol{w}) \right\|_2^2 \le G^2, \forall n \in \mathcal{N}$.

Theorem 1 (Convergence of SDEFL): For SDEFL, under certain assumptions, if the step size satisfies $\eta = \sqrt{\frac{MN}{K}}$, the convergence rate of the proposed scheme satisfies, with

---

the quantization strategies, q_g, q_w and local computing control strategy H,

$$\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E} \left[\left\| \nabla F(\boldsymbol{w}^k) \right\|_2^2 \right]$$
$$\le \frac{4\mathbb{E} \left[F(\boldsymbol{w}^0) - F(\boldsymbol{w}^K) \right]}{\sqrt{MNK}} + \frac{2 L\sigma^2 (2H\bar{\delta}_g + \bar{p})}{\sqrt{MNK}}$$
$$+ \frac{12MLH\bar{\delta}_g G^2}{\sqrt{MNK}} + 2L\sqrt{d}\tau \sum_{n=1}^N p_n^2(\delta_{g,n} + 1)\delta_{w,n}, \quad (10)$$

where $\bar{\delta}_g = \sum_{n=1}^N p_n^2 \delta_{g,n}$, $\bar{p} = \sum_{n=1}^N p_n^2$ and $\bar{\delta}_w = \sum_{n=1}^N p_n^2 \delta_{w,n}$.

Proof: See Appendix. □

Theorem 1 implies that the convergence rate of SDEFL is $\mathcal{O}((1 + \bar{\delta}_g H)/\sqrt{NK})$. It converges at the same rate as the full-precision SGD, but only to a non-zero error (the last term) associated with the quantization resolution δ_g, δ_w, and model dimension d. Hence, weight quantization makes FL converge to a neighborhood of the optimal solution without affecting the convergence rate. The 16-bit weight quantization does not impede down convergence performance, which is consistent with our empirical observations of weight quantization. Furthermore, 16-bit weight quantization has little effect on model learning performance. While gradient quantization and local iterations would affect the convergence by a factor of $\mathcal{O}(H\bar{\delta}_g)$. Given a target model accuracy, a higher value H and a lower value $q_{g,n}$ result in a higher bound of training iterations. On the other hands, A higher H may lead to fewer communication rounds $\frac{K}{H}$. Based on our key observation that the computing delay for on-device training and communication delay are on the same scale, we can use H to trade-off computing and communication delays to optimize the service delay as shown in Section III-A. Similarly, a smaller value of $q_{g,n}$ (i.e., less than 8 bits) leads to less communication delay per round and more training iterations. It also demonstrates a trade-off computing and communication delay to optimize the service delay as shown in Section III-B.

C. Problem Formulation

Given the computing model and the communication model above, we reformulate the problem in (4a) as follows

$$\min_{\substack{H, K, \\ q_{g,n}, q_{w,n}}} T_{tot} \quad (11a)$$

$$\text{s.t.,} \quad \frac{A_1 + A_0 H \sum_{n=1}^N p_n^2 \delta_{g,n}}{\sqrt{NK}} + C_0 \sum_{n=1}^N p_n^2 \delta_{w,n}$$
$$+ B_0 H \sum_{n=1}^N p_n^2 \delta_{g,n} \delta_{w,n} \le \epsilon, \quad (11b)$$

$$H \in \mathcal{H}, \quad q_{g,n} \in \mathcal{Q}_g, \ q_{w,n} \in \mathcal{Q}_w, \ \forall n \in \mathcal{N}. \quad (11c)$$

Here, we leverage the convergence upper bound as an approximation of constraint (4b) to achieve ϵ-global model convergence. Constraint (11b) represents that after K training iterations, the training loss should be smaller than a pre-set

threshold ϵ. For simplicity, we use the coefficients A_1, A_0, B_0, and C_0 to characterize the loss function related properties and statistical heterogeneity of non-i.i.d. data in Theorem 1. With the proper values of A_1, A_0, B_0, and C_0, we can characterize the nonlinear relationship between $\frac{K}{H}$ and $q_{g,n}$ as shown in Section III-B. The coefficients can be estimated by using a sampling set of experimental training results. Constraints (11c) indicate that optimization variables take the values from a set of non-negative integers. We relax K, q_w, and q_g as continuous variables for theoretical analysis, which later are rounded back to the nearest integer. For the relaxed problem, we observe that the variable K is monotonously decreased with the objective function. Hence, for optimal K, constraint (11b) is always satisfied with equality, and we can obtain K as

$$K = \frac{(A_1 + A_0 H \sum_{n=1}^{N} p_n^2 \delta_{g,n})^2}{N(\epsilon - B_0 H \sum_{n=1}^{N} p_n^2 \delta_{g,n} \delta_{w,n} - C_0 \sum_{n=1}^{N} p_n^2 \delta_{w,n})^2}. \tag{12}$$

For notation simplicity, we further simplify the description of GPU time model as $T_n^{cp} = H(\beta_n^1 q_{w,n} + \beta_n^0) + t_n^0$, where $\beta_n^0 = a_n^0 + (1-m)t_n^{core}$ and $\beta_n^1 = a_n^1 \theta_n^{mem}/f_n^{mem} + m t_n^{core}/32$ and communication model as $u_n^1 = ds_1/r_n$ and $u_n^2 = s_0/r_n$. By substituting the (12) into its expression, we reformulate the problem (11a) as follows,

$$\min_{H,q_{g,n},q_{w,n}} \frac{(A_1 + A_0 H \sum_{n=1}^{N} p_n^2 \delta_{g,n})^2}{HN(\epsilon - \sum_{n=1}^{N} p_n^2 \delta_{w,n}(B_0 H \delta_{g,n} + C_0))^2}$$
$$\cdot \max_{n \in \mathcal{N}} \{u_n^1 q_{g,n} + u_n^2 + H(\beta_n^1 q_{w,n} + \beta_n^0) + t_n^0\} \tag{13a}$$

$$\text{s.t. } H \in \widetilde{\mathcal{H}}, \quad q_{g,n} \in \widetilde{\mathcal{Q}}_g, \ q_{w,n} \in \mathcal{Q}_w, \ \forall n \in \mathcal{N}, \tag{13b}$$

where $\widetilde{\mathcal{H}}$ and $\widetilde{\mathcal{Q}}_g$ are the relaxed set. The problem (13) is non-convex problem. The non-convexity arises from (1) the min-max formulation; and (2) multiplicative form of $q_{w,n}$, $q_{g,n}, \forall n \in \mathcal{N}$ and H in both the objective function and constraints, which makes the optimization NP-hard to solve.

D. Solutions to the Optimal SDEFL

To reduce the computational complexity, we convert the optimization problem in (13) to a geometric programming problem, which is then converted into a standard convex problem via log-sum-exp forms [33]. Then we introduce two slack variables, $\Psi \geq 0$ and $\phi \geq 0$, such that

$$\min_{\substack{\Psi,\phi,H, \\ q_g,n,q_w,n}} \Psi, \tag{14a}$$

$$\text{s.t. } \frac{\phi^2}{HN} \left(u_n^1 q_{g,n} + u_n^2 + H(\beta_n^1 q_{w,n} + \beta_n^0) + t_n^0 \right) \leq \Psi, \quad \forall n \in \mathcal{N}, \tag{14b}$$

$$0 \leq \frac{A_1 + A_0 H \sum_{n=1}^{N} p_n^2 \delta_{g,n}}{\epsilon - \sum_{n=1}^{N} p_n^2 \delta_{w,n}(B_0 H \delta_{g,n} + C_0)} \leq \phi. \tag{14c}$$

The problem (14) is equivalent to the original standard form problem (13). Indeed, since the objective function in (14) is

a increasing function of Ψ and ϕ, the constraints (14b) and (14c) always hold with equality for the optimal solution.

According to the Definition 1, we can assume that $\delta_{g,n}$ and $\delta_{w,n}$ decrease monotonically with $q_{g,n}$ and $q_{w,n}, \forall n$, respectively. Without loss of generality, we consider the coefficient function of δ in term of q as $\log(\delta) = a - b\log(2^q - 1)$ where $a > 0, b > 0$ (See Figure 5 in [19]) to characterize the relationship between q and δ in SQ schemes.[7] We consider the following coefficient functions, for $\xi \in \{g, w\}$,

$$\delta_{\xi,n} = 10^a/(2^{q_{\xi,n}} - 1)^b, \quad n \in \mathcal{N}. \tag{15}$$

We then introduce additional auxiliary variables $v_{\xi,n}$, where $q_{\xi,n} = \log_2(1 + 1/v_{\xi,n}) \approx \log(2)/v_{\xi,n}$ and $\delta_{\xi,n} = 10^a v_{\xi,n}^b$.

