Adaptive Quantization based on Ensemble Distillation to Support FL enabled Edge Intelligence

Yi-Jing Liu*, Shuang Qin*, Gang Feng*, Dusit Niyato†, *Fellow, IEEE*,
Yao Sun‡, Jianhong Zhou§
*National Key Laboratory of Science and Technology on Communications,
and Yangtze Delta Region Institute (Huzhou),
University of Electronic Science and Technology of China
†School of Computer Science and Engineering, Nanyang Technological University
‡James Watt School of Engineering, University of Glasgow
§School of Computer and Software Engineering, Xihua University
E-mail:blueqs@uestc.edu.cn

***Abstract*—Federated learning (FL) has recently become one of the most acknowledged technologies in promoting the development of intelligent edge networks with the ever-increasing computing capability of user equipment (UE). In traditional FL paradigm, local models are usually required to be homogeneous for aggregation to achieve an accurate global model. Moreover, considerable communication cost and training time may be incurred in resource-constrained edge networks due to a large number of UEs participating in model transmission and the large size of transmitted models. Therefore, it is imperative to develop effective training schemes for heterogeneous FL models, while reducing communication cost as well as training time. In this paper, we propose an adaptive quantization scheme based on ensemble distillation (AQeD) for FL to facilitate personalized quantized model training over heterogeneous local models with different size, structure, and quantization level, etc. Specifically, we design an augmented loss function by jointly considering distillation loss function, quantization values and available wireless resources, where UEs train their local personalized machine learning models and send the quantized models to a server. Based on local quantized models, the server first performs global aggregation for cluster ensembles and then sends the aggregated model of the cluster back to the participating UEs. Numerical results show that our proposed AQeD scheme can significantly reduce communication cost as well as training time in comparison with some known state-of-the-art solutions.**

I. Introduction

Federated learning (FL) has been widely acknowledged as one of the most essential enablers to bring edge intelligence into reality, as it facilitates collaborative training of machine learning (ML) models while preserving individual user privacy and data security. However, FL still faces many challenges, especially when deployed at edge networks. Although the transmitted models are lightweight parameters/gradients instead of the raw data, the communication cost incurred in model transmission could be still fairly significant and cannot be ignored. For example, the experimental results in [1] illustrate that the model size of a 5-layer convolutional neural network used for MNIST classification is approximately 4.567MB per global iteration for images with 28×28 pixels. Therefore, it is crucial to develop an adaptive training scheme for FL models with different size, structure, task, etc, while reducing the communication resource consumption for transmitting model updates for FL in wireless edge networks.

Some prior investigations have suggested that quantization is an effective yet efficient method to reduce communication cost and transmission latency by transmitting quantized models instead of the original full-precision ones while maintaining similar learning accuracy [2]–[4]. However, the transmitted local models could be heterogeneous in quantization level and quantization precision, even in the size, structure, task and numerical precision, which makes the implementation of FL in wireless edge networks more challenging. Fortunately, recently proposed ensemble distillation technique could be used for facilitating the collaboration of heterogeneous models by augmenting the local objective with a certain knowledge distillation (KD) loss [5]–[7]. The authors of [5] proposed a quantized and personalized FL algorithm to facilitate personalized model training by introducing ensemble KD loss functions into the local loss function. In [6], the authors proposed an ensemble distillation-based FL framework to reduce wireless resource consumption by considering the diversity of computing nodes. Both the authors of [5] and [6] verified the effectiveness and efficiency of introducing ensemble distillation into the FL framework in edge networks. However, they have not considered the impact of wireless resources and channel quality on quantization levels, which is an essential issue for FL-enabled edge networks as both wireless resource constraints and wireless channel impairments may degrade the learning performance. Therefore, it is imperative to explore an adaptive quantization scheme for heterogeneous FL models while taking into account both the wireless environment and learning algorithm design.

This work was supported by the Key Research and Development Projects under No. 2020YFB1806804, the National Natural Science Foundation under No. 62071091, and the Huawei Cooperation Project under No. TC20210316002.

In this paper, we propose an adaptive quantization scheme based on ensemble distillation, called AQeD, to support FL-enabled edge networks. The main contributions can be summarized as follows: (1) We propose an AQeD scheme to allow the UEs in different clusters to learn quantized personalized models with different quantization levels, model structure, dimensions, and size. (2) We propose an augmented loss function to train an acceptable FL model that can be flexibly quantized based on the available bandwidth resources and channel quality while guaranteeing certain FL performance. (3) We theoretically analyze the convergence property of our proposed AQeD scheme and demonstrate its effectiveness via simulations.

