Green, Quantized Federated Learning over Wireless Networks: An Energy-Efficient Design

Minsu Kim, Walid Saad, *Fellow, IEEE*, Mohammad Mozaffari, *Member IEEE*, and Merouane Debbah, *Fellow, IEEE*

***Abstract*—The practical deployment of federated learning (FL) over wireless networks requires balancing energy efficiency, convergence rate, and a target accuracy due to the limited available resources of devices. Prior art on FL often trains deep neural networks (DNNs) to achieve high accuracy and fast convergence using 32 bits of precision level. However, such scenarios will be impractical for resource-constrained devices since DNNs typically have high computational complexity and memory requirements. Thus, there is a need to reduce the precision level in DNNs to reduce the energy expenditure. In this paper, a green-quantized FL framework, which represents data with a finite precision level in both local training and uplink transmission, is proposed. Here, the finite precision level is captured through the use of quantized neural networks (QNNs) that quantize weights and activations in fixed-precision format. In the considered FL model, each device trains its QNN and transmits a quantized training result to the base station. Energy models for the local training and the transmission with quantization are rigorously derived. To minimize the energy consumption and the number of communication rounds simultaneously, a multi-objective optimization problem is formulated with respect to the number of local iterations, the number of selected devices, and the precision levels for both local training and transmission while ensuring convergence under a target accuracy constraint. To solve this problem, the convergence rate of the proposed FL system is analytically derived with respect to the system control variables. Then, the Pareto boundary of the problem is characterized to provide efficient solutions using the normal boundary inspection method. Design insights on balancing the tradeoff between the two objectives while achieving a target accuracy are drawn from using the Nash bargaining solution and analyzing the derived convergence rate. Simulation results show that the proposed FL framework can reduce energy consumption until convergence by up to 70% compared to a baseline FL algorithm that represents data with full precision without damaging the convergence rate.**

I. Introduction

Federated learning (FL) is an emerging paradigm that enables distributed learning among wireless devices [2]. In FL, a central server (e.g., a base station (BS)) and multiple mobile devices collaborate to train a shared machine learning model without sharing raw data. Many FL works employ deep neural networks (DNNs), whose size constantly grows to match the increasing demand for higher accuracy [3]. Such DNN architectures can have tens of millions of parameters and billions of multiply-accumulate (MAC) operations. Moreover, to achieve fast convergence, these networks typically represent data in 32 bits of full precision level, which may consume significant energy due to high computational complexity and memory requirements [4]. Additionally, a large DNN can induce a significant communication overhead [5]. Under such practical constraints, it may be challenging to deploy FL over resource-constrained Internet of Things (IoT) devices due to its large energy cost. To design an energy-efficient, green FL scheme, one can reduce the precision level to decrease the energy consumption during the local training and communication phase. However, a low precision level can jeopardize the convergence rate by introducing quantization errors. Therefore, finding the optimal precision level that balances energy efficiency and convergence rate while meeting desired FL accuracy constraints will be a major challenge for the practical deployment of green FL over wireless networks.

Several works have studied the energy efficiency of FL from a system-level perspective [6]–[11]. The work in [6] investigated the energy efficiency of FL algorithms in terms of the carbon footprint compared to centralized learning. In [7], the authors formulated a joint minimization problem for energy consumption and training time by optimizing heterogeneous computing and wireless resources. The work in [8] developed an approach to minimize the total energy consumption by controlling a target accuracy during local training based on a derived convergence rate. The authors in [9] proposed a sum energy minimization problem by considering joint bandwidth and workload allocation of heterogeneous devices. In [10], the authors studied a joint optimization problem to minimize the energy and the training time under a target accuracy. The work in [11] developed a resource management scheme by leveraging the information of loss functions of each device to maximize the accuracy under constrained communication and computation resources. However, these works [6]–[11] did not consider the energy efficiency of their DNN structure during training. Since mobile devices have limited computing and memory resources, deploying an energy-efficient DNN will be necessary for green FL.

To further improve FL energy efficiency, model compression methods such as quantization were studied in [12]–[15]. In [12], the authors developed an over-the-air FL system that uses one-bit gradient quantization aggregation scheme. The authors in [13] developed an approach to minimize the training time by optimizing transmission precision level and bandwidth allocation. The work in [14] proposed an approach to minimize

M. Kim and W. Saad are with the Wireless@VT Group, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA, USA (email: {msukim, walids}@vt.edu.)

M. Mozaffari is with Ericsson Research, Santa Clara, CA, USA (email: mohammad.mozaffari@ericsson.com).

M. Debbah is with Khalifa University of Science and Technology, Abu Dhabi, UAE (email: merouane.debbah@ku.ac.ae).

A preliminary version of this work was presented at IEEE ICC 2022 [1].

This research was supported by the U.S. National Science Foundation under Grant CNS-2114267.

The implementation code is available on https://github.com/news-vt the energy consumption and the loss function by optimizing model compression design for uplink transmission and device selection strategy. In [15], the authors studied an energy minimization problem by controlling local iterations, bandwidth allocation, and precision level for both local training and transmission under full device participation scheme. However, the works in [12]–[14] only considered the communication efficiency while there can be a large energy consumption in local training due to high precision level. Although the work in [15] considered quantization for both local training and transmission, it used full device participation scheme, which is not practical due to stragglers, and only the energy consumption is minimized. In our previous work [1], an energy minimization problem was formulated to investigate the tradeoff between energy, precision, and accuracy. However, the same precision level was used for local training and transmission as done in [15]. As such, the results of [1] cannot be directly applied for more general cases such as those with heterogeneous devices and non-i.i.d datasets. Moreover, the number of local iterations and the number of selected devices were not jointly optimized. To the best of our knowledge, there are no current works that jointly consider the tradeoff between energy efficiency, convergence rate, and accuracy while simultaneously controlling local iterations, the number of scheduled devices, and precision levels in local training and transmission for green FL over wireless networks.

The main contribution of this paper is a novel green, energy-efficient quantized FL framework that can represent data with a finite precision level in both local training and uplink transmission. Our contributions include:

- We propose an FL framework that takes into account stochastic quantization in both local training and transmission with different precision levels. All devices train their quantized neural networks (QNNs), whose weights and activations are quantized with a finite precision level, so as to decrease energy consumption for computation and memory access. In uplink communication, each device performs quantization to its training result to improve the communication efficiency.
- To quantify the energy consumption, we propose a rigorous energy model for the local training based on the physical structure of a processing chip. We also derive the energy model for the uplink transmission with quantization. Although a low precision level can save the energy consumption per iteration, it decreases the convergence rate because of quantization errors. Thus, there is a need for a new approach to analyze the tradeoff between energy efficiency, convergence rate, and target accuracy by optimizing the precision levels. To this end, we formulate a novel multi-objective optimization problem by controlling the precision levels to jointly minimize the total energy consumption and the number of communication rounds while ensuring convergence with a target accuracy. We also incorporate two additional control variables: the number of local iterations and the number of selected devices at each communication round, which have a significant impact on both the energy consumption and the convergence time.

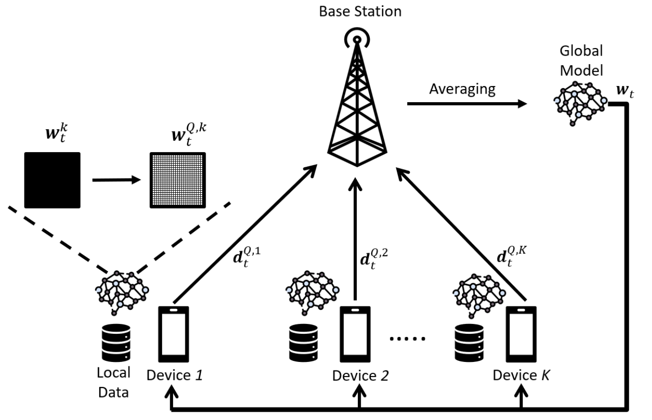


Fig. 1: An illustration of the quantized FL model over wireless network.

- To solve this problem, we first analytically derive the convergence rate of our FL framework with respect to the control variables under non-iid data distribution. We then optimize sampling probabilities for devices based on the derived convergence rate. Subsequently, we use the normal boundary inspection (NBI) method to obtain the Pareto boundary of our multi-objective optimization problem. To balance the tradeoff between the two objectives, we present and analyze two practical operating points: the Nash bargaining solution (NBS) and the sum minimizing solution (SUM) points.
- Based on the aforementioned operating points and the derived convergence rate, we provide design insights into the proposed FL framework. For instance, the total energy consumption initially decreases as the precision levels increase, however, after a certain threshold, a higher precision will induce higher energy costs. Meanwhile, the convergence rate will always improve with a higher precision. However, this improvement becomes negligible after a certain level. We also show that we need a higher precision level to achieve higher target accuracy at the expense of more energy and communication rounds. We then provide the impacts of system parameters such as the number of devices and model size on the performance of the proposed FL.

Simulation results show that our FL model can reduce the energy consumption around 70% compared to FedAvg without damaging the convergence rate.

The rest of this paper is organized as follows. Section II presents the system model. In Section III, we describe the studied problem. Section IV provides simulation results. Finally, conclusions are drawn in Section V.

II. System Model

Consider an FL system having N devices connected to a BS as shown in Fig. 1. Each device k has its own local dataset $\mathcal{D}_k = \{\boldsymbol{x}_{kl}, \boldsymbol{y}_{kl}\}$, where $l = 1, \ldots, D_k$. For example, $\{\boldsymbol{x}_{kl}, \boldsymbol{y}_{kl}\}$ can be an input-output pair for image classification, where $\boldsymbol{x}_{kl}$ is an input vector and $\boldsymbol{y}_{kl}$ is the corre-

TABLE I: List of notations.

Notation	Description	Notation	Description
N	Number of devices	P_k^{tx}	Transmission power
$(\boldsymbol{x}_{kl}, \boldsymbol{y}_{kl})$	Data sample	h_k	Average channel gain
D_k	Dataset size	N_0	Power spectral density of noise
$\boldsymbol{w}^k$	Model parameters	$E^{UL,k}(m)$	Energy consumption for uplink transmission
$F_k(\cdot)$	Local loss function	L	Smoothness parameter
I	Number of local iterations	μ	Convexity parameter
K	Number of sampled devices	Γ	Degree of non-iidness
m	Precision level for transmission	G	Bound of the norm of stochastic gradients
n	Precision level for local training	σ	Bound of the variance of stochastic gradients
ϵ	Target accuracy	d	Number of model parameters
$E^{C,k}(n)$	Energy consumption for one local iteration	N_c	Number of MAC operations
B	Allocated bandwidth	O_s	Number of neurons

sponding output. We define a loss function $f(\boldsymbol{w}^k, \boldsymbol{x}_{kl}, \boldsymbol{y}_{kl})$ to quantify the performance of a machine learning (ML) model with parameters $\boldsymbol{w}^k \in \mathbb{R}^d$ over $\{\boldsymbol{x}_{kl}, \boldsymbol{y}_{kl}\}$, where d is the number of parameters. Since device k has D_k data samples, its local loss function can be given by $F_k(\boldsymbol{w}^k) = \frac{1}{D_k}\sum_{l=1}^{D_k} f(\boldsymbol{w}^k, \boldsymbol{x}_{kl}, \boldsymbol{y}_{kl})$. The FL process aims to find the global parameters $\boldsymbol{w}$ that can solve the following optimization problem:

$$\min_{\boldsymbol{w}^1,\dots,\boldsymbol{w}^N} F(\boldsymbol{w}) = \sum_{k=1}^{N} \frac{D_k}{D} F_k(\boldsymbol{w}^k) = \frac{1}{D}\sum_{k=1}^{N}\sum_{l=1}^{D_k} f(\boldsymbol{w}^k, \boldsymbol{x}_{kl}, \boldsymbol{y}_{kl}) \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{w}^1 = \boldsymbol{w}^2 = \cdots = \boldsymbol{w}^N = \boldsymbol{w}, \tag{2}$$

where $D = \sum_{k=1}^{N} D_k$ is the total size of the entire dataset $\mathcal{D} = \cup_{k=1}^{N} \mathcal{D}_k$. Without loss of generality, we assume datasets across devices are non-iid.

Solving problem (2) typically requires an iterative process between the BS and devices. However, in practical systems, such as IoT systems, these devices are resource-constrained, particularly when it comes to computing and energy. Hence, we propose to manage the precision level of parameters used in our FL algorithm to reduce the energy consumption for computation, memory access, and transmission. As such, we adopt a QNN architecture whose weights and activations are quantized in fixed-point format rather than conventional 32-bit floating-point format [16]. During the training time, a QNN can reduce the energy consumption for MAC operation and memory access due to quantized weights and activations.

A. Quantized Neural Networks

In our model, each device trains a QNN of identical structure using n bits for quantization. High precision can be achieved if we increase n at the cost of more energy usage. We can represent any given number in a fixed-point format such as $[\Omega.\Phi]$, where Ω is the integer part and Φ is the fractional part of the given number [17]. Here, we use one bit to represent the integer part and $(n-1)$ bits for the fractional part. Then, the smallest positive number that we can present is $\kappa = 2^{-n+1}$, and the possible range of numbers with n bits will be $[-1, 1-2^{-n+1}]$. Note that a QNN restricts the value of weights to [-1, 1]. Otherwise, weights can be very large without meaningful impact on the performance. We consider a stochastic quantization scheme [17] since it generally performs better than deterministic quantization [18]. Any given number $w \in \boldsymbol{w}$ can be stochastically quantized as follows:

$$Q(w) = \begin{cases} \lfloor w \rfloor, & \text{with probability } \frac{\lfloor w \rfloor + \kappa - w}{\kappa}, \\ \lfloor w \rfloor + \kappa, & \text{with probability } \frac{w - \lfloor w \rfloor}{\kappa}, \end{cases} \tag{3}$$

where $\lfloor w \rfloor$ is the largest integer multiple of κ less than or equal to w. In the following lemma, we analyze the features of the stochastic quantization.