Further, since ϵ is a constant and we know that $\epsilon \geq B_0 H \sum_{n=1}^{N} p_n^2 \delta_{g,n} \delta_{w,n}$, then constraint (14c) is equivalent to

$$\frac{A_1 + A_0' H \sum_{n=1}^{N} p_n^2 v_{g,n}^b}{\phi \epsilon}$$
$$+ \frac{\sum_{n=1}^{N} p_n^2 v_{w,n}^b (B_0' H v_{g,n}^b + C_0')}{\epsilon} \leq 1, \tag{16}$$

where $A_0' = A_0 10^a$, $B_0' = B_0 10^{2a}$, and $C_0' = C_0 10^a$. With auxiliary variables $v_{g,n}$ and $v_{w,n}, \forall n \in \mathcal{N}$, we can observe that the objective function in (17) is *monomial* and all the inequality constraints are *posynomials* [33], which satisfies geometric programming (GP) in standard form. Then we make a logarithmic change of all the variables and multiplicative constants and equivalently transform the following GP problem in convex form [33]:

$$\min_{\substack{\Psi,\widetilde{\phi},\widetilde{H}, \\ \widetilde{v}_{g,n},\widetilde{v}_{w,n}}} \Psi, \tag{17a}$$

$$\text{s.t. } \log \left(\exp\left(2\widetilde{\phi} - \widetilde{H} - \widetilde{v}_{g,n} + \log(\widetilde{u}_n^1) \right) \right.$$
$$+ \exp\left(2\widetilde{\phi} - \widetilde{H} + \log(\frac{u_n^2}{N}) \right) + \exp\left(2\widetilde{\phi} + \log(\frac{\beta_n^0}{N}) \right)$$
$$+ \exp\left(2\widetilde{\phi} - \widetilde{H} + \log(\frac{t_n^0}{N}) \right)$$
$$\left. + \exp\left(2\widetilde{\phi} - \widetilde{v}_{w,n} + \log(\widetilde{\beta}_n^1) \right) \right) \leq \Psi, \quad \forall n \in \mathcal{N}, \tag{17b}$$

$$\log \left(\sum_{n=1}^{N} \exp\left(\log(A_0' p_n^2/\epsilon) + \widetilde{H} + b\widetilde{v}_{g,n} - \widetilde{\phi} \right) \right.$$
$$+ \sum_{n=1}^{N} \exp\left(\log(B_0' p_n^2/\epsilon) + \widetilde{H} + b\widetilde{v}_{g,n} + b\widetilde{v}_{w,n} \right)$$
$$+ \sum_{n=1}^{N} \exp\left(\log(C_0' p_n^2/\epsilon) + b\widetilde{v}_{w,n} \right)$$
$$\left. + \exp\left(\log(A_1/\epsilon) - \widetilde{\phi} \right) \right) \leq 0, \tag{17c}$$

where $\widetilde{u}_n^1 = \log(2)u_n^1/N$ and $\widetilde{\beta}_n^1 = \log(2)\beta_n^1/N$. Through the change $\widetilde{\phi} = \log(\phi), \widetilde{H} = \log(H), \widetilde{v}_{g,n} = \log(v_{g,n}), \widetilde{v}_{w,n} = \log(v_{w,n}), \forall n$, we get the convex form, which can be confirmed to be convex optimization because of the convexity

---

[7]We can set different values of a and b for different SQ schemes. For example, $a = \log((1 + \sqrt{2d - 1})/2)$ and $b = 1$ in QSGD [8].

Algorithm 2 GP-Based Local Computing Control and Quantization Selection for SDEFL in (13)

1: **Input:** $N, r_n, t_n^{core}, f_n^{mem}, t_n^0$, estimated model convergence parameters (A_0, A_1, B_0, C_0), model training loss ϵ, and parameters of the SQ scheme (a, b)
2: **Output:** $H^*, \{q_{g,n}^*\}_{n=1}^N, \{q_{w,n}^*\}_{n=1}^N$
3: Substitute $N, r_n, t_n^{core}, f_n^{mem}, t_n^0, A_0, A_1, B_0, C_0, \epsilon, a$, and b into (13) and reformulate problem (13) into GP in (17)
4: Solve GP in (17) via primal-dual interior point methods [34] to obtain the optimal $\widetilde{H}^*, \{\widetilde{v}_{g,n}^*, \widetilde{v}_{g,n}^*\}_{n=1}^N$
5: Compute $H = \exp(\widetilde{H}^*)$, $q_{g,n} = \log_2(1 + 1/\exp(\widetilde{v}_{g,n}^*))$, and $q_{w,n} = \log_2(1 + 1/\exp(\widetilde{v}_{w,n}^*))$
6: Round $\lfloor H \rceil$, $\lfloor q_{g,n} \rceil$, and $\lfloor q_{w,n} \rceil$, and set the combination of weight and gradient quantization strategies and local computing control that has the minimum objective value in (13a) as ($q_{g,n}^*$ and $q_{w,n}^*$, H^*).
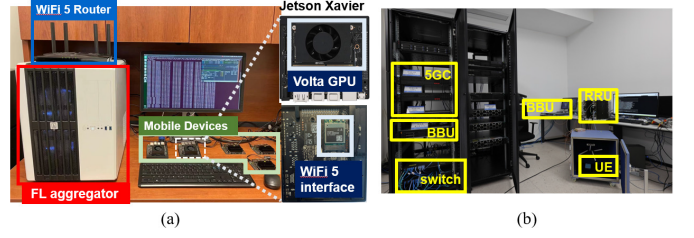

Fig. 5. The SDEFL system configuration: (a) The SDEFL testbed with RTX 8000 (FL aggregator) and Jetson Xavier kits (FL clients) in our lab; (b) The 5G emulation platform in the Purple Mountain Laboratories, Nanjing, China.

of log-sum-exp function [33]. Hence, it can be efficiently solved optimally by the edge server through the interior point primal dual method [34], with polynomial running times that are $\mathcal{O}(\sqrt{2N+3})$, where $2N+3$ is the total number of optimization variables in our proposed service delay minimization and N is the total number of participating devices. Once the optimal vector $(\Psi^*, \widetilde{\phi}^*, \widetilde{H}^*, \widetilde{v}_{g,n}^*, \widetilde{v}_{w,n}^*)$ is determined, we can compute the strategy of $(H^*, q_{g,n}^*, q_{w,n}^*)$, that is $H^* = \exp(\widetilde{H}^*)$, $q_{g,n} = \log_2(1 + 1/\exp(\widetilde{v}_{g,n}^*))$, and $q_{w,n} = \log_2(1 + 1/\exp(\widetilde{v}_{w,n}^*))$, for each device n. The overall algorithm for obtaining $(H^*, q_{g,n}^*, q_{w,n}^*)$ is summarized in Alg. 2

The edge server is in charge of solving the optimization in (14). It is practical because in the FL protocol in [35] requires mobile devices to check in with the FL server first. Hence, the FL server can collect the information from mobile devices and find the optimal strategy $(H, q_{w,n}$ and $q_{g,n})$ for each device via an optimization algorithm.

VI. SDEFL PERFORMANCE EVALUATION

In this section, we illustrate the setup of testbeds, and conduct extensive experiments/simulations using different learning models and wireless transmission techniques (i.e., Wi-Fi 5 and 5G) to evaluate the performance of the proposed SDEFL in terms of service delay and learning accuracy.

A. Experimental Setup

1) System Configurations: We evaluate the SDEFL's performance in both testbeds (for mobile devices with Wi-Fi 5 transmission capability) and a simulated environment (for mobile devices with 5G transmission capability). Our SDEFL testbed consists of an NVIDIA RTX 8000 as the edge server and several Jeston Xavier kits as mobile devices, as shown in Fig. 5(a). Each Xavier has a Volta architecture, with 384 NVIDIA CUDA cores and 48 Tensor cores, and 8GB GPU memory. All devices are wirelessly connected via Wi-Fi 5 according to the WebSocket [36] communication protocol. We also adopt the iPerf3 network speed test tool to measure

the achievable bandwidth of the wireless transmissions. As for the simulated environment, since Jeston Xavier doesn't support 5G transmissions,[8] we have to separately test the computing delay and 5G transmission delay of SDEFL. Here, we measure the computing delay using Xaviers in our lab, measure the transmission delay using a 5G emulation platform, and combine the results to evaluate the service delay of SDEFL. The system configuration is shown in Fig. 5.

2) Learning Datasets and Models: We evaluate our results on the CIFAR-10 dataset, which consists of 50000 training images and 10000 test images in 10 classes. Without loss of generality, we choose to train deep neural networks ResNet20 and MobileNetv2 from the scratch, since these two DNN models have different architectures and numbers of model parameters. For all experiments, we set the batch size $M = 128$. In each round of FL, we consider 10 participating mobile devices, which run H steps of SGD in parallel. We employ an initial learning rate $\eta = 0.1$ with a fixed decay rate of 0.996. Each result is averaged over 5 experiments.