In the rest of this paper, we begin with the system model in Section II. Then we present our proposed AQeD scheme in Section III. In Section IV, the convergence property of our AQeD scheme is analyzed. In Section V, we present the numerical results and conclude the paper in Section VI.

II. System Model

A. FL enabled Edge Networks

We consider an FL-enabled edge network consisting of N UEs which are grouped into U logical clusters and a central edge server co-located with the base station (BS). The UEs in the same cluster may have the same size, task and structure. As shown in Fig. 1, the UEs can be regarded as local computing nodes for local model training, while the edge server serves as the model aggregator [8]. For a specific cluster $u \in U$, let $\mathcal{S}_u = \{n_u^1, \ldots, n_u^i, \ldots, n_u^{k_u}\}$ represent the set of UEs that have the same model in size, structure, and FL task, where n_u^i represents the i-th UE in the u-th cluster and k_u denotes the total number of UEs in the u-th cluster.

B. Quantization Function

Binary/uniform is the most used quantization function that maps individual weights to the closest quantization centers, where the derivative of the function is zero almost everywhere, which discourages the use of gradient-based methods in optimizing the objective with the quantization function. Therefore, to solve this problem, similar to [2], [5], we use a differentiable soft quantization (DSQ) function $Q_{c^t}(\cdot)$ to approximate the uniform quantizer for UE n_u^i at time t, which is given by

$$Q_{c^t}(w_u^i) = \begin{cases} l, w_u^i < l, \\ e, w_u^i > e, \\ l + \Delta\left(a + \frac{\phi(w_u^i)+1}{2}\right), a \in \mathcal{P}_a, \end{cases} \tag{1}$$

where w_u^i represents the model parameters to be quantized of UE n_u^i, c^t means that the bit width is c at time t, and (l, e) represents the original range of w_u^i which is divided into $2^c - 1$ intervals $\mathcal{P}_a, a \in \{0, 1, 2, ..., 2^c - 1\}$. Moreover, $\phi(w_u^i) = a\tanh(k(w_u^i - m_a))$. Specifically, $m_a = l + (a + 0.5)\Delta$, $s = \frac{1}{\tanh(0.5k\Delta)}$, and $\Delta = \frac{e-l}{2^c-1}$. In addition, k represents the coefficient associated with the shape of $\phi(w_u^i)$, where the greater k, the more $\phi(w_u^i)$ behaves like the desired uniform function with multiple quantization levels [2].

C. Communication Model

The main difference between the original model (the local model without quantization) and quantized models is the data volume of the transmitted models. In general, the original local model consists of the training weights that are float 32-bit, which may be quantized to integer c-bit $(1 \le c < 32)$ quantized model [2]. Let $b(Q_{c^t}(w_u^{i,r}) = (1+\log_2(c+1))|w_u^{i,r}|$ represent the volume of the transmitted quantized model of n_u^i, which is a function of the size of the quantized weights (*i.e.*, $|w_u^{i,r}|$) as well as the bit width c [9]. Therefore, based on Shannon's theorem, the wireless bandwidth used to transmit the local quantized model of UE n_u^i during the $r-$th communication round can be given by

$$b_u^{i,r}(Q_{c^t}(w_u^{i,r}) = \frac{b(Q_{c^t}(w_u^{i,r})}{t_u^{i,r} log_2(1 + SINR_u^{i,r})}, \tag{2}$$

where $t_u^{i,r}$ denotes the transmission time and $SINR_u^{i,r}$ denotes the channel Signal-to-Interference-plus-Noise-Ratio.

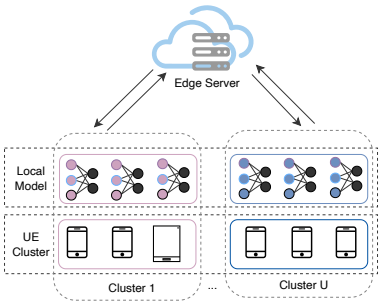

Fig. 1. FL-enabled Edge network with clusters.