Lemma 1. *For the stochastic quantization $Q(\cdot)$, a scalar w, and a vector $\boldsymbol{w} \in \mathbb{R}^d$, we have*

$$\mathbb{E}[Q(w)] = w, \quad \mathbb{E}[(Q(w) - w)^2] \leq \frac{1}{2^{2n}}, \tag{4}$$

$$\mathbb{E}[Q(\boldsymbol{w})] = \boldsymbol{w}, \quad \mathbb{E}[||Q(\boldsymbol{w}) - \boldsymbol{w}||^2] \leq \frac{d}{2^{2n}}. \tag{5}$$

Proof. We first derive $\mathbb{E}[Q(w)]$ as

$$\mathbb{E}[Q(w)] = \lfloor w \rfloor \frac{\lfloor w \rfloor + \kappa - w}{\kappa} + (\lfloor w \rfloor + \kappa)\frac{w - \lfloor w \rfloor}{\kappa} = w. \tag{6}$$

Similarly, $\mathbb{E}[(Q(w) - w)^2]$ can be obtained as

$$\begin{aligned}\mathbb{E}[(Q(w)-w)^2] &= (\lfloor w \rfloor - w)^2 \frac{\lfloor w \rfloor + \kappa - w}{\kappa} + (\lfloor w \rfloor + \kappa - w)^2 \frac{w - \lfloor w \rfloor}{\kappa} \\ &= (w - \lfloor w \rfloor)(\lfloor w \rfloor + \kappa - w) \leq \frac{\kappa^2}{4} = \frac{1}{2^{2n}},\end{aligned} \tag{7}$$

where (7) follows from the arithmetic mean and geometric mean inequality. Since expectation is a linear operator, we have $\mathbb{E}[Q(\boldsymbol{w})] = \boldsymbol{w}$ from (6). From the definition of the square norm, $\mathbb{E}[||Q(\boldsymbol{w}) - \boldsymbol{w}||^2]$ can obtained as

$$\mathbb{E}[||Q(\boldsymbol{w}) - \boldsymbol{w}||^2] = \sum_{j=1}^{d} \mathbb{E}[(Q(w_j) - w_j)^2] \leq \frac{d}{2^{2n}}. \tag{8}$$

□

From Lemma 1, we can see that our quantization scheme is unbiased as its expectation is zero. However, the quantization error can still increase for a large model.

For device k, we denote the quantized weights of layer l as $\boldsymbol{w}_{(l)}^{Q,k} = Q(\boldsymbol{w}_{(l)}^{k})$, where $\boldsymbol{w}_{(l)}^{k}$ is the parameters of layer l. Then, the output of layer l will be: $o_{(l)} = g_{(l)}(\boldsymbol{w}_{(l)}^{Q,k}, o_{(l-1)})$, where $o_{(l-1)}$ is the output from the previous layer $l-1$, and $g(\cdot)$ is the operation of layer l on the input, including the linear sum of $\boldsymbol{w}_{(l)}^{Q,k}$ and $o_{(l-1)}$, batch normalization, and activation.

Note that our activation includes the stochastic quantization after a normal activation function such as ReLU. Then, the output of layer l, i.e., $o_{(l)}$, is fed into the next layer as an input. For training, we use the stochastic gradient descent (SGD) algorithm as follows

$$\boldsymbol{w}_{\tau+1}^{k} \leftarrow \boldsymbol{w}_{\tau}^{k} - \eta \nabla F_k(\boldsymbol{w}_{\tau}^{Q,k}, \xi_{\tau}^{k}), \tag{9}$$

where $\tau = 1 \dots I$ is training iteration, η is the learning rate, and ξ is a sample from D_k for the current update. The update of weights is done in full precision so that stochastic gradient (SG) noise can be averaged out properly [16]. Then, we restrict the values of $\boldsymbol{w}_{\tau+1}^{k}$ to $[-1, 1]$ as $\boldsymbol{w}_{\tau+1}^{k} \leftarrow \text{clip}(\boldsymbol{w}_{\tau+1}^{k}, -1, 1)$ where $\text{clip}(\cdot, -1, 1)$ projects an input to 1 if it is larger than 1, and projects an input to -1 if it is smaller than -1. Otherwise, it returns the same value as the input. Otherwise, $\boldsymbol{w}_{\tau+1}^{k}$ can become significantly large without a meaningful impact on quantization [16]. After each training, $\boldsymbol{w}_{\tau+1}^{k}$ will be quantized as $\boldsymbol{w}_{\tau+1}^{Q,k}$ for the forward propagation.

B. FL model

For learning, without loss of generality, we adopt FedAvg [4] to solve problem (2). At each communication round t, the BS selects K devices according to probability p_k for device k such that $\sum_{k=1}^{N} p_k = 1$, and we denote the sampled set as $\mathcal{N}_t$. The BS transmits the current global model $\boldsymbol{w}_t$ to the scheduled devices. Each device in $\mathcal{N}_t$ trains its local model based on the received global model by running I steps of SGD as below

$$\boldsymbol{w}_{t,\tau}^{k} = \boldsymbol{w}_{t,\tau-1}^{k} - \eta_t \nabla F_k(\boldsymbol{w}_{t,\tau-1}^{Q,k}, \xi_{\tau}^{k}), \forall \tau = 1, \dots, I, \tag{10}$$

where η_t is the learning rate at communication round t. Note that unscheduled devices do not perform local training. Then, each device in $\mathcal{N}_t$ calculates the model update $\boldsymbol{d}_{t+1}^{k} = \boldsymbol{w}_{t+1}^{k} - \boldsymbol{w}_{t}^{k}$, where $\boldsymbol{w}_{t+1}^{k} = \boldsymbol{w}_{t,I-1}^{k}$ and $\boldsymbol{w}_{t}^{k} = \boldsymbol{w}_{t,0}^{k}$ [19]. Typically, $\boldsymbol{d}_{t+1}^{k}$ has millions of elements for DNN. It is not practical to send $\boldsymbol{d}_{t+1}^{k}$ with full precision for energy-constrained devices. Hence, we apply the same quantization scheme used in QNNs to $\boldsymbol{d}_{t+1}^{k}$ by denoting its quantized equivalent as $\boldsymbol{d}_{t+1}^{Q,k}$ with precision level m. Thus, each device in $\mathcal{N}_t$ clips its model update $\boldsymbol{d}_{t+1}^{k}$ using $\text{clip}(\cdot)$ to match the quantization range and transmits its quantized version to the BS. The received model updates are averaged by the BS, and the next global model $\boldsymbol{w}_{t+1}$ will be generated as below

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \frac{1}{K} \sum_{k \in \mathcal{N}_{t+1}} \boldsymbol{d}_{t+1}^{Q,k}. \tag{11}$$

The FL system repeats this process until the global loss function converges to a target accuracy constraint ϵ. We summarize this algorithm in Algorithm 1. Next, we propose the energy model for the computation and the transmission of our FL system.

Algorithm 1: Quantized FL Algorithm

```
Input: $K$, $I$, initial model $\boldsymbol{w}_0$, $t = 0$, target accuracy $\epsilon$
repeat
    The BS randomly selects a subset of devices $\mathcal{N}_t$
      and transmits $\boldsymbol{w}_t$ to the selected devices;
    Each device $k \in \mathcal{N}_t$ trains its QNN by running $I$
      steps of SGD as (9);
    Each device $k \in \mathcal{N}_t$ transmits $\boldsymbol{d}_{t+1}^{Q,k}$ to the BS;
    The BS generates a new global model
      $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \frac{1}{K} \sum_{k \in \mathcal{N}_t} \boldsymbol{d}_{t+1}^{Q,k}$;
    $t \leftarrow t + 1$;
until target accuracy $\epsilon$ is satisfied;
```

C. Computing and Transmission model

1) Computing model: We consider a typical two-dimensional processing chip for convolutional neural networks (CNNs) as shown in Fig. 2 [5]. This chip has a DRAM, a parallel neuron array with p MAC units, and two memory levels: a main SRAM buffer that stores the weights and activations and a local buffer that caches currently used weights and activations. Since the main SRAM buffer has a limited size, the input dataset is stored in the DRAM. Some weights can also be stored in the DRAM, if the whole weights cannot be fit in the main SRAM buffer. We use the MAC operation energy model of [20] whereby $E_{\text{MAC}}(n) = A\,(n/n_{\text{max}})^{\alpha}$ for precision level n, where $A > 0$, $1 < \alpha < 2$, and n_{max} is the maximum precision level. Here, a MAC operation includes neuron output calculation, batch normalization, activation, and back-propagation. From [20], the energy consumption for accessing a local buffer can be modeled as $E_{\text{MAC}}(n)$, and the energy for accessing a main buffer can be given by $E_{\text{m}}(n) = 2E_{\text{MAC}}(n)$. The energy consumption to access a DRAM can be modeled as $E_{\text{D}}(n) = A_d E_{\text{MAC}}(n)$, where $A_d >> 1$. [5].

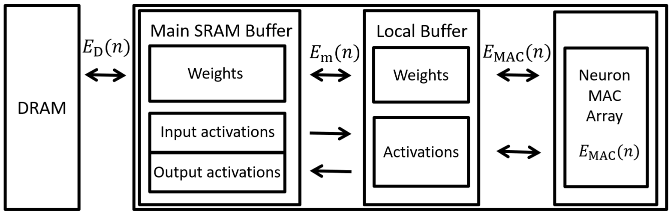


Fig. 2: An illustration of the two-dimensional processing chip.

The energy consumption of device k for doing inference (i.e., forward propagation) is $E_{\text{inf}}^{k}(n)$ when n bits are used for the quantization. Then, $E_{\text{inf}}^{k}(n)$ is the sum of the computing energy $E_{\text{C}}(n)$, the access energy for fetching weights from the buffers $E_{\text{W}}(n)$, the access energy for fetching activations from the buffers $E_{\text{A}}(n)$ and the access energy for fetching input features and weights from the DRAM $E_{\text{DRAM}}(n)$, as follows [20]:

$$\begin{aligned}
&E_{\text{inf}}^{k}(n) = E_{\text{C}}(n) + E_{\text{W}}(n) + E_{\text{A}}(n) + E_{\text{DRAM}}(n),\\
&E_{\text{C}}(n) = E_{\text{MAC}}(n) N_c + 2O_c\ E_{\text{MAC}}(n_{\text{max}}),\\
&E_{\text{W}}(n) = E_{\text{m}}(n) d + E_{\text{MAC}}(n) N_c \sqrt{n/p n_{\text{max}}},\\
&E_{\text{A}}(n) = 2E_{\text{m}}(n) O_c + E_{\text{MAC}}(n) N_c \sqrt{n/p n_{\text{max}}},\\
&E_{\text{DRAM}}(n) = E_{\text{D}}(n_{\text{max}}) x_{\text{in}} + 2E_{\text{D}}(n) \max(dn + O_c n - S, 0),
\end{aligned} \tag{12}$$

where N_c is the number of MAC operations, d is the number of weights, O_c is the number of intermediate outputs in the network, x_{in} is the input dimension, and S is the size of the main SRAM buffer. For $E_{\text{C}}(n)$, in a QNN, batch normalization and activation are done in full-precision n_{max} to each output [16]. We store quantized weights and activations in the SRAM main buffer. Once we fetch weights from a main to a local buffer, they can be reused in the local buffer afterward as shown in $E_{\text{W}}(n)$. In Fig. 2, a MAC unit fetches weights from a local buffer to do computation. Since we are using a two-dimensional MAC array of p MAC units, they can share fetched weights with the same row and column, which has $\sqrt{p}$ MAC units respectively. In addition, a MAC unit can fetch more weights due to the n bits quantization compared with when weights are represented in n_{max} bits. Thus, we can reduce the energy consumption to access a local buffer by the amount of $\sqrt{n/pn_{\text{max}}}$. A similar process applies to $E_{\text{A}}(n)$ since activations are fetched from the main buffer and should be saved back to it for the calculation in the next layer. For $E_{\text{DRAM}}(n)$, input features are processed in full-precision, and weights that cannot be stored in the SRAM will be fetched and stored to the DRAM.