3) Implementation: We implement SDEFL and the other baselines by building on top of Flower [37]. In total, we add 400 lines to perform weight and gradient quantization. In particular, we use NVIDIA's APEX library for quantization implementation. The optimal local computing control, weight quantization, and gradient quantization strategies are obtained by solving the SDEFL optimization problem in (13). The values of $t_{core}, t_n^0, \alpha_2(q_w), \theta^{mem}, s_1$ and s_0 used in SDEFL optimization are derived from platform-based experiments by measuring the average computing delay and communication delay using Nvidia profiling tools [38] on Jetson Xaiver in our testbed. The average $t_n^0 = 4.69$ ms and $\alpha_2(q_w) = 0.003q_w + 0.3$. The average $s_1 = 0.0000166$ and $s_0 = 0.2$ MB. Besides, t_{core} and m, θ^{mem} depend on DNN configurations. Here, we take ResNet-18 as example. The average $t_{core} = 76$ ms, $\theta^{mem} = 939706055$, and $m = 0.32$. For the traditional SQ scheme, $a = \log((1 + \sqrt{2d-1})/2)$ and $b = 1$ [8]. The SDEFL optimization is conducted using "fmincom-geometric" solver. It takes 30 ms to solve the proposed optimization by using MATLAB, which is definitely affordable for the FL edge server. Besides quantization implementation, there are some other implementation details as follows.

Experiments with i.i.d data: We set up the testbed and conduct the first series of experiments using ResNet20 and

---

[8]The NVIDIA Jetson Xavier NX developer kits we have in the lab don't support Wi-Fi 6 or 5G transmission.
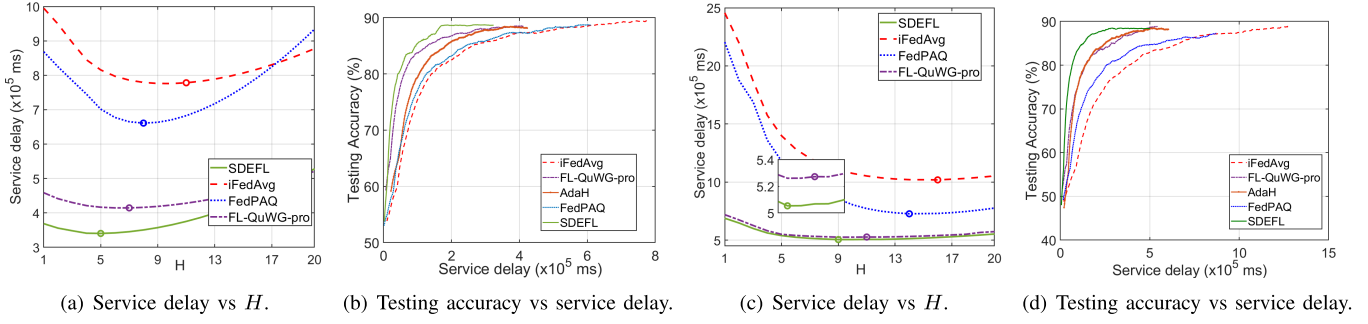
Fig. 6. Training performance of ResNet20. The estimation of $A_0 = 0.35$, $A_1 = 32.3$, $B_0 = 0.001$, and $C_0 = 0.06$. (a)-(b): Experimental Results (Wi-Fi 5: 88 Mbps); (c)-(d): Emulation Results (5G: 14 Mbps). The dots show that the optimal H in different schemes.

MobilenetV2 with CIFAR10 dataset with $N = 10$ participating devices with homogeneous data distribution. In particular, we uniformly distribute 6,000 data samples and 10 classes among the participating devices.

Experiments with non-i.i.d data: With the same testbed, we conduct the second series of experiments using ResNet20 and CIFAR10 dataset with heterogeneous data distribution. To generate unbalanced data, we sample the number of data samples via a lognormal distribution with a standard deviation from $\{0.3, 0.6\}$, and each device contains only 4 digits labels.

Implementation of wireless transmissions: Using the 5G emulation platform with the configuration of a frequency of 2.6GHz and a bandwidth of 100MHz, we measure the achievable 5G uplink transmission rate, which is 14 Mbps on average with standard deviation (SD) 3.85 Mbps. Using the SDEFL testbed in our lab, we measure Wi-Fi 5's achievable uploading transmission rate from Xaviers to the RTX server, which is 88 Mbps on average with SD 16 Mbps.

4) Peer Schemes for Comparison: We compare our proposed SDEFL scheme with the following FL schemes: (1) *iFedAvg*: Similar to the work in [16],[9] all the devices follows FedAvg algorithm with the optimal value of H and do not consider weight and gradient quantization. (2) *FedPAQ*: In FedPAQ [21], all the devices perform local training with full precision weight and transmit the quantized version of model updates to the edge server. In this scheme, all clients are assign with the same gradient quantization level. (3) *AdaH*: Similar to the work in [39], all the devices follow the same dynamic H policy. The dynamic H policy gradually decreases H-value as $H^t = \sqrt{\frac{F^t}{F^0}} H^0$, and we set $H^0 = 30$. (4) *FL-QuWG-pro*: We assign different gradient quantization levels proportionally according to the allocated bandwidth of each mobile device.

B. Results and Analysis

1) Service Delay and Learning Accuracy: Fig. 6 shows the service delay for reaching the target loss of 0.15 under different numbers of local iterations H using our testbed with Wi-Fi 5 transmissions. In Fig. 6(a) and Fig. 6(c), where the communication and computing delay is comparable, the optimal H exists (not extremely big/small), which further

confirms our empirical and theoretical analysis. Besides, from Fig. 6(b) and Fig. 6(d), we observe that *AdaH*, *FedPAQ*, *FL-QuWG-pro*, and SDEFL can effectively reduce the service delay. Moreover, our proposed SDEFL scheme outperforms the others: 1) It is because SDEFL is more flexible than *FedPAQ* in terms of quantization decision making, which can better fit the heterogeneous communication conditions across participants. 2) SDEFL outperforms *AdaH* since the dynamic policy in *AdaH* only depends on the training loss and does not consider the trade-off between communication and computing delay. It also ignores the heterogeneous resource across participants. Different from it, the proposed SDEFL focuses on the trade-off between communication and computing delay and device heterogeneity to further reduce the service delay. 3) Compared with *FL-QuWG-pro*, it demonstrates the effectiveness of the proposed optimization that can find the optimal strategies for different mobile devices. It shows when reaching FL convergence, SDEFL can reduce 56% service delay with 0.3% accuracy loss compared with *FedAvg*, and reduce round 50% service delay with 0.23% accuracy loss compared with *FedPAQ*.

2) Impacts of Wireless Transmission Rates: As the measured transmission rate of 5G is slower than that of Wi-Fi 5, the SDEFL with 5G transmissions has larger service delay as shown in Fig. 6. Actually, different transmission rates yield different workload allocations between "working" and "talking" to achieve the minimum service delay, which is reflected by the computing delay/communication delay ratio ρ defined in Sec. IV. Here, ρ equals to 0.9 for SDEFL with Wi-Fi 5 transmission, and ρ equals to 0.1 for SDEFL with 5G transmission. In Fig. 6 (a) and (c), we find that the optimal value of H decreases as ρ increases (from 0.1 to 0.9). The reason is that SDEFL prefers to reduce the service delay by "talking" more in the case of the high wireless transmission rate, i.e., the accumulated wireless transmission delay is comparable or smaller than the accumulated local computing delay for mobile devices during FL training. It further verify our observation and theoretical analysis about the trade-off between communication and computation in FL over mobile devices. Besides, for different ρ values, the proposed SDEFL scheme outperforms other schemes.

3) Impacts of Learning Model Dimensions: We evaluate the proposed SDEFL with different learning models. In particular, the model sizes (i.e., d) of ResNet20 and MobileNetv2 are

---

[9]In this work, Luo et al consider the client selection and the optimal number of participant is full participants when we only consider the service delay. Hence, we only consider the optimal value of H.
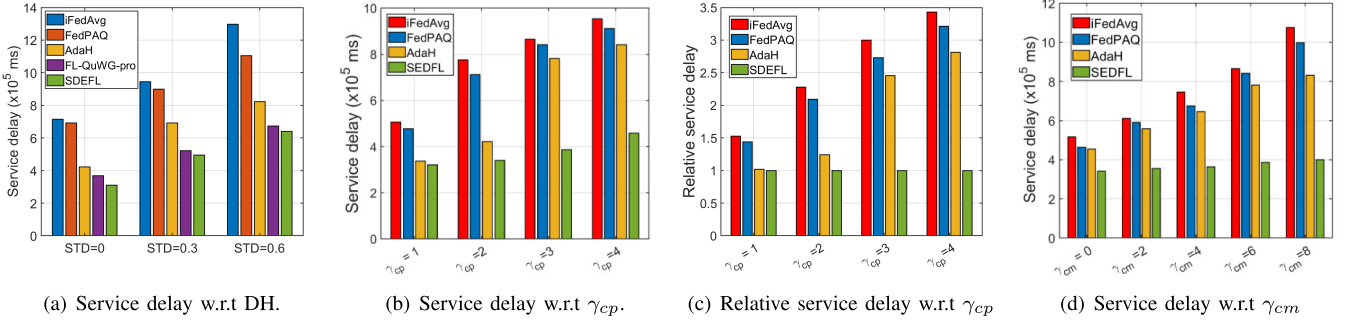
(a) Service delay w.r.t DH.   (b) Service delay w.r.t γ_{cp}.   (c) Relative service delay w.r.t γ_{cp} (d) Service delay w.r.t γ_{cm}

Fig. 7.   Service delay with different heterogeneous settings.