D. Quantizated FL Model

1) Loss Function: Let a specific UE $n_u^i \in N$ have a local dataset $\mathcal{D}_u^i$ with D_u^i data samples, where $\mathcal{D}_u^i = \{x_u^{im} \in \mathbb{R}^d, y_u^{im} \in \mathbb{R}\}_{m=1}^{D_u^i}$. We first define $f_m(w_u^i; x_u^{im}, y_u^{im})$ as a loss function for data sample m of UE n_u^i to describe the learning performance. The loss function is different for various FL learning tasks [10]. For example, for a linear regression, the loss function is $f_m(w_u^i; x_u^{im}, y_u^{im}) = \frac{1}{2}(x_u^{im^{\mathrm{T}}} w_u^i - y_u^{im})^2$. For neural network, the loss function could be the mean squared error (*i.e.*, $\frac{1}{n}\sum_{n=1}^{N}(y_u^{im} - \widehat{y}_u^{im})$) where $\widehat{y}_u^{im}$ is the predicted value of y_u^{im}. In this paper, to reduce communication cost, while guaranteeing FL performance of the quantized model, inspired by [5], we define the local loss function $\tilde{f}_u^i(w_u^i(t), Q_{c^t}(w_u^i(t))) : \mathbb{R}^m \to \mathbb{R}$ for the UE n_u^i, as

$$\begin{aligned} &\tilde{f}_u^i(w_u^i(t), Q_{c^t}(w_u^i(t))) \triangleq f_u^i(w_u^i(t)) + f_u^i(Q_{c^t}(w_u^i(t))) \\ &\triangleq \frac{1}{D_u^i} \sum_{(x_u^{im}, y_u^{im}) \in D_u^i} f_m(w_u^i; x_u^{im}, y_u^{im}) + \\ &\qquad \frac{1}{D_u^i} \sum_{(x_u^{im}, y_u^{im}) \in D_u^i} f_m(Q_{c^t}(w_u^i(t)); x_u^{im}, y_u^{im}). \end{aligned} \tag{3}$$

Moreover, based on the local loss function, we define $\tilde{f}_u(w_u, Q_{c^t}(w_u))$ as the cluster loss function on all associated

distributed datasets to measure the learning performance of the cluster, *i.e.*,

$$\tilde{f}_u(w_u, Q_{c^t}(w_u)) \triangleq \frac{\sum_{i=1}^{k_u} D_u^i \tilde{f}_u^i(w_u^i(t), Q_{c^t}(w_u^i(t)))}{D_u}, \quad (4)$$

where w_u represents the model of cluster u, $D_u \triangleq \sum_{i=1}^{k_u} D_u^i$ denotes the total number of data samples in cluster u. Furthermore, we define $\tilde{f}(w, c)$ as the global loss function among all clusters, *i.e.*,

$$\tilde{f}(w, c) \triangleq \frac{\sum_{u=1}^{U} D_u \tilde{f}_u(w_u, Q_{c^t}(w_u))}{D}, \quad (5)$$

where $D \triangleq \sum_{u=1}^{U} D_u$. The goal of the edge server is to fit vectors ω and $\mathcal{C}$ for all UEs so as to minimize $\tilde{f}(w, c)$, *i.e.*, $\{\omega, \mathcal{C}\}^* \triangleq \arg_{w,c} \min \tilde{f}(w, Q_{c^t}(w))$.

III. Adaptive Quantization based on Ensemble Distillation

In this section, we first define ensemble distillation loss functions to bridge the gap between the heterogeneous model and the global model. Then we present the proposed AQeD scheme by combining distillation loss and quantization values.

A. Ensemble Distillation

As there is no explict expression for the original model $w_u^i(t)$, the quantized model $Q_{c^t}(w_u^i(t))$ and the global model g_u^t, we define two separated knowledge distillation loss fuctions (*i.e.*, $f_{u,i}^{\text{KD}}(w_u^i(t), g_u^t)$, $f_{u,i}^{\text{KD}}(Q_{c^t}(w_u^i(t)), g_u^t)$) based on Kullback-Leibler (KL) divergence [5], [11] to respectively represent the gap between $w_u^i(t)$ and g_u^t as well as the gap between $Q_{c^t}(w_u^i(t))$ and g_u^t. Specifically, $f_{u,i}^{\text{KD}}(w_u^i(t), g_u^t)$ is to ensure that the behavior of the model without quantization (*i.e.*, $w_u^i(t)$) is close to the behavior of the global model (*i.e.*, g_u^t), which is given by

$$f_{u,i}^{\text{KD}}(w_u^i(t), g_u^t) = \sum_{u=1}^{U} p_{g_u^t}(u) \log \frac{p_{g_u^t}(u)}{p_{w_u^i(t)}(u)}, \quad (6)$$