As introduced in Section II-A, we calculate gradients in full-precision to average out the noise from SGD. In back-propagation, each layer calculates the gradients of its weights and the gradients of the activations of the previous layer. Hence, we can approximate the number of MAC operations as $2N_c$ as done in [21]. Then, the energy consumption for back-propagation is

$$\begin{aligned} E_{\text{back}} = {} & 2N_c E_{\text{MAC}}(n_{\text{max}}) + 2E_{\text{m}}(n_{\text{max}})O_c + E_{\text{m}}(n_{\text{max}})d \\ & + 2E_{\text{MAC}}(n_{\text{max}})N_c\sqrt{\frac{1}{p}} \\ & + 2E_{\text{D}}(n_{\text{max}})\max(dn_{\text{max}} + O_c n_{\text{max}} - s_m, 0). \end{aligned} \tag{13}$$

Since back-propagation is done in full-precision, weights must first be fetched from the DRAM. Then, we fetch weights from the main buffer to the local buffer. The neuron MAC array proceeds with the calculation by fetching the cached weights and activations from the local buffer. Therefore, the energy consumption for one iteration of device k is given by

$$E^{C,k}(n) = E^k_{\text{inf}}(n) + E_{\text{back}},\ k \in \{1, \dots, N\}. \tag{14}$$

2) Transmission Model: We use the orthogonal frequency domain multiple access (OFDMA) to transmit model updates to the BS. Each device occupies one resource block. The achievable rate of device k will be:

$$r_k = B\log_2\left(1 + \frac{P^{\text{tx}}_k \bar{h}_k}{N_0 B}\right), \tag{15}$$

where B is the allocated bandwidth, $\bar{h}_k$ is the average channel gain between device k and the BS during training[1][2], P^{tx}_k is the transmit power of device k, and N_0 is the power spectral density of white noise. After local training, device k normalizes the model update as $\boldsymbol{d}^k_t/||\boldsymbol{d}^k_t||$ to match the predetermined quantization range $[-1, 1]$. Then, it transmits $\boldsymbol{d}^{Q,k}_t$ to the BS at given communication round t. The transmission time T_k for uploading $\boldsymbol{d}^{Q,k}_t$ is given by

$$T_k(m) = \frac{dm}{r_k}. \tag{16}$$

Then, the energy consumption for the uplink transmission is given by

$$E^{UL,k}(m) = T_k(m) \times P^{\text{tx}}_k = \frac{P^{\text{tx}}_k dm}{B\log_2\left(1 + \frac{P^{\text{tx}}_k \bar{h}_k}{N_0 B}\right)}. \tag{17}$$

III. Time and Energy efficient Federated QNN

Given our model, we now formulate a multi-objective optimization problem to minimize the energy consumption and the number of communication rounds while ensuring convergence under a target accuracy. We show that a tradeoff exists between the energy consumption and the number of communication rounds as a function of I, K, m, and n. For instance, we can reduce the amount of energy spent per iteration by using low precision and sampling a small number of devices. However, this slows the convergence rate because of quantization errors. Meanwhile, the system can allocate more bits and sample more devices to converge faster, i.e, to reduce the number of communication rounds, at the expense of spending more energy. However, this improvement becomes negligible after a certain threshold as shown later in the convergence analysis and simulations (see Theorem 1 and Section IV. Hence, finding the optimal solutions is important to balance this tradeoff and to achieve the target accuracy.

We aim to minimize both the expected total energy consumption and the number of communication rounds[3] until convergence under a target accuracy ϵ as follows:

$$\min_{I,K,m,n} \quad \left[\mathbb{E}\left[\sum_{t=1}^{T}\sum_{k\in\mathcal{N}_t} E^{UL,k}(m) + IE^{C,k}(n)\right], T\right] \tag{18a}$$

$$\text{s.t.} \quad I \in [I_{\text{min}}, \dots, I_{\text{max}}], K \in [K_{\text{min}}, \dots, N] \tag{18b}$$

$$m \in [1, \dots, m_{\text{max}}], n \in [1, \dots, n_{\text{max}}] \tag{18c}$$

$$\mathbb{E}[F(\boldsymbol{w}_T)] - F(\boldsymbol{w}^*) \le \epsilon, \tag{18d}$$

where I is the number of local iterations, I_{min} and I_{max} denote the minimum and maximum of I, respectively, $\mathbb{E}[F(\boldsymbol{w}_T)]$ is the expectation of global loss function after T communication rounds, $F(\boldsymbol{w}^*)$ is the minimum value of F, and ϵ is the target accuracy. The possible values of I and K are given by (18b). Constraint (18c) represents the maximum precision levels in the transmission and the computation, respectively. Constraints (18d) captures the required number of communication rounds to achieve ϵ.

This problem is challenging since the analytical expression of (18d) with respect to the control variables is unknown. Hence, it is not trivial to derive the exact number of T to satisfy (18d). Quantization errors from local training and transmission will slow the convergence rate, thereby making achieving the target accuracy challenging. The convergence

[1] For future work, our approach can be extended to the case with instantaneous time-varying channels by considering a stochastic optimization formulation.

[2] An important subject of future work here can be the integration of more advanced MIMO-based communication channels.

[3] Minimizing the total training time by considering the impact of quantization on the computation time can be an important subject of future research.

is also not always guaranteed under non-iid data distribution. Lastly, a global optimal solution, which minimizes each objective function simultaneously, is generally infeasible for a multi-objective optimization problem [22]. Therefore, a closed-form solution may not exist due to the tradeoff between two objectives.

To solve this problem, we first obtain the analytical relationship between (18d) and I, K, m, and n to derive T with respect to ϵ. As done in [10], [19], [23], we make the following assumptions on the loss function as follows

Assumption 1. *The loss function has the following properties*

- *$F_k(\boldsymbol{w})$ is L-smooth: $\forall$ $\boldsymbol{v}$ and $\boldsymbol{w}$ $F_k(\boldsymbol{v}) \le F_k(\boldsymbol{w}) + (\boldsymbol{v}-\boldsymbol{w})^T \nabla F_k(\boldsymbol{w}) + \frac{L}{2}||\boldsymbol{v}-\boldsymbol{w}||^2$*
- *$F_k(\boldsymbol{w})$ is μ-strongly convex: $\forall$ $\boldsymbol{v}$ and $\boldsymbol{w}$ $F_k(\boldsymbol{v}) \ge F_k(\boldsymbol{w}) + (\boldsymbol{v}-\boldsymbol{w})^T \nabla F_k(\boldsymbol{w}) + \frac{\mu}{2}||\boldsymbol{v}-\boldsymbol{w}||^2$*
- *The variance of SG is bounded: $\mathbb{E}[||\nabla F_k(\boldsymbol{w}_t^k, \xi_t^k) - \nabla F_k(\boldsymbol{w}_t^k)||^2] \le \sigma_k^2,\ \forall k = 1, \ldots, N$.*
- *The squared norm of SG is bounded: $\mathbb{E}[||\nabla F_k(\boldsymbol{w}_t^k, \xi_t^k)||^2] \le G^2,\ \forall k = 1, \ldots, N$.*

These assumptions hold for some practical loss functions. Such examples include logistic regression, l_2 norm regularized linear regression, and softmax classifier [24]. Since we use the quantization in both local training and transmission, the quantization error negatively affects the accuracy and the convergence of our FL system. We next leverage the results of Lemma 1 so as to derive T with respect to ϵ in the following theorem.

Theorem 1. *For learning rate $\eta_t = \min(\frac{\beta}{t+\gamma}, \frac{1}{\rho}), \beta > \frac{1}{\mu}, \rho \gg 1, \gamma \ge 0$ and, the degree of non-iid $\Gamma = \sum_{k=1}^N p_k(F_k(\boldsymbol{w}^*) - F_k^*))$, we have*

$$\mathbb{E}[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)] \le \frac{L\beta}{2(\beta\mu - 1)}\left[\frac{\beta\psi_2}{TI+\gamma} + \psi_1\right], \quad (19)$$

where ψ_1 and ψ_2 are

$$\psi_1 = \frac{d(\rho-\mu)}{2^{2n}},$$
$$\psi_2 = \sum_{k=1}^N p_k^2\sigma_k^2 + 4(I-1)^2G^2 + \frac{4dIG^2}{K2^{2m}} + \frac{4I^2G^2}{K} + 4L\Gamma. \quad (20)$$

Proof. See Appendix C. □

We can see that ψ_1 is unavoidable because of the quantization in local training. We also observe that high precision levels for n and m can improve the convergence rate. In particular, we can decrease the quantization error related terms in ψ_1 and ψ_2 by increasing n and m. However, this improvement becomes negligible after a certain level since those terms decrease exponentially with respect to precision levels. For Γ, it quantifies the difference between the loss function at the global optimum $F_k(\boldsymbol{w}^*)$ and the one at the local optimum F_k^*. Hence, we can see that the degree of non-iid Γ degrades the convergence rate. If we set $n = n_{\text{max}}$ and $m = m_{\text{max}}$, we can approximately recover the result of [23] since the quantization error decays exponentially with respect to n and m. The convergence rate also increases with K. However, all these improvements come at the cost of consuming more energy. We can also see that (19) has the sampling probabilities related term $\sum_{k=1}^N p_k^2\sigma_k^2$ in its numerator. Therefore, we can further improve the convergence rate by optimizing p_k as follows

$$\min_{p_1,\ldots,p_N} \quad \sum_{k=1}^N p_k^2\sigma_k^2, \quad \text{s.t.} \quad \sum_{k=1}^N p_k = 1, p_k \ge 0, \forall k. \quad (21)$$

Since the above problem is convex, we can use KKT condition to solve the problem. Then, the optimal sampling probabilities can be given by $p_k = \frac{1/\sigma_k^2}{\sum_{k=1}^N 1/\sigma_k^2}$.

From Theorem 1, we can bound (19) using ϵ in (18d) as follows

$$\mathbb{E}[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)] \le \frac{L\beta}{2(\beta\mu - 1)}\left[\frac{\beta\psi_2}{TI+\gamma} + \psi_1\right] \le \epsilon. \quad (22)$$

Since ψ_1 term is not decreasing with T, there exists the minimum value of precision level n_{min} to achieve ϵ as follows

$$n_{\text{min}} = \left\lceil \frac{1}{2}\log_2\left(L\beta\frac{d(\rho-\mu)}{2\epsilon}\right)\right\rceil, \quad (23)$$

where $\lceil\cdot\rceil$ is the smallest integer larger than or equal to the input. To guarantee the convergence, we change the constraint of n in (18c) as $n \in [n_{\text{min}}, \ldots, n_{\text{max}}]$. Now, we express each objective function as function of the control variables using Theorem 1. For notational simplicity, we use $g_1(I, K, m, n)$ for the expected total energy consumption and $g_2(I, K, m, n)$ for the number of communication rounds T. Since each device k is selected with probability p_k, $\forall k$, we can derive the expectation of the energy consumption in (18a) as follows

$$\begin{aligned} g_1(I,K,m,n) &= \mathbb{E}\left[\sum_{t=1}^T \sum_{k\in\mathcal{N}_t} E^{UL,k}(m) + IE^{C,k}(n)\right] \\ &= KT\sum_{k=1}^N p_k\left\{E^{UL,k}(m) + IE^{C,k}(n)\right\}. \quad (24)\end{aligned}$$

Next, we derive $g_2(I, K, m, n)$ in a closed-form to fully express the objective functions and to remove the accuracy constraint (18d). For any feasible solution that satisfies (18d) with equality, we can always choose $T_0 > T$ such that T_0 still satisfies (18d). Since such T_0 will increase the value of the objectives, the accuracy constraint (18d) should be satisfied with equality [10]. Hence, we take equality in (22) to obtain:

$$g_2(I,K,m,n) = \frac{\beta^2\psi_2}{I(\beta\mu-1)(\frac{2\epsilon}{L} - \frac{\beta\psi_1}{\beta\mu-1})} - \frac{\gamma}{I}. \quad (25)$$

Then, we can change the original problem as below

$$\min_{I,K,m,n} \quad [g_1(I,K,m,n), g_2(I,K,m,n)] \quad \text{s.t.} \quad (18b), (18c). \quad (26a)$$

Since we have two conflicting objective functions, it is infeasible to find a global optimal solution to minimize each objective function simultaneously. Although introducing a weighted sum of the objective functions might provide a unique solution, its optimality is not always guaranteed. We also need to solve the problem again if those weights change. Hence, we instead consider the set of *Pareto optimal points* to obtain an efficient collection of solutions to minimize each objective function and capture the tradeoff. It is known that the set of all Pareto optimal points forms a Pareto boundary in two-dimensional space. Therefore, we use the so-called normal boundary inspection (NBI) method since it provides evenly distributed Pareto optimal points [25].

We first introduce some terminologies to facilitate the analysis. For a multi-objective function $\boldsymbol{g}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \dots g_M(\boldsymbol{x})]^T$ and a feasible set $\mathcal{C}$, we define $\boldsymbol{x}_i^*$ as a global solution to minimize $g_i(\boldsymbol{x})$, $i = 1 \dots M$, over $\boldsymbol{x} \in \mathcal{C}$. Let $\boldsymbol{g}_i^* = \boldsymbol{g}(\boldsymbol{x}_i^*)$ for $i = 1 \dots M$, and we define the utopia point $\boldsymbol{g}^*$, which is composed of individual global minima $\boldsymbol{g}_i^*$. We define the $M \times M$ matrix Φ, whose ith column is $\boldsymbol{g}_i^* - \boldsymbol{g}^*$. The set of the convex combinations of $\boldsymbol{g}_i^* - \boldsymbol{g}^*$ such that $\{\Phi\boldsymbol{\zeta} \mid \zeta_i \geq 0 \text{ and } \sum_{i=1}^{M} \zeta_i = 1\}$ is defined as convex hull of individual minima (CHIM) [25]. For simplicity, we now use $\mathcal{C}$ to represent all feasible constraint sets (18b) - (18c). We also define $\boldsymbol{x}_i^*$ as (I, K, m, n) such that $g_i(I, K, m, n)$ can be minimized over $\mathcal{C}$ for $i = 1$ and 2.