TABLE III
TESTING ACCURACY WITH DIFFERENT FL SCHEMES

Model	Methods	Acc.(%)	Service delay in (13) (ms)	Measured delay (ms)
ResNet20	iFedAvg [16]	**89.98**	776435	714653
	FedPAQ [21]	87.15	682118	655170
	FL-QuWG-pro	87.15	415864	362726
	SDEFL	88.64	**340273**	**316819**
MobileNetv2	iFedAvg [16]	**88.19**	1933978	1765978
	FedPAQ [21]	82.62	1808992	1704431
	FL-QuWG-pro	87.15	1109046	1011043
	SDEFL	83.47	**931106**	**871240**

0.27M and 3.4M, respectively. From Table III, we observe that the service delay of MobileNetv2 training is larger than that of ResNet20 since a more complex learning model requires more communication and computing delay. Besides, the proposed SDEFL can save more service delay in MobileNetv2 than that in ResNet20. In the case of MobileNetv2, it shows that when reaching FL convergence, SDEFL can reduce 46% service delay compared with *FedAvg*, reduce 38% delay compared with *FedPAQ*, and reduce 15% delay compared with *FL-QuWG-pro*. We also calculate the ρ in Section IV of the system straggler: $\rho = 0.61$ in the case of ResNet and $\rho = 4.21$ in the case of MobileNetv2. The computing delay has more profound impact than communication delay in MobileNetv2. Hence, the *FedPAQ* can only consume 6% less than *FedAvg*. Since SDEFL can leverage weight quantization to reduce the heavy computing workload incurred in large networks, it can effectively reduce the service delay. However, the accuracy of SDEFL drops by 6% in MobileNetv2, which is higher than the 3% drop in ResNet20. A larger d leads to a larger loss upon convergence. Those results are consistent with the analysis in Theorems 1. Besides, we apply the optimal solution derived by Alg. 2 to perform the FL training in our testbed and measure the corresponding service delay using Pytorch. The measured delay is presented in Table III. It shows that our approach has the maximum error of 12% and an average error of 8%.

4) Impacts of Data & Device Heterogeneity: We further evaluate the performance of SDEFL with different data distributions in the context of skew class distribution and unbalanced number of training data samples. The model is trained by ResNet20. Sample distributions become skewer as the standard deviation (STD) goes from 0, 0.3 to 0.6. As shown in Fig. 7(a), we find that training with non-IID data incurs

a longer service delay than that of the IID case. From the results in Fig. 7(a), we observe that a large value of G requires more training iterations to converge, which verifies the analysis in Theorem 1. Compared with *FedAvg* The proposed SDEFL can efficiently reduced the service delay by 0.42, 0.72 and 0.6 in the case of "STD = 0," "STD = 0.3," "STD = 0.6," respectively.

We now examine the impact of device heterogeneity (DH) w.r.t computing capacities and wireless bandwidth on the service delay. Here, we divide ten participating devices into four groups, corresponding to four capacity levels. Assume that mobile devices belonging to the same group are allocated with proportional wireless bandwidth for gradient exchanges. To vary r_n, we use Linux *tc*, *qdisc*, and *iptables* to limit the upload bandwidth of the connected devices with a tunable parameter γ_{cm} such as $\lambda_n - 0.05\gamma_{cm}$, $\lambda_n - 0.03\gamma_{cm}$, $\lambda_n + 0.03\gamma_{cm}$ and $\lambda_n + 0.05\gamma_{cm}$, respectively. The values of $\gamma_{cm} \in \{0, 2, 4, 6, 8\}$. On the other hand, to vary t_n^{comp}, we set $\gamma_{cp} \in \{1, 2, 3, 4\}$ to control the total types of working modes. For instance, $\gamma_{cp} = 3$ represents that mobile devices perform with three different kinds of working power and GPU cores. Higher values of γ_{cm} and γ_{cp} mean that devices have more diverse computing and communication conditions. In the following, we fix $\gamma_{cp} = 3$ when varying γ_{cm} and fix $\gamma_{cm} = 6$ when varying γ_{cp}. From Fig. 7(b), we observe that the service delay grows with the device heterogeneity, indicating a negative impact of heterogeneous mobile devices on FL training. Further, Fig. 7(c) and Fig. 7(d) show that our proposed scheme SDEFL can greatly reduce the service delay when the device capabilities are highly diverse since we consider the differences among mobile devices to the gradient quantization strategy and optimize the local computing control selection.

5) Numerical Illustration of Optimal $(H, q_{g,n}, q_{w,n})$ Strategies: Due to the fact that the Tensor cores in Xaiver NX only support hardware acceleration for training with FP16, we only consider the feasible set of weight quantization level as $q_w \in \{16, 32\}$ in the testbed experiments above.

In the following, we conduct simulations with the extended feasible set of weight quantization level, and further examine the impacts of available bandwidth on the optimal values of $(H, q_{g,n}, q_{w,n})$. Here, we set the number of mobile devices 40 and the FL task is ResNet-20 with Cifar10. The expanded feasible set of weight quantization level $q_w \in \{4, 8, 16, 32\}$,

(a) Impact of available bandwidth to optimal $\{q_{w,n}\}$ and H.

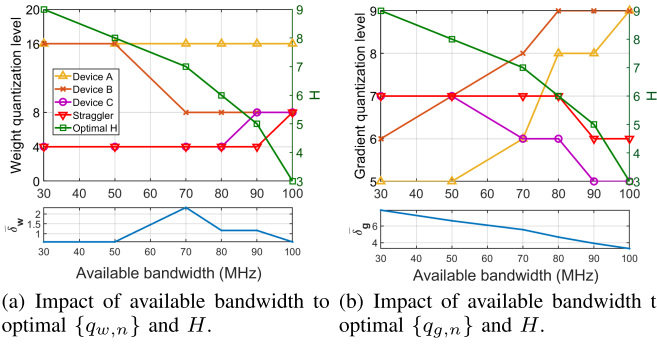(b) Impact of available bandwidth to optimal $\{q_{g,n}\}$ and H.

Fig. 8. Optimal strategies of different devices. Here, the relationship of computing capabilities of four devices is $T_S^{cp} < T_C^{cp} < T_B^{cp} < T_A^{cp}$ and the relationship of communication conditions of four devices is $r_A^{cm} < r_B^{cm} < r_C^{cm} < r_S^{cm}$. For straggler device and Device C, they are bottlenecked by the computing delays. For Devices A and B, they are bottlenecked by the communication delays.

(a) i.i.d distribution.
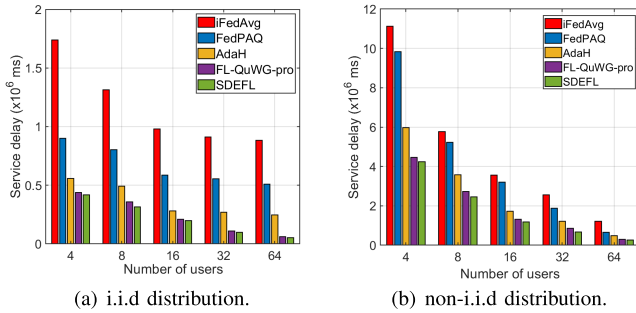
(b) non-i.i.d distribution.

Fig. 9. Service delay with numbers of devices.

which is a standard-setting and hardware friendly. Eqn. (5) and Eqn. (9) are used to calculate the computing and communication delays, respectively. The optimal values of $(H, q_{g,n}, q_{w,n})$ are obtained by Alg. 2. Let $\bar{\delta}_g = \sum_{n=1}^{N} p_n^2 \delta_{g,n}$ and $\bar{\delta}_w = \sum_{n=1}^{N} p_n^2 \delta_{w,n}$ be the average quantization error of gradient and weight, respectively.

In Fig. 8, we show the optimal quantization levels of the straggler device and three other mobile devices. We see that SDEFL prefers "working" more and "talking" less in the case of small wireless bandwidth, as expected. Furthermore, we see that the tradeoffs between "working" and "talking" among devices vary. Rather surprisingly, we find that, as the available bandwidth increases, for the devices with small computing capabilities, the optimal weight quantization levels ($q_{w,S}^*$ and $q_{w,C}^*$) increase while their optimal gradient quantization levels decrease; for the devices with slow communication rates, the optimal weight quantization level of Device B ($q_{w,B}^*$) decreases while his optimal gradient quantization level increases. In fact, the per-round computing delay of the straggler is not increased due to a small value of H.[10] Besides, with smaller value of $\bar{\delta}_w$ and $\bar{\delta}_g$ (the bottom of Figs. 8(a) and 8(b)) in the case of the large available bandwidth (e.g., 90 and 100), the number of training iteration to converge (i.e., K in Eqn. 12) is smaller. Thus the service delay can be efficiently reduced.