while $f_{u,i}^{\text{KD}}(Q_{c^t}(w_u^i(t)), g_u^t)$ is to ensure that the behavior of the quantized mode (*i.e.*, $Q_{c^t}(w_u^i(t))$) is close to the behavior of the global model, given by

$$f_{u,i}^{\text{KD}}(Q_{c^t}(w_u^i(t)), g_u^t) = \sum_{u=1}^{U} p_{g_u^t}(u) \log \frac{p_{g_u^t}(u)}{p_{c^t}(u)}, \quad (7)$$

where $u \in U$ is the index of clusters and it also represents the class across all local models. Moreover, $p_{w_u^i(t)}(u) = \frac{exp(\frac{\sum_{i=1}^{k_u} f_u^i(w_u^i(t))}{\text{Tem}})}{\sum_{u \in U} exp(\frac{\sum_{i=1}^{k_u} f_u^i(w_u^i(t))}{\text{Tem}})}$ represents the probability of the samples that belong to class u for the original model. $_{c^t}(u) = \frac{exp(\frac{\sum_{i=1}^{k_u} f_u^i(Q_{c^t}(w_u^i(t)))}{\text{Tem}})}{\sum_{u \in U} exp(\frac{\sum_{i=1}^{k_u} f_u^i(Q_c^t(w_u^i(t)))}{\text{Tem}})}$ represents the probability of the samples that belong to class u for the quantized model. $p_{g_u^t}(u) \frac{exp(\frac{f_u(g_u^t)}{\text{Tem}})}{\sum_{u \in U} exp(\frac{f_u(g_u^t)}{\text{Tem}})}$ denotes the corresponding value for the cluster model. Note that Tem denotes a temperature. A higher value of Tem, a softer probability distribution over clusters.

B. Problem Formulation

In traditional FL-enabled edge networks, we always formulate the optimization problem to minimize the loss function under the network resource and learning performance constraints, as follows:

$$\min_{w,c} \quad \frac{\sum_{u=1}^{U} \sum_{i=1}^{k_u} D_u^i \tilde{f}_u^i}{D} \quad (8)$$

$$\text{s.t.} \quad b_u^{i,r}(Q_{c^t}(w_u^i(t))) \le B_u^{i,r}, \forall r \times i \times u \in R \times k_u \times U, \quad (8.1)$$

$$f_{u,i}^{\text{KD}}(w_u^i, g_u^t) + f_{u,i}^{\text{KD}}(Q_{c^t}(w_u^i), g_u^t) \le k, \forall r \times i \times u \in R \times k_u \times U, \quad (8.2)$$

where (8.1) represents the bandwidth constraint, which means that the bandwidth used for transmitting the local model of

Algorithm 1 : AQeD Algorithm.

Input: η_1, η_2, η_3; λ_1, λ_2; R; τ; c^0; $w_u^i(0)$; $Q_{c^t}(w_u^i(0))$.
output: Quantized models $Q_{c^T}(w_u^i(T))$.