The basic premise of NBI is that any intersection points between the boundary of $\{\boldsymbol{g}(I, K, m, n) \mid (I, K, m, n) \in \mathcal{C}\}$ and a vector pointing toward the utopia point emanating from the CHIM are Pareto optimal. We can imagine that the set of Pareto optimal points will form a curve connecting $\boldsymbol{g}(\boldsymbol{x}_1^*) = [g_1(\boldsymbol{x}_1^*), g_2(\boldsymbol{x}_1^*)]$ and $\boldsymbol{g}(\boldsymbol{x}_2^*) = [g_1(\boldsymbol{x}_2^*), g_2(\boldsymbol{x}_2^*)]$. Hence, we first need to obtain $\boldsymbol{x}_1^*$ and $\boldsymbol{x}_2^*$. In the next two subsections, we will minimize $g_1(I, K, m, n)$ and $g_2(I, K, m, n)$ separately.

A. Minimizing $g_1(I, K, m, n)$

Since $\boldsymbol{x}_1^*$ is a global solution to minimize $g_1(I, K, m, n)$, we can find it solving:

$$\min_{I,K,m,n} \quad g_1(I, K, m, n), \quad \text{s.t.} \quad (I, K, m, n) \in \mathcal{C}. \tag{27a}$$

This problem is non-convex because the control variables are an integer and the constraints are not a convex set. For tractability, we relax the control variables as continuous variables. The relaxed variables will be rounded back to integers for feasibility. From (24) and (25), we can see that $g_1(I, K, m, n)$ is a linear function with respect to K. Therefore, K_{min} always minimizes $g_1(I, K, m, n)$. Moreover, the relaxed problem is convex with respect to I since $\frac{\partial^2 g_1(I,K,m,n)}{\partial I^2} > 0$. Hence, we can obtain the optimal I to minimize $g_1(I, K, m, n)$ from the first derivative test as

$$\frac{\partial g_1(I, K, m, n)}{\partial I} = H_1 I^3 + H_2 I^2 + H_3 = 0, \tag{28}$$

where

$$H_1 = (8KG^2 + 8G^2) \sum_{k=1}^{N} p_k E^{C,k}(n), \tag{29}$$

$$\begin{aligned} H_2 = {} & (4KG^2 + 4G^2) \sum_{k=1}^{N} p_k E^{UL,k}(m) \\ & + (-8KG^2 + \frac{4dG^2}{2^{2m}}) \sum_{k=1}^{N} p_k E^{C,k}(n), \end{aligned} \tag{30}$$

$$\begin{aligned} H_3 = {} & -K \left(\sum_{k=1}^{N} p_k^2 \sigma_k^2 + 4G^2 + 4L\Gamma \right. \\ & \left. -\gamma(\beta\mu - 1) \left(\frac{2\epsilon}{L} - \frac{\beta\psi_1}{\beta\mu - 1} \right) \frac{1}{\beta^2} \right) \sum_{k=1}^{N} p_k E^{UL,k}(m) \end{aligned} \tag{31}$$

Here, H_1 and H_3 express the cost of local training and the cost of transmission, respectively, while H_2 depends on both of them. We next present a closed-form solution of the above equation from Cardano's formula [26].

Lemma 2. *For given m and n, the optimal I' to minimize $g_1(I, K, m, n)$ is given by*

$$\begin{aligned} I' = {} & \sqrt[3]{-\frac{H_2^3}{27H_1^3} - \frac{H_3}{2H_1} + \sqrt{\frac{1}{4}\left(\frac{2H_2^3}{27H_1^3} + \frac{H_3}{H_1}\right)^2 + \frac{1}{27}\left(\frac{H_2^2}{3H_1^2}\right)^3}} \\ & + \sqrt[3]{-\frac{H_2^3}{27H_1^3} - \frac{H_3}{2H_1} - \sqrt{\frac{1}{4}\left(\frac{2H_2^3}{27H_1^3} + \frac{H_3}{H_1}\right)^2 + \frac{1}{27}\left(\frac{H_2^2}{3H_1^2}\right)^3}} \\ & - \frac{H_2}{3H_1} \end{aligned} \tag{32}$$

From Lemma 2, we can see that the value of I' decreases due to the increased cost of local training H_1 as we allocate a larger n. Since the quantization error decreases as n increases, a large I' is not required. Hence, an FL system can decrease the value of I' to reduce the increased local computation energy. We can also see that I' increases as the cost of transmission H_3 increases. Then, for convergence, the FL algorithm can perform more local iterations instead of frequently exchanging model parameters due to the increased communication overhead.

Although $g_1(I, K, m, n)$ is non-convex with respect to m, there exists $m' \in \mathcal{C}$ such that for $m \leq m'$, $g_1(I, K, m, n)$ is non-increasing, and for $m \geq m'$, $g_1(I, K, m, n)$ is non-decreasing. This is because $g_1(I, K, m, n)$ decreases as the convergence rate becomes faster for increasing m. Then, $g_1(I, K, m, n)$ increases after m' due to unnecessarily allocated bits. Since $g_1(I, K, m, n)$ is differentiable at m, we can find such local optimal m' from $\partial g_1(I, K, m, n)/\partial m = 0$ using Fermat's Theorem [4]. To obtain m', we formulate the transcendental equation as below

$$\frac{\partial g_1(I, K, m, n)}{\partial m} = 0 \leftrightarrow m = M_A 2^{2m} + M_B, \tag{33}$$

where

$$\begin{aligned} M_A = {} & \frac{K(\sum_{k=1}^{N} p_k^2 \sigma_k^2 + 4(I-1)^2 G^2 + 4I^2 G^2/K + 4L\Gamma}{4dIG^2 \log 4} \\ & - \frac{\gamma \frac{I(\beta\mu - 1)(2\epsilon/L - \beta d(\rho - \mu))/(2^{2n}(\beta\mu - 1))))}{\beta^2 I}}{4dIG^2 \log 4}, \\ M_B = {} & \frac{\frac{4dIG^2 M_C}{K} - 4I^2 G^2 \log 4 \sum_{k=1}^{N} p_k E^{C,k}(n)}{4dIG^2 \log 4 M_C} \\ M_C = {} & \sum_{k=1}^{N} p_k \frac{P_k^{\text{tx}}}{B \log_2 \left(1 + \frac{P_k^{\text{tx}} \bar{h}_k}{N_0 B}\right)}. \end{aligned} \tag{34}$$

We present a closed-form solution of the above equation in the following Lemma.

Lemma 3. *For given I and n, the local optimal m' to minimize $g_1(I, K, m, n)$ will be:*

$$m' = M_B - \frac{1}{\log 4} W\left(-M_A \log 4 \exp(M_B \log 4)\right), \tag{35}$$

where $W(\cdot)$ is the Lambert W function.

Following the same logic of obtaining m', we can find a local optimal solution n' from the first derivative test. Although there is no analytical solution for n', we can still obtain it numerically using a line search method. Then, problem (27a) can be optimized iteratively. We first obtain two analytical solutions for I and m. From these solutions, we numerically find a local optimal n'. Since $g_1(I, K, m, n)$ has a unique solution to each variable, it converges to a stationary point [27]. Although these points cannot guarantee to obtain globally Pareto optimal, using the NBI method, we are still guaranteed to reach locally Pareto optimal points [25]. In Section IV, we will also numerically show that the obtained points can still cover most of the practical portion of a global Pareto boundary. For ease of exposition, hereinafter, we refer to these local Pareto optimal points as "Pareto optimal".

B. Minimizing $g_2(I, K, m, n)$

Now, we obtain $\boldsymbol{x}_2^*$ from the following problem to complete finding the utopia point.

$$\min_{I,K,m,n} \quad g_2(I, K, m, n), \quad \text{s.t.} \quad (I, K, m, n) \in \mathcal{C}. \tag{36a}$$

From (25), the objective function is a decreasing function with respect to K, m, and n. Hence, $N, m_{\max}$, and $n_{\max}$ are always the optimal solutions to the above problem. Then, the problem can be reduced to a single variable optimization problem with respect to I. We check the convexity of the reduced problem as follows:

$$\begin{aligned}\frac{\partial^2 g_2(I, K, m, n)}{\partial I^2} &= \frac{\beta^2}{(\beta\mu - 1)(\frac{2\epsilon}{L} - \frac{\beta\psi_1}{\beta\mu-1})} \\ &\quad \times \left\{\sum_{k=1}^{N} \frac{2p_k^2\sigma_k^2}{I^3} + \frac{8G^2}{I^3} + \frac{8L\Gamma}{I^3}\right\} - \frac{2\gamma}{I^3}.\end{aligned} \tag{37}$$

Hence, it is a convex problem for $\gamma < \frac{\beta^2}{(\beta\mu-1)(\frac{2\epsilon}{L}-\frac{\beta\psi_1}{\beta\mu-1})}\left\{\sum_{k=1}^{N} p_k^2\sigma_k^2 + 4G^2 + 4L\Gamma\right\}$. Since γ is an arbitrary constant such that $\gamma \geq 0$, we can always find γ that satisfies the above condition. We present a closed-form solution of I from the first derivative test in the following lemma.

Lemma 4. *For $\gamma < \frac{\beta^2\left\{\sum_{k=1}^{N} p_k^2\sigma_k^2+4G^2+4L\Gamma\right\}}{(\beta\mu-1)(\frac{2\epsilon}{L}-\frac{\beta\psi_1}{\beta\mu-1})}$, the optimal value of I'' to minimize $g_2(I, K, m, n)$ is given by*

$$I'' = \sqrt{\frac{\sum_{k=1}^{N} p_k^2\sigma_k^2 + 4G^2 + 4L\Gamma - \gamma(\beta\mu - 1)(\frac{2\epsilon}{L} - \frac{\beta\psi_1}{\beta\mu-1})/\beta^2}{4G^2 + \frac{4G^2}{K}.}} \tag{38}$$

From Lemma 4, we can see that the optimal value of I'' increases as n decreases. This is because the system has to reduce quantization error by training more number of times.

C. Normal Boundary Inspection

We now obtain the Pareto boundary using NBI. We redefine $\boldsymbol{g}(I, K, m, n) := \boldsymbol{g}(I, K, m, n) - \boldsymbol{g^*}$ so that the utopia point can be located at the origin. The NBI method aims to find intersection points between the boundary of $\boldsymbol{g}(I, K, m, n)$ and a normal vector $\hat{\boldsymbol{n}} = -\boldsymbol{\Phi 1}$, where $\mathbf{1}$ denotes the column vector consisting of only ones which are pointing toward the origin. Then, the set of points on such a normal vector will be: $\boldsymbol{\Phi}\zeta + s\hat{\boldsymbol{n}}$, where $s \in \mathbb{R}$. The intersection points can be obtained from the following subproblem:

$$\max_{I,K,m,n,s} \quad s \tag{39a}$$

$$\text{s.t.} \quad (I, K, m, n) \in \mathcal{C} \tag{39b}$$

$$\boldsymbol{\Phi\zeta} + s\hat{\boldsymbol{n}} = \boldsymbol{g}(I, K, m, n), \tag{39c}$$

where (39c) makes the set of points on $\boldsymbol{\Phi}\zeta + s\hat{\boldsymbol{n}}$ be in the feasible area. From the definitions of $\boldsymbol{\Phi}$ and $\hat{\boldsymbol{n}}$, constraint (39c) can be given as

$$\boldsymbol{\Phi\zeta} + s\hat{\boldsymbol{n}} = \begin{bmatrix} g_1(\boldsymbol{x}_2^*)(\zeta_2 - s) \\ g_2(\boldsymbol{x}_1^*)(\zeta_1 - s) \end{bmatrix} = \begin{bmatrix} g_1(I, K, m, n) \\ g_2(I, K, m, n) \end{bmatrix}. \tag{40}$$

From (40), we obtain the expression of s as below

$$s = \zeta_1 - \frac{g_2(I, K, m, n)}{g_2(\boldsymbol{x}_1^*)} = \zeta_2 - \frac{g_1(I, K, m, n)}{g_1(\boldsymbol{x}_2^*)}. \tag{41}$$

Hence, we can change problem (39a) as follows

$$\min_{I,K,m,n} \quad \frac{g_2(I, K, m, n)}{g_2(\boldsymbol{x}_1^*)} - \zeta_1 \tag{42a}$$

$$\text{s.t.} \quad (I, K, m, n) \in \mathcal{C} \tag{42b}$$

$$1 - 2\zeta_1 + \frac{g_2(I, K, m, n)}{g_2(\boldsymbol{x}_1^*)} - \frac{g_1(I, K, m, n)}{g_1(\boldsymbol{x}_2^*)} = 0, \tag{42c}$$

where we substituted s with (41) for the objective function, constraint (42c) is from (41), and $\zeta_1 + \zeta_2 = 1$. To remove the equality constraint (42c), we approximate the problem by introducing a quadratic penalty term λ as below

$$\begin{aligned}\min_{I,K,m,n} \quad & \frac{g_2(I, K, m, n)}{g_2(\boldsymbol{x}_1^*)} - \zeta_1 + \lambda\Bigg(1 - 2\zeta_1 + \frac{g_2(I, K, m, n)}{g_2(\boldsymbol{x}_1^*)} \\ & - \frac{g_1(I, K, m, n)}{g_1(\boldsymbol{x}_2^*)}\Bigg)^2 \end{aligned} \tag{43a}$$

$$\text{s.t.} \quad (I, K, m, n) \in \mathcal{C}. \tag{43b}$$

For λ, we consider an increasing sequence $\{\lambda_i\}$ with $\lambda_i \to \infty$ as $i \to \infty$ to penalize the constraint violation more strongly. We then obtain the corresponding solution $\boldsymbol{x}^i$, which is (I, K, m, n) for minimizing problem (43a) with penalty parameter λ_i.