[10]In this case, the estimated per-round computing delay of ($q_{w,n} = 8$, $H = 5$) is 247ms while the estimated per-round computing delay of ($q_{w,n} = 4, H = 5$) is 207ms.

It demonstrates that our SDEFL fully consider the device heterogeneity and allows devices to balance the workload between "working" and "talking" in saving their service delay during FL training.

6) Impact of the Number of Users: We evaluate the impact of the number of users in the simulated settings. Fig. 9 demonstrates that bring more participating devices helps reduce the overall service delay in both i.i.d and non-i.i.d cases. The reason is that increasing the number of users can help speed up the convergence rate, and thereby reduce the service delay, which is also consistent with the sub-linear speedup in Theorem 1. In i.i.d case, as N keeps growing, the marginal service delay reduction becomes smaller and smaller. Beside, among different settings, the proposed SDEFL is superior to other schemes in terms of the service delay.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have studied the service delay efficient FL (SDEFL) over mobile devices via joint design of local computing control, weight quantization and gradient quantization. We have empirically investigated the impacts of the weight quantization, gradient quantization and local computing control strategies on the service delay, and provided the convergence rate of FL with compression from a theoretical perspective. Guided by the derived theoretical convergence rate, we have investigated the tradeoff between "working" (i.e., computing delay) and "talking" (i.e., communication delay), and formulated the SDEFL training problem into a mixed integer programming optimization. We have converted it to the equivalent convex programming optimization, and developed feasible solutions. Extensive experimental and emulation results have demonstrated the effectiveness and efficiency of our proposed SDEFL scheme, and its advantages over peer designs in various learning and wireless transmission settings.

Note that current commercial off-the-shelf mobile Volta and Tuning GPUs can only support the speedup for FP16 training (i.e., 16-bit weight quantization). Therefore, to implement fast local model training with lower precision than 16-bit (in particular lower than 8-bit), we should jointly consider the algorithm development with the hardware/architecture design. As a possible future extension of our work, we would like to expand our existing SDEFL testbed with FPGA based Xilinx Zynq [40] via hardware and software co-designs to support lower level weight quantization (e.g., 8-bit or 4-bit quantization) and show its advantages.

APPENDIX
PROOF OF THEOREM 1

A. Additional Notation

For simplicity of notations, we denote the error of weight quantization $\boldsymbol{r}_n^k \triangleq Q_w(\boldsymbol{w}_n^{k+1}) - \boldsymbol{w}_n^{k+1}$, and the local "gradient" with weight quantization as $\widetilde{\boldsymbol{g}}_n^k \triangleq \triangledown \widetilde{f}_n(\boldsymbol{w}_n^k) - \boldsymbol{r}_n^k/\eta$, $\forall n \in \mathcal{N}$. Since the quantization scheme Q_w we used is an unbiased scheme, $\mathbb{E}_Q[\boldsymbol{r}_n^k] = 0$.

Inspired by the iterate analysis framework in we define the following virtual sequences:

$$\boldsymbol{u}_n^{k+1} = \boldsymbol{w}_n^k - \eta \widetilde{\boldsymbol{g}}_n^k, \tag{18}$$

$$\boldsymbol{w}_n^{k+1} = \begin{cases} \boldsymbol{u}_n^{k+1}, & ((k+1) \bmod H) \\ & \neq 0, \\ \boldsymbol{u}_n^{k'} - \sum_{n=1}^{N} p_n Q_{g,n}(\Delta_n^{k'}), & otherwise. \end{cases} \tag{19}$$

Here, $k' = k+1-H$ is the last synchronization step and $\Delta_n^{k'} = \boldsymbol{u}_n^{k'} - \boldsymbol{u}_n^{k+1}$ is the differences since the last synchronization. The following short-hand notation will be found useful in the convergence analysis of the proposed FL framework:

$$\boldsymbol{u}^k = \sum_{n=1}^{N} p_n \boldsymbol{u}_n^k, \quad \boldsymbol{w}^k = \sum_{n=1}^{N} p_n \boldsymbol{w}_n^k, \tag{20}$$

$$\widetilde{\boldsymbol{g}}^k = \sum_{n=1}^{N} p_n \widetilde{\boldsymbol{g}}_n^k, \quad \boldsymbol{g}^k = \sum_{n=1}^{N} p_n \nabla F_n(\boldsymbol{w}_n^k). \tag{21}$$

Thus, $\boldsymbol{u}^{k+1} = \boldsymbol{w}^k - \eta \widetilde{\boldsymbol{g}}^k$. Note that we can only obtain $\boldsymbol{w}^{k+1}$ when $((k+1) \bmod H) = 0$. Further, due to the unbiased gradient quantization scheme, Q_g, no matter whether $((k+1) \bmod H) = 0$ or $((k+1) \bmod H) \neq 0$, we always have $\mathbb{E}[\mathbb{E}_{Q_g}[\boldsymbol{w}^{k+1}]] = \mathbb{E}[\boldsymbol{u}^{k+1}]$.

B. Key Lemmas

Now, we give four important lemmas to convey our proof.
Lemma 1 (Bounding the Weight Quantization Error [8]):

$$\mathbb{E}_{Q_w}\left[\left\|\boldsymbol{r}_n^k\right\|_2^2\right] \leq \eta \sqrt{d}\delta_{w,n}\tau. \tag{22}$$

Lemma 2: According to the proposed algorithm the expected inner product between stochastic gradient and full batch gradient can be bounded with:

$$\mathbb{E}\left[\mathbb{E}_Q\left[\langle \nabla F(\boldsymbol{w}^k), \boldsymbol{u}^{k+1} - \boldsymbol{w}^k \rangle\right]\right]$$
$$\leq -\frac{\eta}{2}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2\right] - \frac{\eta}{2}\mathbb{E}\left[\left\|\boldsymbol{g}^k\right\|_2^2\right]$$
$$+ \frac{\eta L^2}{2}\sum_{n=1}^{N} p_n^2 \mathbb{E}\left[\mathbb{E}_Q\left[\left\|\boldsymbol{w}^k - \boldsymbol{w}_n^k\right\|_2^2\right]\right]. \tag{23}$$

Proof:

$$\mathbb{E}\left[\mathbb{E}_Q\left[\langle \nabla F(\boldsymbol{w}^k), \boldsymbol{u}^{k+1} - \boldsymbol{w}^k \rangle\right]\right]$$
$$= -\eta \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}^k), \mathbb{E}_Q[\widetilde{\boldsymbol{g}}^k] \rangle\right]$$
$$\overset{(a)}{=} -\eta \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}^k), \boldsymbol{g}^k \rangle\right]$$
$$\overset{(b)}{=} \frac{\eta}{2}\mathbb{E}\left[-\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2 - \left\|\boldsymbol{g}^k\right\|_2^2 + \left\|\nabla F(\boldsymbol{w}^k) - \boldsymbol{g}^k\right\|_2^2\right]$$
$$\overset{(c)}{\leq} -\frac{\eta}{2}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2\right] - \frac{\eta}{2}\mathbb{E}\left[\left\|\boldsymbol{g}^k\right\|_2^2\right]$$
$$+ \frac{\eta L^2}{2}\sum_{n=1}^{N} p_n^2 \mathbb{E}\left[\mathbb{E}_Q\left[\left\|\boldsymbol{w}^k - \boldsymbol{w}_n^k\right\|_2^2\right]\right] \tag{24}$$

where (a) is due to $\mathbb{E}[\widetilde{\boldsymbol{g}}_n^k] = \nabla F_n(\boldsymbol{w}_n^k)$ and $\mathbb{E}_Q[\widetilde{\boldsymbol{g}}^k] = \boldsymbol{g}^k$ and (b) is due to $-2\langle a, b \rangle = \|a - b\|^2 - \|a\|^2 - \|b\|^2$,

and (c) follows from L-smoothness assumption. The proof is completed. $\square$

Lemma 3: According to the proposed algorithm the expected inner product between stochastic gradient and full batch gradient can be bounded with:

$$\frac{L}{2}\mathbb{E}\left[\mathbb{E}_Q\left[\left\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\right\|_2^2\right]\right]$$
$$\leq \frac{\eta^2 L \bar{p}\sigma^2}{2M} + \frac{\eta^2 L}{2}\left\|\boldsymbol{g}^k\right\|_2^2 + \frac{\eta L}{2}\sqrt{d}\tau \sum_{n=1}^{N} p_n^2 \delta_{w,n}$$
$$+ \frac{L}{2}\sum_{n=1}^{N} p_n^2 \delta_{g,n}\mathbb{E}\left[\left\|\boldsymbol{u}_n^k - \boldsymbol{u}^{k'}\right\|_2^2\right]. \tag{25}$$

where $k' = k+1-H$ $(k' < k)$ is the nearest synchronization step.