for $t = 0$ to $T-1$ **do**
  if $t \mod \tau \neq 0$ **then**
    for $u = 1$ to U **do**
      Parallel Each UE $i = 1, 2, ..., k_u$
      Compute $w_u^i(t) = \nabla_{w_u^i(t)} f_u^i(w_u^i(t)) + \nabla_{w_u^i(t)} f_u^i(Q_{c^t} w_u^i(t)) + \lambda_1 \nabla_{Q_{c^t}(w_u^i(t))} b(Q_{c^t}(w_u^i(t))) + \lambda_2 \nabla_{w_u^i(t)} f_{u,i}^{\text{KD}}(w_u^i(t), g_u^t) + \lambda_2 \nabla_{w_u^i(t)} f_{u,i}^{\text{KD}}(Q_{c^t}(w_u^i(t)), g_u^t))$ and $\tilde{w}_u^i(t+1) = \tilde{w}_u^i(t) - \eta_1 w_u^i(t)$
      Compute $h_i^t = \nabla_{c_i^t} f_u^i(Q_{c^t} w_u^i(t)) + \lambda_1 \nabla_{c_i^t} b(Q_{c^t}(w_u^i(t))) + \lambda_2 \nabla_{c_i^t} f_{u,i}^{\text{KD}}(Q_{c^t}(w_u^i(t)), g_u^t)$ and $c_i^{t+1} = c_i^t - \eta_2 h_i^t$
      $\tilde{w}_u^i(t+1) = \tilde{w}_u^i(t) - \eta_3 \lambda_2 (\nabla_{w_u^i(t)} f_{u,i}^{\text{KD}}(\tilde{w}_u^i(t+1), w_u^i(t)) + \nabla_{w_u^i(t)} f_{u,i}^{\text{KD}}(Q_{c^{t+1}}(\tilde{w}_u^i(t+1), w_u^i(t)))$
    end for
    if $t \mod \tau = 0$ **then**
      Parallel Each UE sends $Q_{c^{t+1}}(\tilde{w}_u^i(t))$ to the server
    end if
  end if
  if $t \mod \tau = 0$ **then**
    On Server do:
    Compute $g_u^t = \frac{\sum_{i=1}^{k_u} D_u^i Q_{c^{t+1}}(\tilde{w}_u^i(t))}{D_u}$
    Compute $\tilde{f}(Q_c^t(w_u^t), g_u^t)) = \frac{\sum_{u=1}^{U} \sum_{i=1}^{k_u} D_u^i f_{u,i}^{KD}(Q_c^t(w_u^i), g_u^t)}{D}$
    Server sends g_u^t to UEs
    Server sends $\tilde{f}(Q_c^t(w_u^t), g_u^t))$ to UEs
    On Each UE do:
    Receive g_u^t and $\tilde{f}(Q_c^t(w_u^t), g_u^t))$ from the server
    Set $w_u^i(t+1) = g_u^t$
    Set $f_{u,i}^{KD}(Q_c^{t+1}(w_u^i(t+1)), g_u^t) = \tilde{f}(Q_c^t(w_u^t), g_u^t))$
  end if
end for
output Quantized model $Q_{c^T}(w_u^i(T))$ for each UE.

each UE cannot exceed the maximal available bandwidth of the BS that can be allocated to the UE. (8.2) means that some difference between the original model, quantized model, and distillation model is allowed to some extent. However, due to the dynamic nature of the edge network, the computational complexity incurred by searching the optimal quantization level among heterogeneous FL models could be too high and the changes of FL training (*e.g.*, global aggregation) may not be accurately described in training process. To solve this problem, we propose an AQeD scheme to train a certain FL model that can be flexibly quantized based on the available bandwidth resources and channel quality while guaranteeing the acceptable FL performance among heterogeneous models.

C. AQeD Scheme

We first re-write problem (8) by using the Lagrangian method, as follows:

$$\min_{w_u^i, c} \tilde{F}(w_u^i(t), Q_{c^t}(w_u^i(t)), g_u^t) = \frac{\sum_{u=1}^{U}\sum_{i=1}^{k_u} D_u^i}{D}\{\tilde{f}_u^i(w_u^i(t), Q_{c^t}(w_u^i(t)), g_u^t) + \lambda_1(b_u^{i,r}(Q_{c^t}(w_u^i(t))) - B_u^{i,r}) + \lambda_2(f_{u,i}^{\mathrm{KD}}(w_u^i(t), g_u^t) + f_{u,i}^{\mathrm{KD}}(Q_{c^t}(w_u^i(t)), g_u^t) - k)\}. \quad (9)$$

Note that problem (8) could be equivalent to problem (9), as we can always find the available multipliers λ_1 and λ_2 for problem (9) to approximate the optimal solution of problem (8) [12]. In other words, we can find the optimal solution of problem (8) via problem (9). In this paper, with the aim to obtain an adaptive FL quantized model based on the available bandwidth resources and channel quality while guaranteeing learning performance among heterogeneous models, we propose AQeD scheme based on FL to solve problem (9) by introducing a number of local and global iterations. According to Problem (9), we define the global augmented loss function as $\tilde{F}(w_u^i(t), Q_{c^t}(w_u^i(t)), g_u^t)$. From $\tilde{F}(w_u^i(t), Q_{c^t}(w_u^i(t)), g_u^t)$, we can obtain the local loss function as $\tilde{f}_u^i(w_u^i(t), Q_{c^t}(w_u^i(t)), g_u^t) + \lambda_1(b_u^{i,r}(Q_{c^t}(w_u^i(t))) - B_u^{i,r}) + \lambda_2(f_{u,i}^{\mathrm{KD}}(w_u^i(t), g_u^t) + f_{u,i}^{\mathrm{KD}}(Q_{c^t}(w_u^i(t)), g_u^t) - k)$.