Theorem 2. *For $\lambda_i \to \infty$ as $i \to \infty$, solution $\boldsymbol{x}^i$ approaches the global optimal solution of problem* (43a)*, and it also becomes Pareto optimal.*

Proof. For notational simplicity, we use $\boldsymbol{x}$ to denote $(I, K, m, n) \in \mathcal{C}$. Let $q^p(\boldsymbol{x})$ denote the quadratic penalty term in problem (43a). We also define a global optimal solution to the problem (42a) as $\bar{\boldsymbol{x}}$. Since $\boldsymbol{x}^i$ minimizes the above problem with penalty parameter λ_i, we have

$$\begin{aligned}\frac{g_2(\boldsymbol{x}^i)}{g_2(\boldsymbol{x}_1^*)} - \zeta_1 + \lambda_i q^p(\boldsymbol{x}^i) &\leq \frac{g_2(\bar{\boldsymbol{x}})}{g_2(\boldsymbol{x}_1^*)} - \zeta_1 + \lambda_i q^p(\bar{\boldsymbol{x}}) \\ &\leq \frac{g_2(\bar{\boldsymbol{x}})}{g_2(\boldsymbol{x}_1^*)} - \zeta_1, \end{aligned} \tag{44}$$

where the last inequality is from the fact that $\bar{\boldsymbol{x}}$ minimizes problem (42a) with the equality constraint of $q^p(\bar{\boldsymbol{x}})$ being zero. Then, we obtain the inequality of $q^p(\boldsymbol{x}^i)$ as follows

$$q^p(\boldsymbol{x}^i) \leq \frac{1}{\lambda_i}\left(\frac{g_2(\bar{\boldsymbol{x}})}{g_2(\boldsymbol{x}_1^*)} - \frac{g_2(\boldsymbol{x}^i)}{g_2(\boldsymbol{x}_1^*)}\right). \tag{45}$$

By taking the limit as $i \to \infty$, we have

$$\lim_{i\to\infty} q^p(\boldsymbol{x}^i) \leq \lim_{i\to\infty} \frac{1}{\lambda_i}\left(\frac{g_2(\bar{\boldsymbol{x}})}{g_2(\boldsymbol{x}_1^*)} - \frac{g_2(\boldsymbol{x}^i)}{g_2(\boldsymbol{x}_1^*)}\right) = 0. \tag{46}$$

Hence, as $\lambda_i \to \infty$, we can see that $\boldsymbol{x}^i$ approaches the global optimal solution of (42a), which aims to find a Pareto optimal point. □

From Theorem 2, we can obtain a global optimal solution of (42a), and this correspond to a Pareto optimal point for specific values of ζ_1 and ζ_2. Note that problem (43a) can be solved using a software solver. To fully visualize the boundary, we iterate problem (39a) for various combinations of ζ_1 and ζ_2. The overall algorithm is given in Algorithm 2.

The main complexity of Algorithm 2 at each iteration is to solve problem (42a), which corresponds to line $9-13$. We approximated problem (42a) to problem (43a), which can be solved by a software solver. If we use the interior point method with a desired accuracy ϵ_{in}, then the complexity can given by $\mathcal{O}(\log(\frac{1}{\epsilon_{\text{in}}}))$ [28]. Since we solve (43a) by increasing λ_i at iteration i, the complexity of this outer loop can be given by $\mathcal{O}(\log(\frac{1}{\epsilon_{\text{out}}}))$ with a desired accuracy ϵ_{out}. Therefore, the complexity of Algorithm 2 is $\mathcal{O}(\log(\frac{1}{\epsilon_{\text{out}}})\log(\frac{1}{\epsilon_{\text{in}}}))$.

D. Nash Bargaining Solution

Since the solutions from (18a) are Pareto optimal, there is always an issue of choosing the best point. This is because any improvement on one objective function leads to the degradation of another. We can tackle this problem considering a bargaining process [29] between two players: one tries to minimize the energy consumption and another aims to reduce the number of communication rounds. Since the parameters of FL, i.e., (I, K, m, n), are shared, the players should reach a certain agreement over the parameters. It is known that NBS can be a unique solution to this bargaining process. The NBS was chosen here because it satisfies several fairness axioms [29], and thus, it has been used as a fair solution to resource management problems [30], [31]. We can obtain the NBS from the following problem [29]:

$$\begin{aligned}\max_{g_1(\boldsymbol{x}), g_2(\boldsymbol{x})} \quad & (g_1(\boldsymbol{D}) - g_1(\boldsymbol{x}))(g_2(\boldsymbol{D}) - g_2(\boldsymbol{x})) \\ \text{s.t.} \quad & (g_1(\boldsymbol{x}), g_2(\boldsymbol{x})) \in \overline{g_{\text{ach}}}, \end{aligned} \tag{47a}$$

where $g_{\text{ach}} = \underset{\boldsymbol{x}\in\mathcal{C}}{\cup} (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}))$ is the achievable set of $(g_1(\boldsymbol{x}), g_2(\boldsymbol{x}))$, $\overline{g_{\text{ach}}}$ represents the convex hull of g_{ach}, and $\boldsymbol{D}$ is the outcome when the players fail to cooperate. Since the NBS always lies on the Pareto boundary, we perform the bargaining process on the obtained boundary from Algorithm 2. Then, we can find the NBS graphically by finding a tangential point where the boundary and a parabola $(g_1(\boldsymbol{D}) - g_1(\boldsymbol{x}))(g_2(\boldsymbol{D}) - g_2(\boldsymbol{x}) = \Delta$ intersects with constant Δ.

Algorithm 2: NBI approach to obtain Pareto boundary

```
   Input: $N, B, P^{\text{tx}}, I_{\min}, K_{\min}, m_{\max}, n_{\max}, \beta, \gamma, G, \sigma, \mu, L, A, \alpha$,
          accuracy constraint $\epsilon$, loss function $F_k(\cdot)$,
          stopping criterion $\epsilon_{\text{uto}}$ and $\epsilon_{\text{out}}$, and a structure
          of QNN
 1 To find $\boldsymbol{g}_1^*$, initialize $(I, K, m, n)$ and set $K = N$
 2 while $\sqrt{(I - I')^2 + (m - m')^2 + (n - n')^2} > \epsilon_{uto}$ do
 3     Update $(I, m, n)$ as $(I', m', n')$
 4     Obtain $I$' from (32)
 5     Obtain $m'$ for fixed $I'$ from (35)
 6     Obtain $n'$ for fixed $I'$ and $m'$ using a line search
 7 To find $\boldsymbol{g}_2^*$, calculate $I''$ from Lemma 4 and set
    $(K, m, n) = (N, m_{\max}, n_{\max})$
 8 while $\zeta_1 \leq 1$ do
 9     Initialize $\boldsymbol{x}$, which denotes a vector $(I, K, m, n)$
        repeat
10          Update $\boldsymbol{x}$ as $\boldsymbol{x}'$
11          Obtain $\boldsymbol{x}'$ from problem (43a)
12          Increase $\lambda$
13     until $\sqrt{||\boldsymbol{x} - \boldsymbol{x}'||^2} \leq \epsilon_{out}$;
14     Round $(I, K, m, n)$ and increase $\zeta_1$
```

IV. Simulation Results and Analysis

For our simulations, unless stated otherwise, we uniformly deploy $N = 50$ devices over a square area of size 500 m × 500 m serviced by one BS at the center, and we assume a Rayleigh fading channel with a path loss exponent of 4. We assume that the FL algorithm is used for a classification task with MNIST dataset. We distribute the training dataset over devices in a non-iid fashion by allocating labels from a Dirichlet distribution with parameter 0.1. A softmax classifier is used to measure our FL performance. We also use $P_k^{\text{tx}} = 100$ mW, $B = 10$ MHz, $N_0 = -173$ dBm, $S = 2$ MB, $x_{\text{in}} = 786$, $m_{\max} = 32$ bits, $n_{\max} = 32$ bits, $I_{\min} = 1$, $I_{\max} = 30$, $K_{\min} = 1$, $\epsilon = 0.1$, $\rho = 100$, and $\gamma = 1$, $\forall k = 1, \ldots, N$. For L, we used the reported value $L = 0.097$ with the same dataset and the loss function [32]. However, estimating μ is more challenging than the estimation of L. Since the value of μ is widely assumed to be a small value between $[0.001, 1]$ [33] [34], we used $\mu = 0.05$ as done in [34] with the same dataset. We assume that each device trains a QNN structure with five convolutional layers and three fully-connected layers. Specifically, the convolutional layers consist of 128 kernels of size 3 × 3, two of 64 kernels of size 3×3, and two of 32 kernels of size 3x3. The first layer is followed by 3x3

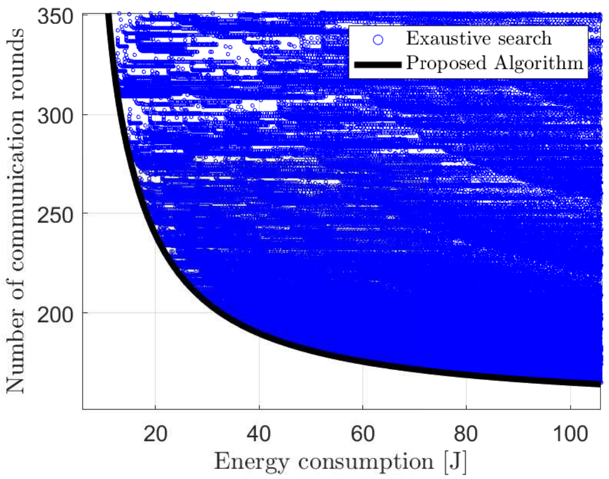


Fig. 3: Pareto Boundary from Algorithm 2 and feasible area from exhaustive search

	$N = 10$	$N = 50$	$N = 200$
NBS	$(2, 2, 12, 19)$	$(1, 5, 12, 19)$	$(1, 23, 12, 19)$
SUM	$(3, 3, 12, 20)$	$(1, 7, 12, 20)$	$(1, 35.12.19)$
E_{min}	$(1, 1, 10, 15)$	$(1, 1, 11, 15)$	$(1, 1, 12, 15)$
T_{min}	$(3, 10, 32, 32)$	$(1, 50, 32, 32)$	$(1, 200, 32, 32)$

(a) Solutions for varying N

	CNN1	CNN2	CNN3
NBS	$(1, 3, 11, 19)$	$(1, 5, 12, 19)$	$(1, 8, 14, 20)$
SUM	$(1, 4, 11, 20)$	$(1, 7, 12, 20)$	$(1, 2, 14, 20)$
E_{min}	$(1, 1, 10, 15)$	$(1, 1, 11, 15)$	$(1, 1, 14, 16)$
T_{min}	$(2, 50, 32, 32)$	$(1, 50, 32, 32)$	$(1, 50, 32, 32)$

(b) Solutions for varying model size

TABLE II: Solutions of NBS, SUM, E_{min}, and T_{min} for varying N and the model size.

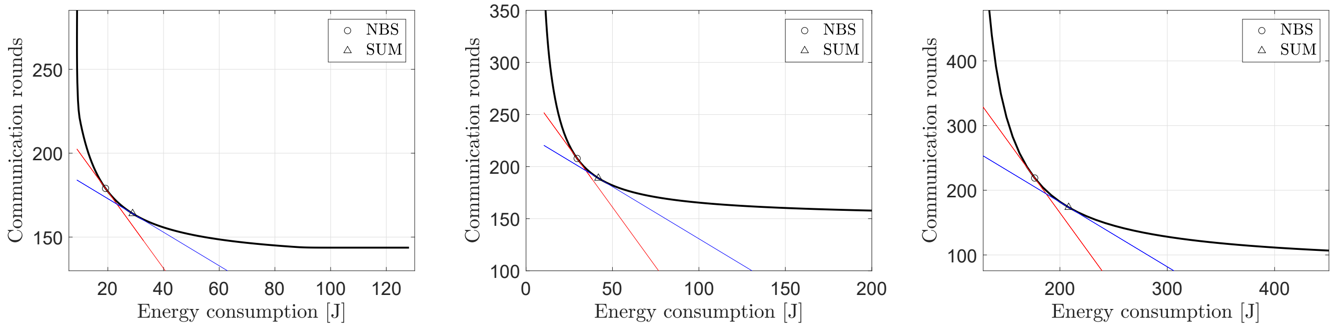
pooling and the second and the fifth layer are followed by 3x3 max pooling with a stride of two. Then, we have one dense layer of 2000 neurons, one fully-connected layer of 100 neurons, and the output layer. In this setting, we have $N_c = 0.0405 \times 10^9, d = 0.41 \times 10^6$, and $O_s = 4990$. To estimate G and σ_k, we measure every device's average maximum norm of stochastic gradients G_k for the initial 20 local iterations and set $G = \max_k \ G_k, \forall k = \{1, \dots, N\}$. We used the same gradients information to estimate σ_k while measuring G_k. Since the norm of the stochastic gradient generally decreases with training epochs, we use the initial values of G_k to estimate G as in [35]. Similarly, since loss functions are in general decreasing with training epoch, we can bound Γ as $\Gamma = \sum_{k=1}^{N} p_k(F_k(\boldsymbol{w}^*) - F_k^*) \leq \sum_{k=1}^{N} p_k F_k(\boldsymbol{w}^*) \leq \sum_{k=1}^{N} p_k F_k(\boldsymbol{w}')$, where $\boldsymbol{w}'$ can be a global model in early stage. From the above setting, we estimated $G = 0.25$. We then used the global model, which was used to measure G, to estimate $\Gamma = 0.6$. For the computing model, we use a 28 nm technology processing chip and set $A = 3.7$ pJ, $A_d = 150$, and $\alpha = 1.25$ as done in [20]. For the disagreement point $\boldsymbol{D}$, we use $(I_{\text{max}}, 1, 1, n_{\text{min}})$ as this setting is neither biased towards minimizing the energy consumption nor towards the number of communication rounds. We assume that each device has the same architecture of the processing chip. All statistical results are averaged over a number of independent runs.