Proof: According to the update rule defined in (19), when $(k+1) \bmod H = 0$, we have,

$$\boldsymbol{w}^{k+1} - \boldsymbol{w}^k$$
$$= \boldsymbol{w}^{k+1} - \boldsymbol{u}^{k+1} + \boldsymbol{u}^{k+1} - \boldsymbol{w}^k$$
$$= \sum_{n=1}^{N} p_n \left(\boldsymbol{u}_n^{k'} - \sum_{n=1}^{N} p_n Q_{g,n}(\boldsymbol{u}_n^{k'} - \boldsymbol{u}_n^{k+1})\right) - \boldsymbol{u}^{k+1}$$
$$+ \boldsymbol{u}^{k+1} - \boldsymbol{w}^k$$
$$= \sum_{n=1}^{N} p_n \left(\boldsymbol{u}_n^{k'} - \boldsymbol{u}_n^{k+1} - Q_{g,n}(\boldsymbol{u}_n^{k'} - \boldsymbol{u}_n^{k+1})\right)$$
$$+ (\boldsymbol{u}_n^{k+1} - \boldsymbol{w}_n^k)\Big)$$
$$\overset{(a)}{=} \sum_{n=1}^{N} p_n \left(\widetilde{\Delta}_n^{k'} - Q_{g,n}(\widetilde{\Delta}_n^{k'})\right) + \eta \widetilde{\boldsymbol{g}}^k. \tag{26}$$

Here, (a) is due to the short-hand notation defined in (20)-(21) and $k' = k+1-H$ is the last synchronization step. We then have,

$$\frac{L}{2}\mathbb{E}\left[\mathbb{E}_Q\left[\left\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\right\|_2^2\right]\right]$$
$$= \frac{L}{2}\mathbb{E}\left[\mathbb{E}_Q\left[\left\|\sum_{n=1}^{N} p_n \left(Q_{g,n}(\widetilde{\Delta}_n^{k'}) - \widetilde{\Delta}_n^{k'}\right)\right\|_2^2 + \left\|\eta\widetilde{\boldsymbol{g}}^k\right\|_2^2\right]\right]$$
$$\overset{(a)}{\leq} \frac{\eta^2 L}{2}\mathbb{E}\left[\left\|\widetilde{\boldsymbol{g}}^k\right\|_2^2\right] + \frac{L}{2}\sum_{n=1}^{N} p_n^2 \delta_{g,n}\mathbb{E}\left[\left\|\boldsymbol{u}_n^k - \boldsymbol{u}^{k'}\right\|_2^2\right], \tag{27}$$

where (a) is due to Definition 1. The first term in (27) can be bounded by

$$\frac{\eta^2 L}{2}\mathbb{E}\left[\left\|\sum_{n=1}^{N} p_n(\nabla \widetilde{f}_n(\boldsymbol{w}_n^k) - \boldsymbol{r}_n^k/\eta)\right\|_2^2\right]$$
$$= \frac{\eta^2 L}{2}\mathbb{E}\left[\left\|\sum_{n=1}^{N} p_n(\nabla \widetilde{f}_n(\boldsymbol{w}_n^k) - \nabla F_n(\boldsymbol{w}_n^k) + \nabla F_n(\boldsymbol{w}_n^k))\right\|_2^2\right]$$
$$+ \frac{L}{2}\sum_{n=1}^{N} p_n^2 \mathbb{E}\left[\left\|\boldsymbol{r}_n^k\right\|_2^2\right]$$
$$\leq \frac{\eta^2 L \bar{p}\sigma^2}{2M} + \frac{\eta^2 L}{2}\left\|\boldsymbol{g}^k\right\|_2^2 + \frac{\eta L}{2}\sqrt{d}\tau \sum_{n=1}^{N} p_n^2 \delta_{w,n}. \tag{28}$$

Replacing the first term in (27) with the result in (28), we have

$$\frac{L}{2}\mathbb{E}\left[\mathbb{E}_Q\left[\left\|\boldsymbol{w}^{k+1}-\boldsymbol{w}^k\right\|_2^2\right]\right]$$

$$\leq \frac{\eta^2 L\bar{p}\sigma^2}{2M}+\frac{\eta^2 L}{2}\left\|\boldsymbol{g}^k\right\|_2^2+\frac{\eta L}{2}\sqrt{d}\tau\sum_{n=1}^N p_n^2\delta_{w,n}$$

$$+\frac{L}{2}\sum_{n=1}^N p_n^2\delta_{g,n}\mathbb{E}\left[\left\|\boldsymbol{u}_n^k-\boldsymbol{u}^{k'}\right\|_2^2\right]. \tag{29}$$

The proof is completed. □

Lemma 4 (Bounding the Divergence): Suppose $1-3\eta^2 L^2 H^2 > 0$, *we have,*

$$\sum_{k=0}^{K-1}\sum_{n=1}^N p_n^2\mathbb{E}\left[\mathbb{E}_Q\left[\left\|\boldsymbol{w}^k-\boldsymbol{w}_n^k\right\|_2^2\right]\right]$$

$$\leq \frac{\eta^2 KH\bar{p}\sigma^2}{M(1-3\eta^2 L^2 H^2)}+\frac{\eta\sqrt{d}KH\tau\sum_{n=1}^N p_n^2\delta_{w,n}}{1-3\eta^2 L^2 H^2}$$

$$+\frac{3\eta^2 H^2\bar{p}}{1-3\eta^2 L^2 H^2}\sum_{i=0}^{K-1}\left\|\nabla F(\boldsymbol{w}^i)\right\|_2^2, \tag{30}$$

and

$$\sum_{n=1}^N p_n^2\delta_{g,n}\mathbb{E}\left[\left\|\boldsymbol{u}_n^k-\boldsymbol{u}^{k'}\right\|_2^2\right]$$

$$\leq \eta^2 H\frac{\bar{\delta}_g\sigma^2}{M}+3\eta^2\bar{\delta}_g H\sum_{i=k'}^{k'_H}\left\|\nabla F(\boldsymbol{w}^i)\right\|_2^2$$

$$+3\eta^2 L^2 H\sum_{i=k'}^{k'_H}A1_k+\eta\sqrt{d}H\sum_{n=1}^N p_n^2\delta_{g,n}\delta_{w,n}\tau. \tag{31}$$

where $\bar{p}=\sum_{n=1}^N p_n^2$ *and* $\bar{\delta}_g=\sum_{n=1}^N p_n^2\delta_{g,n}$.

Proof: Recalling that at the synchronization step where $(k'\mod H=0)$, $\boldsymbol{w}_n^{k'}=\boldsymbol{w}^{k'}$ for all $n\in\mathcal{N}$. Therefore, for any $k\geq 0$, such that $k'\leq k\leq k'+H$, we get,

$$A1_k := \sum_{n=1}^N p_n^2\mathbb{E}\left[\mathbb{E}_Q\left[\left\|\boldsymbol{w}^k-\boldsymbol{w}_n^k\right\|_2^2\right]\right]$$

$$=\sum_{n=1}^N p_n^2\mathbb{E}\left[\mathbb{E}_Q\left[\left\|(\boldsymbol{w}^k-\boldsymbol{w}^{k'})-(\boldsymbol{w}_n^k-\boldsymbol{w}^{k'})\right\|_2^2\right]\right]$$

$$\overset{(a)}{\leq}\sum_{n=1}^N p_n^2\mathbb{E}\left[\mathbb{E}_Q\left[\left\|\boldsymbol{w}_n^k-\boldsymbol{w}^{k'}\right\|_2^2\right]\right]$$

$$=\sum_{n=1}^N p_n^2\mathbb{E}\left[\left\|\sum_{i=k'}^k\eta\nabla\widetilde{f}_n(\boldsymbol{w}_n^i)-\sum_{i=k'}^k\boldsymbol{r}_n^i\right\|_2^2\right]$$

$$\overset{(b)}{=}\eta^2\sum_{n=1}^N p_n^2\mathbb{E}\left[\left\|\sum_{i=k'}^k\nabla\widetilde{f}_n(\boldsymbol{w}_n^i)\right\|_2^2\right]$$

$$+\sum_{n=1}^N p_n^2\sum_{i=k'}^k\mathbb{E}_Q\left[\left\|\boldsymbol{r}_n^i\right\|_2^2\right]$$

$$\leq \eta^2\sum_{n=1}^N p_n^2\mathbb{E}\left[\left\|\sum_{i=k'}^{k'_H}\nabla\widetilde{f}_n(\boldsymbol{w}_n^i)\right\|_2^2\right]$$

$$+H\eta\sqrt{d}\sum_{n=1}^N p_n^2\delta_{w,n}\tau, \tag{32}$$

where $k'_H=k'+H-1$, (a) results from $\sum_{n=1}^N p_n(\boldsymbol{w}_i^k-\boldsymbol{w}_i^{k'})=\boldsymbol{w}^k-\boldsymbol{w}^{k'}$, $\boldsymbol{w}_i^{k'}=\boldsymbol{w}^{r'}$, and $\mathbb{E}\|\boldsymbol{x}-\mathbb{E}[\boldsymbol{x}]\|_2^2\leq\mathbb{E}\|\boldsymbol{x}\|_2^2$. (b) holds due to the unbiased quantization scheme, i.e., $\mathbb{E}_Q\left[\boldsymbol{r}_n^k\right]=0$.