In the AQeD scheme, similar to the traditional FL, a number of local and global model update iterations are required for minimizing the global augmented loss function and achieving certain trained model accuracy. Here each global iteration is called a *communication round* [8], which consists of local model updating, local model quantization, local model transmission, global model aggregation and global model transmission. Shown as Algorithm 1, in the local updating process, UEs perform three stochastic gradient descent (SGD)) steps to update the original model, quantized model, and global model respectively. Specifically, the first SGD step is for updating the original model w_u^i, given as follows:

$$w_u^i(t+1) = w_u^i(t) - \eta_1 \nabla_{w_u^i(t)} f_u^i(w_u^i(t)) - \nabla_{w_u^i(t)} f_u^i(Q_{c^t} w_u^i(t)) - \lambda_1 \nabla_{Q_{c^t}(w_u^i(t))} b(Q_{c^t}(w_u^i(t))) - \lambda_2 \nabla_{w_u^i(t)} f_{u,i}^{\mathrm{KD}}(w_u^i(t), g_u^t) - \lambda_2 \nabla_{w_u^i(t)} f_{u,i}^{\mathrm{KD}}(Q_{c^t}(w_u^i(t)), g_u^t)).$$

The second SGD step is for updating the quantization level c, which is given by

$$c_i^{t+1} = c_i^t - \eta_2(\nabla_{c_i^t} f_u^i(Q_{c^t} w_u^i(t)) + \lambda_1 \nabla_{c_i^t} b(Q_{c^t}(w_u^i(t))) + \lambda_2 \nabla_{c_i^t} f_{u,i}^{\mathrm{KD}}(Q_{c^t}(w_u^i(t)), g_u^t))).$$

The third SGD step is for bridging the gap between different clusters, which is given by

$$\tilde{w}_u^i(t+1) = \tilde{w}_u^i(t) - \eta_3 \lambda_2(\nabla_{w_u^i(t)} f_{u,i}^{\mathrm{KD}}(\tilde{w}_u^i(t+1), w_u^i(t)) + \nabla_{w_u^i(t)} f_{u,i}^{\mathrm{KD}}(Q_{c^{t+1}}(\tilde{w}_u^i(t+1), w_u^i(t))).$$

When $t \mod \tau = 0$, UEs quantize $\tilde{w}_u^i(t)$ to $Q_{c^t}(\tilde{w}_u^i(t))$ and sends the local quantized models $Q_{c^t}(\tilde{w}_u^i(t))$ to the edge server. Then the cluster aggregation is performed at the edge server according to $g_u^t = \frac{\sum_{i=1}^{k_u} D_u^i Q_{c^t}(w_u^i(t))}{D_u}$ and $\tilde{f}(Q_c^t(w_u^t), g_u^t)) = \frac{\sum_{u=1}^{U}\sum_{i=1}^{k_u} D_u^i f_{u,i}^{KD}(Q_c^t(w_u^i), g_u^t)}{D}$, after which the edge server send the global model g_u^t and $\tilde{f}(Q_c^t(w_u^t), g_u^t))$ back the the UEs.

IV. Convergence Analysis

To facilitate the convergence analysis, we first make the following assumptions for the function $f_u^i(\cdot)$ and the soft quantizer $Q_{c^t}(\cdot)$, respectively, *i.e.*,

- Assumption 1: Function $f_u^i(x)$ is L-smooth, twice-continuously differentiable, and bounded: *i.e.*, $\forall x, y \in R^d$, $||\nabla f_u^i(x) - \nabla f_u^i(y)|| \leq L||x-y||$, $||\nabla^2 f(x)|| \leq LI, \forall x \in R^d$, and $f_u^i(x) > -\infty$. In addition, $\nabla f_u^i(x)$ is bounded: $\forall x \in R^d$, $||\nabla f_u^i(x)|| \leq G$, where $G < +\infty$.
- Assumption 2: $Q_{c^t}(x)$ is l_{Q_1}-Lipschitz and L_{Q_1}-smooth with respect to x: $\forall c \in R^m, \forall x, y \in R^d$, $||Q_{c^t}(x) - Q_{c^t}(y)|| \leq l_{Q_1}||x-y||$ and $||\nabla_x Q_{c^t}(x) - \nabla_y Q_{c^t}(y)|| \leq L_{Q_1}||x-y||$.
- Assumption 3: $Q_{c^t}(x)$ is l_{Q_2}-Lipschitx, L_{Q_2}-smooth and twice-continuously differentiable with respect to c, *i.e.*, $\forall c, d \in R^m$, $||Q_{c^t}(x) - Q_{d^t}(x)|| \leq l_{Q_2}||c-d||$, $||\nabla_c Q_{c^t}(x) - \nabla_d Q_{d^t}(x)|| \leq L_{Q_2}||c-d||$, and $||\nabla^2_{Q_{c^t}} f(Q_{c^t}(x))|| \leq L_{Q_2}$.