Figure 3 shows the Pareto boundary from Algorithm 2 as well as the feasible area obtained from the exhaustive search for $N = 50$. We can see that our boundary and the actual Pareto boundary match well. Although we cannot find the global Pareto optimal points due to the non-convexity of problem (27a), it is clear that our analysis can still cover most of the important points that can effectively show the tradeoff in the feasible region.

Figure 4 and Table IIa show the Pareto boundaries obtained from the Algorithm 2 and the solutions of four possible operating points, respectively, for varying N. Each solution represents (I, K, m, n), where I is the number of local iterations, K is the number of sampled devices, m is the precision level for transmission, and n is the precision level for local training. SUM represents the point that minimizes the sum of the two objectives. We can obtain the SUM by finding a tangential point between the Pareto boundary and the line $g_1(I, K, m, n) + g_2(I, K, m, n) = \Delta$ with $\Delta \in \mathbb{R}$ using a bisection algorithm. E_{min} and T_{min} are the solutions that separately optimize $g_1(I, K, m, n)$ and $g_2(I, K, m, n)$, respectively. From Fig. 4, we can see that the energy consumption increases while the number of communication rounds decreases to achieve the target accuracy for increasing N. The FL system can choose more devices at each communication round as N increases. Hence, the impact of SG variance decreases as shown in Theorem 1. Since involving more devices in the averaging process implies an increase in the size of the batch, the convergence rate increases by using more energy [36].

From Table IIa and Fig. 4, we can see that NBS points are more biased toward reducing the energy consumption while the SUM points focus on minimizing communication rounds. We can also see that, as N becomes larger, the optimal I decreases while K increases. This is because I is a decreasing function with respect to G as shown in Lemmas 2 and 4. Hence, the FL system decreases I to avoid model discrepancy over devices since the estimated value of G becomes larger for increasing N. However, a small I will slow down the process to reach optimal weights in the local training. To mitigate this, the FL system then increases K so that it can obtain more information in the averaging process by selecting more devices.

Figure 5 and Table IIb present the Pareto boundaries from the Algorithm 2 and the corresponding solutions when increasing the size of the neural networks. We keep the same structure of our default CNN, but we now increase the number of neurons in the convolutional and fully-connected layers. For each CNN model, the number of parameters will be $0.27 \times 10^6, 0.41 \times 10^6$, and 1.61×10^6, respectively. Fig. 5 and Table IIb show that the energy consumption and the number of communication rounds until convergence increase with the model size. For CNN3, the energy cost increased significantly since its large model size cannot be fit into the SRAM even after quantization. From Table IIb, we can see that the FL system requires higher precision levels for larger neural

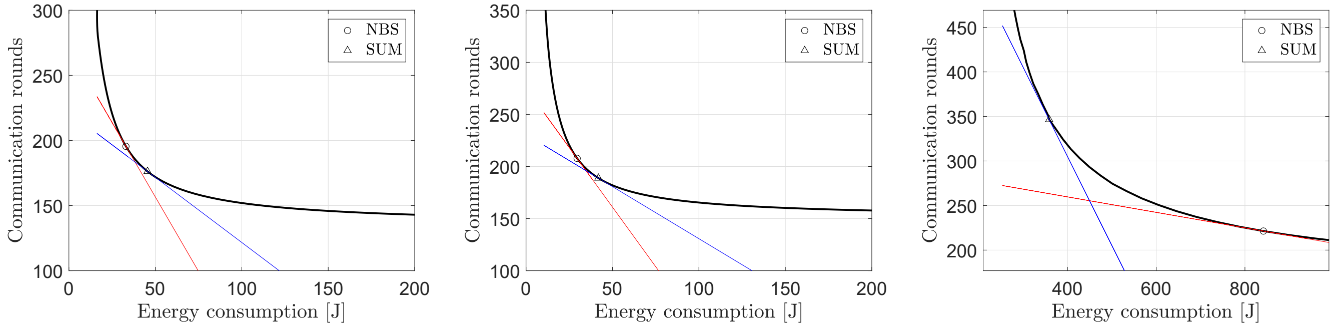


(a) N = 10

(b) N = 50

(c) N = 200

Fig. 4: Pareto boundaries, NBS, and SUM points for varying the number of devices N.



(a) CNN model 1

(b) CNN model 2

(c) CNN model 3

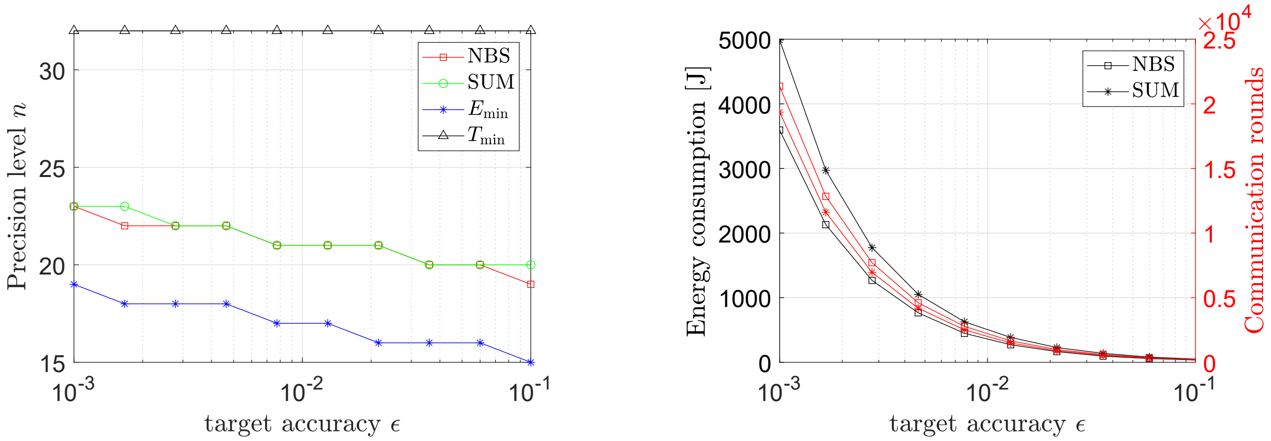
Fig. 5: Pareto boundaries, NBS, and SUM points for varying the model size.

networks. This is because the quantization error increases for larger neural networks, as per Lemma 1. Hence, the FL system allocates more bits for both the computation and the transmission so as to mitigate the quantization error. This, in turn, means that the use of larger neural networks will naturally require more energy, even if the neural network is quantized.

Figure 6a presents the impact of target accuracy ϵ on the optimal precision level n and the performance. We can see that as target accuracy ϵ increases, the optimal precision level n also increases to achieve the convergence. This is because quantization in local training yields the unavoidable term ψ_1 as shown in Theorem 1. Hence, to achieve a higher target accuracy ϵ, we need to allocate more precision level n for local training to achieve the convergence. However, this can increase the number of DRAM accesses to fetch model parameters due to the increased memory size. Since the DRAM access energy is much larger than the MAC operation energy, the energy consumption may increase significantly. In Fig. 6b, we can see that we need much more energy and communication rounds to achieve a higher target accuracy ϵ.

In Fig. 7, we show the performance of the NBS and the SUM points with increasing K. We can see that the required communication rounds decrease as K increases for both schemes. Hence, we can improve the convergence rate by increasing K at the expense of more energy. This corroborates the analysis in Section III-A, which shows the total energy consumption is linear with respect to K. Similarly, it also corroborates the fact that the required number of communication rounds to achieve a certain ϵ is a decreasing function of K, i.e., $\mathcal{O}(\frac{1}{K})$, in Section III-B. However, we can see that this improvement is not much beneficial [23] as it linearly increases the energy consumption.

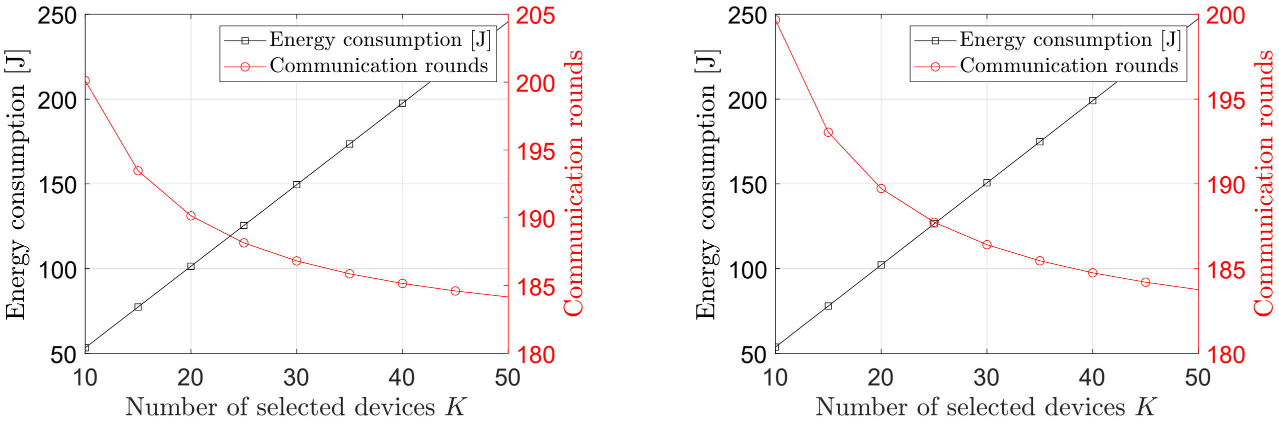
Figure 8 shows the required energy and communication rounds to achieve ϵ using the NBS points. For FedAvg [4], we use $(2, 5, 32, 32)$. FedPaq algorithm [37] uses periodic averaging, partial client participation, and quantization in transmission. Hence, we only optimize m and use the same setting as FedAvg. iFedAvg scheme is proposed in [10], and it optimizes (I, K) while data is represented in full-precision. UnifiedQ is a baseline introduced in the work in [15], and it optimizes (I, n). We use $m = 16$ as done in [15]. Here, we set $K = 5$ for a fair comparison while the original version sampled whole devices at each round. For mnFedAvg, we only optimize (m, n) with $(I, K) = (2, 5)$. All optimal parameters of the baselines are obtained from solving (18a). From Figs. 8a and 8b, we can see that our algorithm is the most efficient because it consumes the least energy and converges faster than other baselines to achieve ϵ. This is because we optimize all system parameters (I, K, m, n) simultaneously. From Baseline 1 and 2, we observe that quantization during transmission is beneficial to save the energy, and it does not significantly affect the convergence rate. In particular, we can achieve around 70% of energy savings compared to FedAvg and around 16% of energy savings compared to UnifiedQ.



(a) Impact of target accuracy ϵ on n

(b) Impact of target accuracy ϵ on the performance

Fig. 6: Impact of target accuracy ϵ on the optimal precision level n and the performance



(a) Performance of NBS with different K

(b) Performance of SUM with different K

Fig. 7: Performance of NBS and SUM points for increasing K.

V. Conclusion

In this paper, we have studied the problem of energy-efficient quantized FL over wireless networks. We have presented the energy model for our FL based on the physical structure of a processing chip considering the quantization. Then, we have formulated a multi-objective optimization problem to minimize the energy consumption and the number of communication rounds simultaneously under a target accuracy by controlling the number of local iterations, the number of selected users, the precision levels for the transmission, and the training. To solve this problem, we first have derived the convergence rate of our quantized FL. Based on it, we have used the NBI method to obtain the Pareto boundary. We also have derived analytical solutions that can optimize each objective function separately. Simulation results have validated our theoretical analysis and provided design insights with two practical operating points. We have also shown that our model requires much less energy than a standard FL model and the baselines to achieve the convergence. In essence, this work provides the first systematic study on how to optimally design quantized FL balancing the tradeoff between energy efficiency and convergence rate, and the target accuracy over wireless networks.