We generalize the result from [32] to upper-bound the first term in RHS of (32), (see the of Theorem 3 and its proof in appendix for the special case of $p_n=\frac{1}{N}$):

$$\eta^2\sum_{n=1}^N p_n^2\mathbb{E}\left[\left\|\sum_{i=k'}^{k'_H}(\nabla\widetilde{f}_n(\boldsymbol{w}_n^i)-\nabla F_n(\boldsymbol{w}_n^i)+\nabla F_n(\boldsymbol{w}_n^i))\right\|_2^2\right]$$

$$\leq \eta^2 H\frac{\bar{p}\sigma^2}{M}+3\eta^2 L^2 H\sum_{i=k'}^{k'_H}\sum_{n=1}^N p_n^2\left\|\boldsymbol{w}^i-\boldsymbol{w}_n^i\right\|_2^2$$

$$+3\eta^2\bar{p}H^2 G^2+3\eta^2\bar{p}H\sum_{i=k'}^{k'_H}\left\|\nabla F(\boldsymbol{w}^i)\right\|_2^2, \tag{33}$$

where $\bar{p}=\sum_{n=1}^N p_n^2$. It follows that

$$\sum_{i=0}^{K-1}A1_i \leq \eta^2 KH\bar{p}(\frac{\sigma^2}{M}+3HG^2)+KH\eta\sqrt{d}\tau\sum_{n=1}^N p_n^2\delta_{w,n}$$

$$+3\eta^2 H^2\sum_{i=0}^{K-1}\left(L^2 A1_i+\bar{p}\left\|\nabla F(\boldsymbol{w}^i)\right\|_2^2\right). \tag{34}$$

Suppose $1-3\eta^2 L^2 H^2 > 0$, we have

$$\sum_{i=0}^{K-1}A1_i \leq \frac{\eta^2 KH\bar{p}(\sigma^2/M+3HG^2)}{(1-3\eta^2 L^2 H^2)}$$

$$\times\frac{\eta\sqrt{d}KH\tau\sum_{n=1}^N p_n^2\delta_{w,n}}{1-3\eta^2 L^2 H^2}$$

$$+\frac{3\eta^2 H^2\bar{p}}{1-3\eta^2 L^2 H^2}\sum_{i=0}^{K-1}\left\|\nabla F(\boldsymbol{w}^i)\right\|_2^2, \tag{35}$$

Similarly, we denote $A2_k := \sum_{n=1}^N p_n^2\delta_{g,n}\mathbb{E}\left[\left\|\boldsymbol{u}_n^k-\boldsymbol{u}^{k'}\right\|_2^2\right]$, and have

$$A2_k \leq \eta^2\sum_{n=1}^N p_n^2\delta_{g,n}\mathbb{E}\left[\left\|\sum_{i=k'}^{k'_H}\nabla\widetilde{f}_n(\boldsymbol{w}_n^i)\right\|_2^2\right]$$

$$+\eta\sqrt{d}H\sum_{n=1}^N p_n^2\delta_{g,n}\delta_{w,n}\tau$$

$$\leq \eta^2 H\frac{\bar{\delta}_g\sigma^2}{M}+3\eta^2\bar{\delta}_g H\sum_{i=k'}^{k'_H}\left\|\nabla F(\boldsymbol{w}^i)\right\|_2^2$$

$$+ 3\eta^2 L^2 H \sum_{i=k'}^{k'_H} A1_k + \eta\sqrt{d}H \sum_{n=1}^{N} p_n^2 \delta_{g,n}\delta_{w,n}\tau.$$

$$(36)$$

and the proof is completed.   □

C. Main Results

Under the L-smooth assumption of F, we have,

$$\mathbb{E}\left[F(\boldsymbol{w}^{k+1}) - F(\boldsymbol{w}^k)\right]$$

$$\leq \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}^k), \boldsymbol{w}^{k+1} - \boldsymbol{w}^k\rangle\right] + \frac{L}{2}\mathbb{E}\left[\left\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\right\|_2^2\right]$$

$$= \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}^k), \boldsymbol{w}^{k+1} - \boldsymbol{u}^{k+1} + \boldsymbol{u}^{k+1} - \boldsymbol{w}^k\rangle\right]$$

$$\quad + \frac{L}{2}\mathbb{E}\left[\left\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\right\|_2^2\right]$$

$$\leq \mathbb{E}[\langle \nabla F(\boldsymbol{w}^k), \boldsymbol{u}^{k+1} - \boldsymbol{w}^k\rangle + \frac{L}{2}\left\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\right\|_2^2]. \quad (37)$$

We use Lemma 1-4 to upper bound the RHS of (37) and set $\eta L \leq 1$, which gets,

$$\mathbb{E}\left[F(\boldsymbol{w}^{k+1}) - F(\boldsymbol{w}^k)\right]$$

$$\leq -\frac{\eta}{2}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2\right] + \frac{\eta L^2}{2}A1_k + \frac{L}{2}A2_k + \frac{\eta^2 L\sigma^2}{2M}$$

$$\quad + \frac{\eta L}{2}\sqrt{d}\tau \sum_{n=1}^{N} p_n^2 \delta_{w,n} - \frac{\eta}{2}(1-\eta L)\left\|\boldsymbol{g}^k\right\|_2^2$$

$$\overset{(a)}{\leq} -\frac{\eta}{2}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2\right] + \frac{\eta L^2}{2}A1_k + \frac{L}{2}A2_k + \frac{\eta^2 L\sigma^2}{2M}$$

$$\quad + \frac{\eta L}{2}\sqrt{d}\tau \sum_{n=1}^{N} p_n^2 \delta_{w,n}$$

$$\overset{(b)}{\leq} -\frac{\eta}{2}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2\right] + \frac{3\eta^2 L\bar{\delta}_g}{2}H \sum_{i=k'}^{k'_H} \mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^i)\right\|_2^2\right]$$

$$\quad + \frac{\eta^2 L\sigma^2}{2M}(\bar{\delta}_g H + \bar{p}) + \frac{\eta L\sqrt{d}\tau}{2}H \sum_{n=1}^{N} p_n^2(\delta_{g,n}+1)\delta_{w,n}$$

$$\quad + \frac{\eta L^2}{2}A1_k + 3\eta^2 L^3 H \sum_{i=k'}^{k'_H} A1_k. \quad (38)$$

where (a) holds due to the setting, i.e., $\eta L \leq 1$, and (b) follows the second result in Lemma 4.

Summing up for all K communication rounds and rearranging the terms gives,

$$\mathbb{E}\left[F(\boldsymbol{w}^K) - F(\boldsymbol{w}^0)\right]$$

$$\leq -\frac{\eta C_1}{2}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2\right] + \frac{\eta L^2 C_2}{2}\sum_{k=0}^{K-1} A1_k$$

$$\quad + \frac{\eta^2 L\sigma^2}{2M}(\bar{\delta}_g H + \bar{p}) + \frac{\eta L\sqrt{d}\tau}{2}H \sum_{n=1}^{N} p_n^2(\delta_{g,n}+1)\delta_{w,n},$$

$$(39)$$

where $C_1 = 1 - 3\eta LH\bar{\delta}_g$ and $C_2 = 1 + 3\eta LH$.

Plugging Lemma 4 into (39), if $C_1' = C_1 - \frac{3\eta^2 H^2\bar{p}(L^2+3\eta L^3 H)}{1-3\eta^2 L^2 H^2} \geq 0$, we have,

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2\right]$$

$$\leq \frac{2\mathbb{E}\left[F(\boldsymbol{w}^0) - F(\boldsymbol{w}^K)\right]}{\eta C_1' K} + \frac{\eta^2 L^2 C_2 H\bar{p}\sigma^2}{C_1' M(1-3\eta^2 L^2 H^2)}$$

$$\quad + \frac{\eta L^2 C_2 H\sqrt{d}\tau\bar{\delta}_w}{C_1'(1-3\eta^2 L^2 H^2)} + \frac{L\sqrt{d}\tau H}{C_1'}\sum_{n=1}^{N} p_n^2(\delta_{g,n}+1)\delta_{w,n}$$

$$\quad + \frac{3\eta^2 L^2 C_2 H^2 G^2}{C_1'(1-3\eta^2 L^2 H^2)} + \frac{\eta L\sigma^2(\bar{\delta}_g H + \bar{p})}{C_1' M}, \quad (40)$$

where $\bar{\delta}_w = u\sum_{n=1}^{N} p_n^2 \delta_{w,n}$.