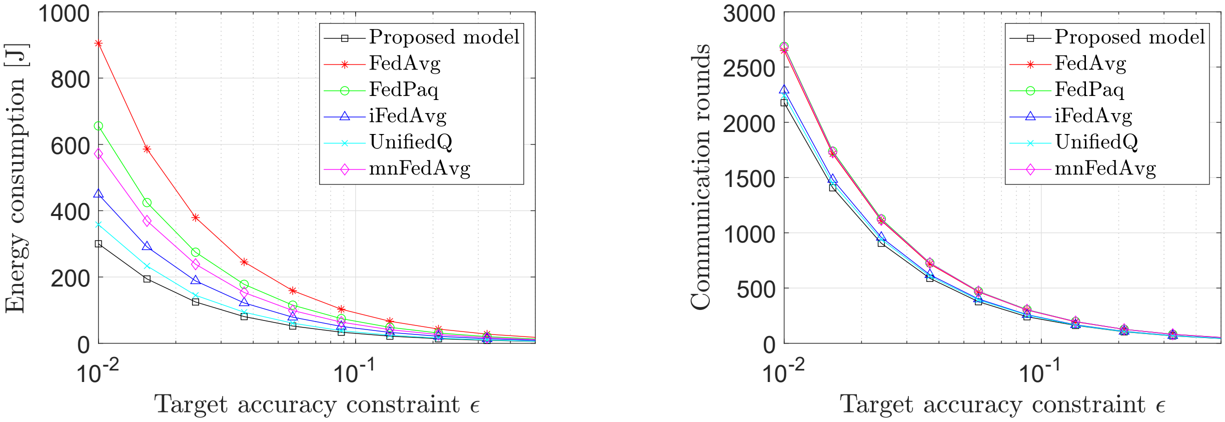
Appendix

A. Additional Notations

As done in [23], we define t as the round of the local iteration with a slight abuse of notation. Then, $\boldsymbol{w}_t^k$ becomes the model parameter at local iteration t of device k. If $t \in \mathcal{I}$, where $\mathcal{I} = \{jI \mid j = 1, 2, \dots\}$, each device transmits model update $\boldsymbol{d}_t^{Q,k}$ to the BS. We introduce an auxiliary variable $\boldsymbol{v}_{t+1}^k$ to represent the result of one step of local training from $\boldsymbol{w}_t^k$. At each local training, device k updates its local model using SGD as below

$$\boldsymbol{v}_{t+1}^k = \boldsymbol{w}_t^k - \eta_t \nabla F_k(\boldsymbol{w}_t^{Q,k}, \xi_t^k). \tag{48}$$

The result of the $(t+1)$th local training will be $\boldsymbol{w}_{t+1}^k = \boldsymbol{v}_{t+1}^k$ if $t+1 \notin \mathcal{I}$ because device k does not send a model update to the BS. If $t+1 \in \mathcal{I}$, each device calculates and transmits its model update, and then the global model is generated as $\boldsymbol{w}_{t+1} = \boldsymbol{w}_{t-I+1} + \frac{1}{K}\sum_{k\in\mathcal{N}_{t+1}} \boldsymbol{d}_{t+1}^{Q,k}$. Note that $\boldsymbol{d}_{t+1}^{Q,k} = Q(\boldsymbol{v}_{t+1}^k -$



(a) Energy consumption to achieve ϵ

(b) Communication rounds to achieve ϵ

Fig. 8: Performance comparison between the proposed model and the baselines to achieve ϵ.

$\boldsymbol{w}_{t-I+1}$) and $\boldsymbol{w}_{t-I+1}$ is the most recent global model received from the BS. We provide the aforementioned cases below:

$$\boldsymbol{w}_{t+1}^k = \begin{cases} \boldsymbol{v}_{t+1}^k & \text{if} \quad t+1 \notin \mathcal{I}, \\ \boldsymbol{w}_{t-I+1} + \frac{1}{K}\sum_{k\in\mathcal{N}_{t+1}} \boldsymbol{d}_{t+1}^{Q,k} & \text{if} \quad t+1 \in \mathcal{I}. \end{cases} \tag{49}$$

Now, we define two more auxiliary variables: $\bar{\boldsymbol{v}}_t = \sum_{k=1}^N p_k \boldsymbol{v}_t^k$ and $\bar{\boldsymbol{w}}_t = \sum_{k=1}^N p_k \boldsymbol{w}_t^k$. Similarly, we denote $\delta_t = \sum_{k=1}^N p_k \nabla F_k(\boldsymbol{w}_t^{Q,k}, \xi_t^k)$ and $\bar{\delta}_t = \sum_{k=1}^N p_k \nabla F_k(\boldsymbol{w}_t^{Q,k})$. From (48), we can see that $\bar{\boldsymbol{v}}_{t+1} = \bar{\boldsymbol{w}}_t - \eta_t \delta_t$.

B. The result of one local iteration

We present a preliminary lemma to prove Theorem 1. We first present the result of one iteration of local training in the following lemma.

Lemma 5. *Under Assumption 1, we have*

$$\mathbb{E}\left[||\bar{\boldsymbol{v}}_{t+1} - \boldsymbol{w}^*||^2\right] \leq (1-\mu\eta_t)\mathbb{E}\left[||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2\right] + \frac{\eta_t d(\rho-\mu)}{2^{2d}} + \eta_t^2\left(\sum_{k=1}^N p_k^2\sigma_k^2 + 4(I-1)^2G^2 + 4L\Gamma\right). \tag{50}$$

Proof. From $\bar{\boldsymbol{v}}_{t+1} = \bar{\boldsymbol{w}}_t - \eta_t\delta_t$, we have

$$\begin{aligned} ||\bar{\boldsymbol{v}}_{t+1} - \boldsymbol{w}^*||^2 &= ||\bar{\boldsymbol{w}}_t - \eta_t\delta_t - \boldsymbol{w}^* - \eta_t\bar{\delta}_t + \eta_t\bar{\delta}_t||^2 \\ &= \underbrace{||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^* - \eta_t\bar{\delta}_t||^2}_{A_1} \\ &\quad + 2\eta_t \underbrace{\langle \bar{\boldsymbol{w}}_t - \boldsymbol{w}^* - \eta_t\bar{\delta}_t, \bar{\delta}_t - \delta\rangle}_{A_2} + \underbrace{\eta_t^2||\delta_t - \bar{\delta}_t||^2}_{A_3}. \end{aligned} \tag{51}$$

Since $\mathbb{E}[\delta_t] = \bar{\delta}_t$, we know that A_2 becomes zero after taking expectation. We also split A_1 into the three terms as follows:

$$\begin{aligned} A_1 &= ||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^* - \eta_t\bar{\delta}_t||^2 \\ &= ||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2 \underbrace{-2\eta_t\langle\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*, \bar{\delta}_t\rangle}_{B_1} + \underbrace{\eta_t^2||\bar{\delta}_t||^2}_{B_2}. \end{aligned} \tag{52}$$

We now derive an upper bound of B_1. From the definition of $\bar{\boldsymbol{w}}_t$ and $\bar{\delta}_t$, we express B_1 as

$$\begin{aligned} B_1 &= -2\eta_t\langle\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*, \bar{\delta}_t\rangle = -2\eta_t\sum_{k=1}^N p_k\langle\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*, \nabla F_k(\boldsymbol{w}_t^{Q,k})\rangle \\ &= -2\eta_t\sum_{k=1}^N p_k\langle\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^{Q,k}, \nabla F_k(\boldsymbol{w}_t^{Q,k})\rangle \\ &\quad - 2\eta_t\sum_{k=1}^N p_k\langle\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}^*, \nabla F_k(\boldsymbol{w}_t^{Q,k})\rangle. \end{aligned} \tag{53}$$

We first derive an upper bound of $-\langle\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^{Q,k}, \nabla F_k(\boldsymbol{w}_t^{Q,k})\rangle$ using the Cauchy-Schwarz inequality as well as arithmetic mean and geometric mean inequalities as follows:

$$\begin{aligned} &-\langle\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^{Q,k}, \nabla F_k(\boldsymbol{w}_t^{Q,k})\rangle \\ &\leq \frac{1}{\sqrt{\eta_t}}||\boldsymbol{w}_t^{Q,k} - \bar{\boldsymbol{w}}_t||\sqrt{\eta_t}||\nabla F_k(\boldsymbol{w}_t^{Q,k})|| \\ &\leq \frac{1}{2\eta_t}||\boldsymbol{w}_t^{Q,k} - \bar{\boldsymbol{w}}_t||^2 + \frac{\eta_t}{2}||\nabla F_k(\boldsymbol{w}_t^{Q,k})||^2. \end{aligned} \tag{54}$$

We use the assumption of μ-convexity of the loss function to derive an upper bound of $-\langle\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}^*, \nabla F_k(\boldsymbol{w}_t^{Q,k})\rangle$. From the fact that $F_k(\boldsymbol{w}^*) \geq F_k(\boldsymbol{w}_t^{Q,k}) + \langle\boldsymbol{w}^* - \boldsymbol{w}_t^{Q,k}, \nabla F_k(\boldsymbol{w}_t^{Q,k})\rangle + \frac{\mu}{2}||\boldsymbol{w}^* - \boldsymbol{w}_t^{Q,k}||^2$, we have

$$\begin{aligned} -\langle\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}^*, \nabla F_k(\boldsymbol{w}_t^{Q,k})\rangle &\leq -\{F_k(\boldsymbol{w}_t^{Q,k}) - F_k(\boldsymbol{w}^*)\} \\ &\quad - \frac{\mu}{2}||\boldsymbol{w}^* - \boldsymbol{w}_t^{Q,k}||^2. \end{aligned} \tag{55}$$

For B_2, we use L-smoothness of the loss function to obtain the upper bound as below

$$\begin{aligned} B_2 = \eta_t^2||\bar{\delta}_t||^2 &\leq \eta_t^2\sum_{k=1}^N p_k||\nabla F_k(\boldsymbol{w}_t^{Q,k})||^2 \\ &\leq 2L\eta_t^2\sum_{k=1}^N p_k(F_k(\boldsymbol{w}_t^{Q,k}) - F_k^*). \end{aligned} \tag{56}$$

Then, we obtain an upper bound of A_1 using (54), (55), and (56) as follows

$$\begin{aligned} A_1 =& ||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2 - \mu\eta_t \sum_{k=1}^N p_k ||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}^*||^2 \\ &+ \sum_{k=1}^N p_k||\boldsymbol{w}_t^{Q,k} - \bar{\boldsymbol{w}}_t||^2 + \eta_t^2 \sum_{k=1}^N p_k ||\nabla F_k(\boldsymbol{w}_t^{Q,k})||^2 \\ &- 2\eta_t \sum_{k=1}^N p_k \left\{F_k(\boldsymbol{w}_t^{Q,k}) - F_k(\boldsymbol{w}^*)\right\} \\ &+ 2L\eta_t^2 \sum_{k=1}^N p_k \left\{F_k(\boldsymbol{w}_t^{Q,k}) - F_k^*\right\} \\ \le& ||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2 - \mu\eta_t \sum_{k=1}^N p_k ||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}^*||^2 \\ &+\rho\eta_t \sum_{k=1}^N p_k||\boldsymbol{w}_t^{Q,k} - \bar{\boldsymbol{w}}_t||^2 + \underbrace{4L\eta_t^2 \sum_{k=1}^N p_k \left\{F_k(\boldsymbol{w}_t^{Q,k}) - F_k^*\right\}}_{C} \\ & \underbrace{-2\eta_t \sum_{k=1}^N p_k \left\{F_k(\boldsymbol{w}_t^{Q,k}) - F_k(\boldsymbol{w}^*)\right\}}_{C}, \end{aligned} \tag{57}$$

where the last inequality follows from the L-smoothness of the loss function using $||\nabla F_k(\boldsymbol{w}_t^{Q,k})||^2 \le 2L(F_k(\boldsymbol{w}_t^{Q,k}) - F_k^*)$ and $\rho\eta_t \ge 1$ with $\rho \gg 1$. Note that F_k^* is the minimum value of F_k. For $\eta_t \le \frac{1}{2L}$, we can derive the upper bound of C as follows

$$\begin{aligned} C \le& 4L\eta_t^2 \sum_{k=1}^N p_k \left\{F_k(\boldsymbol{w}_t^{Q,k}) - F_k^* - F_k(\boldsymbol{w}_t^{Q,k}) + F_k(\boldsymbol{w}^*)\right\} \\ =& 4L\eta_t^2 \sum_{k=1}^N p_k \Gamma = 4L\eta_t^2 \Gamma. \end{aligned} \tag{58}$$

Then, A_1 can be upper bounded as below

$$\begin{aligned} A_1 \le& ||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2 - \mu\eta_t \sum_{k=1}^N p_k ||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}^*||^2 \\ &+ \rho\eta_t \sum_{k=1}^N p_k ||\boldsymbol{w}_t^{Q,k} - \bar{\boldsymbol{w}}_t||^2 + 4\eta_t^2 L\Gamma. \end{aligned} \tag{59}$$

Next, we derive $||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}^*||^2$ in A_1 as follows

$$\begin{aligned} ||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}^*||^2 =& ||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}_t^k + \boldsymbol{w}_t^k - \boldsymbol{w}^*||^2 \\ =& ||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}_t^k||^2 + ||\boldsymbol{w}_t^k - \boldsymbol{w}^*||^2 \\ &+ 2\langle \boldsymbol{w}_t^{Q,k} - \boldsymbol{w}_t^k, \boldsymbol{w}_t^k - \boldsymbol{w}^* \rangle. \end{aligned} \tag{60}$$

Note that $\langle \boldsymbol{w}_t^{Q,k} - \boldsymbol{w}_t^k, \boldsymbol{w}_t^k - \boldsymbol{w}^* \rangle$ becomes zero after taking expectation due to Lemma 1. Then, we can bound A_1 as follows

$$\begin{aligned} A_1 \le& (1 - \mu\eta_t)||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2 - \mu\eta_t \sum_{k=1}^N p_k ||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}_t^k||^2 \\ &+ \rho\eta_t \sum_{k=1}^N p_k ||\boldsymbol{w}_t^{Q,k} - \bar{\boldsymbol{w}}_t||^2 + 4L\eta_t^2 \Gamma \end{aligned} \tag{61}$$

Now we obtain the expectation of (51) using (61) as follows

$$\begin{aligned} &\mathbb{E}\left[||\bar{\boldsymbol{v}}_{t+1} - \boldsymbol{w}^*||^2\right] \\ &\le (1 - \mu\eta_t)\mathbb{E}\left[||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2\right] + \eta_t^2 \mathbb{E}\left[||\delta_t - \bar{\delta}_t||^2\right] \\ &\quad - \mu\eta_t \sum_{k=1}^N p_k \mathbb{E}\left[||\boldsymbol{w}_t^{Q,k} - \boldsymbol{w}_t^k||^2\right] \\ &\quad + \rho\eta_t \sum_{k=1}^N p_k \mathbb{E}\left[||\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^{Q,k}||^2\right] + 4L\eta_t^2 \Gamma \end{aligned} \tag{62}$$

To further bound (62), we express $\mathbb{E}\left[||\delta_t - \bar{\delta}_t||^2\right]$ as

$$\begin{aligned} \mathbb{E}[||\delta_t - \bar{\delta}_t||^2] =& \sum_{k=1}^N p_k^2 \mathbb{E}\left[\left|\left|\nabla F_k(\boldsymbol{w}_t^{Q,k}, \xi_t^k) - \nabla F_k(\boldsymbol{w}_t^{Q,k})\right|\right|^2\right] \\ \le& \sum_{k=1}^N p_k^2 \sigma_k^2, \end{aligned} \tag{63}$$

where (63) is from $\mathbb{E}[\nabla F_k(\boldsymbol{w}_t^{Q,k}, \xi_t^k)] = \nabla F_k(\boldsymbol{w}_t^{Q,k})$ and the last inequality is from Assumption 1. We also derive the upper bound of $\mathbb{E}\left[||\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^{Q,k}||^2\right]$ as below

$$\begin{aligned} \mathbb{E}\left[||\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^{Q,k}||^2\right] =& \mathbb{E}\Big[||\boldsymbol{w}_t^k - \boldsymbol{w}_t^{Q,k}||^2 + ||\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^k||^2 \\ &+ 2\langle \boldsymbol{w}_t^k - \boldsymbol{w}_t^{Q,k}, \bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^k \rangle\Big] \\ \le& \mathbb{E}\left[||\boldsymbol{w}_t^k - \boldsymbol{w}_t^{Q,k}||^2\right] + 4\eta_t^2(I-1)^2G^2, \end{aligned} \tag{64}$$

where the last inequality is from Lemma 1 and the result of [23] for $\eta_t \le 2\eta_{t+I}$ using

$$\sum_{k=1}^N p_k \mathbb{E}\left[||\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^k||^2\right] \le 4\eta_t^2(I-1)^2G^2. \tag{65}$$

Then, we can obtain Lemma 5 by using (5) in Lemma 1. □

C. Proof of Theorem 1

Since we use quantization in both local training and transmission, we cannot directly use the result of [23] to derive the convergence rate due to the quantization errors. We first define an additional auxiliary variable as done in [19] to prove Theorem 1 as below

$$\boldsymbol{u}_{t+1}^k = \begin{cases} \boldsymbol{v}_{t+1}^k & \text{if} \quad t+1 \notin \mathcal{I}, \\ \frac{1}{K}\sum_{k \in \mathcal{N}_{t+1}} \boldsymbol{v}_{t+1}^k & \text{if} \quad t+1 \in \mathcal{I}. \end{cases} \tag{66}$$

We also define $\bar{\boldsymbol{u}}_t = \sum_{k=1}^N p_k \boldsymbol{u}_t^k$ for convenience. Since we are interested in the result of global iterations, we focus on $t+1 \in \mathcal{I}$. Then, we have

$$\begin{aligned} ||\bar{\boldsymbol{w}}_{t+1} - \boldsymbol{w}^*||^2 =& \underbrace{||\bar{\boldsymbol{w}}_{t+1} - \bar{\boldsymbol{u}}_{t+1}||^2}_{D_1} + \underbrace{||\bar{\boldsymbol{u}}_{t+1} - \boldsymbol{w}^*||^2}_{D_2} \\ &+ \underbrace{2\langle \bar{\boldsymbol{w}}_{t+1} - \bar{\boldsymbol{u}}_{t+1}, \bar{\boldsymbol{u}}_{t+1} - \boldsymbol{w}^* \rangle}_{D_3}. \end{aligned} \tag{67}$$

To simplify (67), we adopt the result of $\bar{\boldsymbol{w}}_{t+1}$ and $\bar{\boldsymbol{u}}_{t+1}$ from [19] as follows:

$$\mathbb{E}[\bar{\boldsymbol{w}}_{t+1}] = \bar{\boldsymbol{u}}_{t+1}, \tag{68}$$

$$\mathbb{E}\left[||\bar{\boldsymbol{w}}_{t+1} - \bar{\boldsymbol{u}}_{t+1}||^2\right] \leq \frac{4d\eta_t^2 IG^2}{K2^{2m}}. \tag{69}$$

Then, we can know that D_3 becomes zero after taking the expectation from (68) and D_1 can be bounded by (69). We further obtain the upper bound D_2 as below

$$\begin{aligned} D_2 = &\underbrace{||\bar{\boldsymbol{u}}_{t+1} - \bar{\boldsymbol{v}}_{t+1}||^2}_{E_1} + \underbrace{||\bar{\boldsymbol{v}}_{t+1} - \boldsymbol{w}^*||^2}_{E_2} \\ &+ \underbrace{2\langle \bar{\boldsymbol{u}}_{t+1} - \bar{\boldsymbol{v}}_{t+1}, \bar{\boldsymbol{v}}_{t+1} - \boldsymbol{w}^* \rangle}_{E_3}. \end{aligned} \tag{70}$$

We leverage the result of the random scheduling from [19] to simplify (70) as follows

$$\mathbb{E}[\bar{\boldsymbol{u}}_{t+1}] = \bar{\boldsymbol{v}}_{t+1} \tag{71}$$

$$\mathbb{E}[\bar{\boldsymbol{v}}_{t+1} - \bar{\boldsymbol{u}}_{t+1}||^2] \leq \frac{4}{K}\eta_t^2 I^2 G^2. \tag{72}$$

We can see that E_3 will vanish due to (71). E_1 and E_2 can be upper bounded by (72) and Lemma 5, respectively. Therefore, we have

$$\begin{aligned} \mathbb{E}\left[||\bar{\boldsymbol{w}}_{t+1} - \boldsymbol{w}^*||\right]^2 &\leq \mathbb{E}\left[||\bar{\boldsymbol{v}}_{t+1} - \boldsymbol{w}^*||\right] + \frac{4\eta_t^2 G^2}{K}\left(\frac{dI}{2^{2m}} + I^2\right) \\ &\leq (1-\mu\eta_t)\mathbb{E}\left[||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2\right] + \eta_t^2\psi_2 + \eta_t\psi_1, \end{aligned} \tag{73}$$

where

$$\begin{aligned} \psi_1 &= \frac{d(\rho-\mu)}{2^{2n}}, \\ \psi_2 &= \sum_{k=1}^{N} p_k^2\sigma_k^2 + 4(I-1)^2G^2 + \frac{4dIG^2}{K2^{2m}} + \frac{4I^2G^2}{K} + 4L\Gamma. \end{aligned} \tag{74}$$

Since $\mathbb{E}\left[||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||\right] \leq \frac{\beta^2\psi_2}{(\beta\mu-1)(t+\gamma)} + \frac{\beta\psi_1}{\beta\mu-1}$ satisfies (74) for $\eta_t = \frac{\beta}{t+\gamma}$ as shown in [23]. Then, we can obtain Theorem 1 from L - smoothness of the loss function using $\mathbb{E}[F(\bar{\boldsymbol{w}}_{t+1}) - F(\boldsymbol{w}^*)] \leq \frac{L}{2}\mathbb{E}\left[||\bar{\boldsymbol{w}}_{t+1} - \boldsymbol{w}^*||\right]^2$. Finally, we change the time scale to local iteration.

References

[1] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, "On the tradeoff between energy, precision, and accuracy in federated quantized neural networks," in *Proc. of IEEE Int. Conf. Commun.*, Seoul, South Korea, May 2022.

[2] 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.

[3] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *arXiv preprint arXiv:1907.10597*, 2019.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2017.

[5] B. Moons, K. Goetschalckx, N. Van Berckelaer, and M. Verhelst, "Minimum energy quantized neural networks," in *Proc. of Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Apr. 2017.

[6] S. Savazzi, V. Rampa, S. Kianoush, and M. Bennis, "An energy and carbon footprint analysis of distributed and federated learning," *arXiv preprint arXiv:2206.10380*, 2022.

[7] 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. of IEEE Conf. on Computer Commun.*, Paris, France, May 2019.

[8] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[9] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with cpu-gpu heterogeneous computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, Dec. 2021.

[10] B. Luo, X. Li, S. Wang, J. Huangy, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. of IEEE Conf. on Computer Commun.*, Vancouver, BC, Canada, May 2021.

[11] R. Balakrishnan, M. Akdeniz, S. Dhakal, and N. Himayat, "Resource management and fairness for federated learning over wireless edge networks," in *Proc. of IEEE Workshop on Signal Process. Advances in Wireless Commun.*, Atlanta, GA, USA, May 2020.

[12] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, Mar. 2021.

[13] P. Liu, J. Jiang, G. Zhu, L. Cheng, W. Jiang, W. Luo, Y. Du, and Z. Wang, "Training time minimization for federated edge learning with optimized gradient quantization and bandwidth allocation," *Frontiers of Information Technology & Electronic Engineering*, vol. 23, no. 8, pp. 1247–1263, 2022 .

[14] C. Feng, Z. Zhao, Y. Wang, T. Q. Quek, and M. Peng, "On the design of federated learning in the mobile edge computing systems," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5902–5916, Sep. 2021.

[15] R. Chen, L. Li, K. Xue, C. Zhang, M. Pan, and Y. Fang, "Energy efficient federated learning over heterogeneous mobile devices via joint design of weight quantization and wireless transmission," *IEEE Trans. Mobile Comput.*, pp. 1–13, Oct. 2022.

[16] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations." *arXiv preprint arXiv:1609.07061*, 2016.

[17] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proc. of International Conference on Machine Learning (ICML)*, Lille, France, Jul. 2015.

[18] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2018.

[19] S. Zheng, C. Shen, and X. Chen, "Design and analysis of uplink and downlink communications for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, Jul. 2021.

[20] B. Moons, D. Bankman, and M. Verhelst, *Embedded Deep Learning, Algorithms, Architectures and Circuits for Always-on Neural Network Processing*. Springer, 2018.

[21] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen, "Rigging the lottery: Making all tickets winners," in *Proc. of International Conference on Machine Learning (ICML)*, Vienna, Austria, Apr. 2020, pp. 2943–2952.

[22] E. Bjornson, E. A. Jorswieck, M. Debbah, and B. Ottersten, "Multiobjective signal processing optimization: The way to balance conflicting metrics in 5g systems," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 14–23, Nov. 2014.

[23] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. of International Conference on Learning Representations (ICLR)*, May 2020.

[24] Z. Yuchen, D. J. C., and W. M. J., "Communication-efficient algorithms for statistical optimization," *J. Mach. Learn. Res.*, vol. 14, no. 1, p. 3321–3363, Jan. 2013.

[25] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM journal on optimization*, vol. 8, no. 3, pp. 631–657, Aug. 1998.

[26] R. Wituła and D. Słota, "Cardano's formula, square roots, chebyshev polynomials and radicals," *Journal of Mathematical Analysis and Applications*, vol. 363, no. 2, pp. 639–647, Feb. 2010.

[27] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[28] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[29] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge University Press, 2011.

[30] E. Larsson and E. Jorswieck, "Competition versus cooperation on the miso interference channel," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 7, pp. 1059–1069, Sep. 2008.

[31] P. Hyunggon and M. van der Schaar, “Bargaining strategies for networked multimedia resource management,” *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3496–3511, Jul. 2007.

[32] R. Yedida, S. Saha, and T. Prashanth, “Lipschitzlr: Using theoretically computed adaptive learning rates for fast convergence,” *Applied Intelligence*, vol. 51, Mar. 2021.

[33] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams, “Variance reduced stochastic gradient descent with neighbors,” in *Proc. of Neural Information Processing Systems (NeurIPS)*, Montreal, Canada, Dec. 2015.

[34] Q. Jin and A. Mokhtari, “Exploiting local convergence of quasi-newton methods globally: Adaptive sample size approach,” in *Proc. of Neural Information Processing Systems (NeurIPS)*, Virtual, Dec. 2021.

[35] A. Øland and B. Raj, “Reducing communication overhead in distributed learning by an order of magnitude (almost),” in *Proc. of IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, South Brisbane, QLD, Australia, 2015, pp. 2219–2223.

[36] Y. Sarikaya and O. Ercetin, “Motivating workers in federated learning: A stackelberg game perspective,” *IEEE Net. Lett.*, vol. 2, no. 1, pp. 23–27, Oct. 2020.

[37] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, “FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization,” in *Proc. of International Conference on Artificial Intelligence and Statistics (AISTATS)*, Virtual Conference, Jun. 2020.