If we set $\eta = \sqrt{MN/K}$ and

$$\eta LH\delta_g \geq \frac{\eta^2 H^2\bar{p}(L^2+3\eta L^3 H)}{1-3\eta^2 L^2 H^2}, \quad (41)$$

we can get the $1/C_1' \leq 2$. Thus,

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla F(\boldsymbol{w}^k)\right\|_2^2\right]$$

$$\leq \frac{4\mathbb{E}\left[F(\boldsymbol{w}^0) - F(\boldsymbol{w}^K)\right]}{\sqrt{MNK}} + \frac{2\,L\sigma^2(2H\bar{\delta}_g + p)}{\sqrt{MNK}}$$

$$\quad + \frac{12MLH\bar{\delta}_g G^2}{\sqrt{MNK}} + 2L\sqrt{d}\tau \sum_{n=1}^{N} p_n^2(\delta_{g,n}+1)\delta_{w,n}. \quad (42)$$

and the proof is completed.

REFERENCES

[1] Apple. (May 2021). *Coreml on Device Training*. [Online]. Available: https://developer.apple.com/machine-learning/core-ml

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[3] A. Hard et al., "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[4] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *Int. J. Med. Inform.*, vol. 112, pp. 59–67, Apr. 2018.

[5] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23920–23935, 2020.

[6] Y. Fu et al., "CPT: Efficient deep neural network training via cyclic precision," in *Proc. Int. Conf. Learn. Represent.*, May 2021, pp. 1–14.

[7] D. Zhang, J. Yang, D. Ye, and G. Hua, "LQ-Nets: Learned quantization for highly accurate and compact deep neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 365–382.

[8] H. Li, S. De, Z. Xu, C. Studer, H. Samet, and T. Goldstein, "Training quantized nets: A deeper understanding," in *Proc. Int. Conf. Mach. Learn. Workshop Principled Approaches Deep Learn.*, Long Beach, CA, USA, Aug. 2017, pp. 5811–5821.

[9] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, Apr. 2019, pp. 1387–1395.

[10] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.

[11] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive MIMO for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[12] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[13] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "SignSGD: Compressed optimisation for non-convex problems," in *Proc. 35th Int. Conf. Mach. Learn.*, Jul. 2018, pp. 560–569.

[14] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 1709–1720.

[15] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.

[16] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2021, pp. 1–10.

[17] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2021, pp. 1–10.

[18] D. Shi, L. Li, R. Chen, P. Prakash, M. Pan, and Y. Fang, "Towards energy efficient federated learning over 5G+ mobile devices," 2021, *arXiv:2101.04866*.

[19] F. Fu et al., "Don't waste your bits! squeeze activations and gradients for deep neural networks via tinyscript," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2020, pp. 3304–3314.

[20] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2019, pp. 14695–14706.

[21] L. Qu, S. Song, and C.-Y. Tsui, "FedDQ: Communication-efficient federated learning with descending quantization," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Sicily, Italy, Dec. 2022, pp. 2021–2031.

[22] *Technical Specification Group Services and System Aspects; Release 15 Description; Summary of Rel-15 Work Items*, document 21.915, Version 2.0.0, 3GPP, Sep. 2019. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/Specification Details.aspx?specificationId=3389

[23] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, 2018.

[24] D. Yin, A. Pananjady, M. Lam, D. Papailiopoulos, K. Ramchandran, and P. Bartlett, "Gradient diversity: A key ingredient for scalable distributed learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, Lanzarote, Canary Islands, 2018, pp. 1998–2007.

[25] S. Hong and H. Kim, "An integrated GPU power and performance model," in *Proc. 37th Annu. Int. Symp. Comput. Archit.*, New York, NY, USA, Jun. 2010, pp. 280–289.

[26] NVIDIA. *Deep Learning Performance Documentation*. Accessed: Apr. 6, 2022. [Online]. Available: https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html

[27] A. Haidar, S. Tomov, J. Dongarra, and N. J. Higham, "Harnessing GPU tensor cores for fast FP16 arithmetic to speed up mixed-precision iterative refinement solvers," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2018, pp. 603–613.

[28] T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze, "A method to estimate the energy consumption of deep neural networks," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2017, pp. 1916–1920.

[29] R. Bonghi. (May 2021). *Interactive System-Monitor and Process Viewer for All Nvidia Jetson*. [Online]. Available: https://github.com/topics/jetson-stats

[30] S. Khirirat, S. Magnusson, A. Aytekin, and M. Johansson, "A flexible framework for communication-efficient machine learning: From HPC to IoT," 2020, *arXiv:2003.06377*.

[31] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Represent.*, New Orleans, Louisiana, May 2019, pp. 1–26.

[32] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 2525–2536.

[33] M. Chiang, *Geometric Programming for Communication Systems*. Norwell, MA, USA: Now Publishers, 2005.

[34] S. J. Wright, *Primal-Dual Interior-Point Methods*. Philadelphia, PA, USA: SIAM, 1997.

[35] K. Bonawitz et al., "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*.

[36] I. Fette and A. Melnikov. *The Websocket Protocol RFC 6455*. Accessed: Apr. 4, 2021. [Online]. Available: https://www.hjp.at/doc/rfc/rfc6455.html

[37] D. J. Beutel et al., "Flower: A friendly federated learning research framework," 2020, *arXiv:2007.14390*.

[38] NVIDIA. *Cuda Toolkit Documentation*. Accessed: Aug. 31, 2022. [Online]. Available: https://docs.nvidia.com/cuda/profiler-users-guide/index.html#profiling-controls

[39] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 1205–1221, Jun. 2019.

[40] Xilinx. *Zynq Ultrascale+ Mpsoc*. (Dec. 2021). [Online]. Available: https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

Rui Chen (Member, IEEE) received the B.S. degree from the Marine Electrical Engineering College, Dalian Maritime University, Dalian, China, in 2018. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA. Her major research interests include federated learning, data-driven optimization, and differential privacy.

Dian Shi (Student Member, IEEE) received the B.S. degree in electronic information engineering from the University of Electronic Science and Technology of China in 2017 and the Ph.D. degree in electrical engineering from the University of Houston in 2022. His research interests include deep reinforcement learning, federated learning, game theory, and edge intelligence.

Xiaoqi Qin (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from Virginia Tech. She is currently an Associate Professor with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunication (BUPT). Her research focuses on exploring performance limits of next-generation wireless networks and developing innovative solutions for intelligent and efficient machine-type communications.

Dongjie Liu received the M.S. degree from the Nanjing University of Posts and Telecommunication (NJUPT). He is currently a Senior Engineer of wireless communication system with the Purple Mountain Laboratories (PML). His work focuses on research and developing of next-generation wireless networks and developing innovative platform solutions for intelligent communication.

Miao Pan (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the Dalian University of Technology, China, in 2004, the M.A.Sc. degree in electrical and computer engineering from the Beijing University of Posts and Telecommunications, China, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Florida in 2012. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Houston. His research interests include wireless/AI for AI/wireless, deep learning privacy, cybersecurity, and underwater communications and networking. He is a member of AAAI and ACM. He was a recipient of NSF CAREER Award in 2014. His work won IEEE TCGCC Best Conference Paper Award 2019 and the Best Paper Award in IEEE ICC 2019, IEEE VTC 2018, IEEE GLOBECOM 2017, and IEEE GLOBECOM 2015. He has been serving as a Technical Organizing Committee for several conferences such as the TPC Co-Chair for Mobiquitous 2019 and ACM WUWNet 2019. He is an Editor for IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY and an Associate Editor for IEEE INTERNET OF THINGS JOURNAL (IoT).

Shuguang Cui (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, California, USA, in 2005. Afterwards, he has been working as an Assistant, an Associate, the full Chair Professor in electrical and computer engineering at the University of Arizona, Texas A&M University, UC Davis, and CUHK, Shenzhen. He has also served as the Executive Dean for the School of Science and Engineering. He is currently the Director for Future Network of Intelligence Institute (FNii) at CUHK, Shenzhen, and the Executive Vice Director with the Shenzhen Research Institute of Big Data. His current research interests focus on data driven large-scale system control and resource management, large data set analysis, the IoT system design, energy harvesting based communication system design, and cognitive network optimization. He has been an Elected Member for IEEE Signal Processing Society SPCOM Technical Committee (2009–2014) and the Elected Chair for IEEE ComSoc Wireless Technical Committee (2017–2018). He was elected as an IEEE ComSoc Distinguished Lecturer in 2014 and IEEE VT Society Distinguished Lecturer in 2019. In 2022, he got elected as the EiC for IEEE TRANSACTIONS ON MOBILE COMPUTING. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the Worlds' Most Influential Scientific Minds by Science Watch in 2014. He was a recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and the TPC co-chair for many IEEE conferences. He has also been serving as the Area Editor for *IEEE Signal Processing Magazine* and an Associate Editors for IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Series on *Green Communications and Networking*, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.