Proposition 1. *$\tilde{f}_u^i(x, Q_{c^t}(x))$ is $GL_{Q_2} + G_{Q_2} L l_{Q_2}$- smooth with respect to c.*

Proof:

$$\begin{aligned}
&||\nabla_c f(Q_{c^t}(x)) - \nabla_d f(Q_d(x))|| = \\
&= ||\nabla_{Q_{c^t}(x)} f(Q_{c^t}(x)) \nabla_c Q_{c^t}(x) - \\
&\nabla_{Q_{c^t}(x)} f(Q_{c^t}(x)) \nabla_d Q_d(x) + \nabla_{Q_{c^t}(x)} f(Q_{c^t}(x)) \nabla_d Q_d(x) - \\
&\qquad \nabla_{Q_d(x)} f(Q_d(x)) \nabla_d Q_d(x)|| \\
&\leq ||\nabla_{Q_{c^t}(x)} f(Q_{c^t}(x))|| \cdot ||\nabla_c Q_{c^t}(x) - \nabla_d Q_d(x)|| + \\
&\quad ||\nabla_d Q_d(x)|| \cdot ||\nabla_{Q_{c^t}(x)} f(Q_{c^t}(x)) - \nabla_{Q_d(x)} f(Q_d(x))|| \\
&\leq (GL_{Q_2} + G_{Q_2} L l_{Q_2})||c-d||.
\end{aligned}$$

■

Proposition 2. *$\tilde{f}_u^i(x, Q_{c^t}(x))$ is $L + GL_{Q_1} + G_{Q_1} L l_{Q_1}$- smooth with respect to x.*

Proof: Similar to that in Proposition 1. ■

According to A.6 in [5], both $f_{u,i}^{\mathrm{KD}}(w_u^i, g_u^t)$ and $f_{u,i}^{\mathrm{KD}}(Q_{c^t}(w_u^i), g_u^t)$ are smooth functions. Specifically, $f_{u,i}^{\mathrm{KD}}(w_u^i, g_u^t)$ is L_{D_1}-smooth with respect to w_u^i for a positive constant L_{D_1}, and L_{D_2}-smooth with respect to g_u^t for a

positive constant L_{D_2}. Moreover, $f_{u,i}^{\mathrm{KD}}(Q_{c^t}(w_u^i), g_u^t)$ is L_{DQ_1}-smooth with respect to w_u^i, L_{DQ_2}-smooth with respect to c, and L_{DQ_3} with respect to g_u^t for some positive constants L_{DQ_1}, L_{DQ_2}, and L_{DQ_3}. In addition, we assume that the function $b(Q_{c^t}(w_u^i))$ is also L_b-smooth with respect to w_u^i and L_{DQ_b}-smooth with respect to c. Therefore, we have the following Proposition 3.

Proposition 3. *Local loss function* $\tilde{f}_u^i(x, Q_{c^t}(x), g_u^t) + \lambda_1(b(Q_{c^t}(x)) - B_u^{i,r}) + \lambda_2(f_{u,i}^{KD}(x, g_u^t) + f_{u,i}^{KD}(Q_{c^t}(x), g_u^t) - k)$ *is* $(L + GL_{Q_1} + G_{Q_1}Ll_{Q_1}) + \lambda_1 L_b + \lambda_2(L_{D_1} + L_{DQ_1})$*-smooth with respect to* x*, and* $(GL_{Q_2} + G_{Q_2}Ll_{Q_2}) + \lambda_1 L_{DQ_b} + \lambda_2 L_{DQ_2}$*-smooth with respect to* c*.*

Proof: According to the fact that if two functions f_1 and f_2 are L_1-smooth and L_2-smooth respectively, $f_1 + f_2$ is $L_1 + L_2$-smooth. ∎

Moreover, we assume that at any $t \in \{0, 1, ..., T-1\}$, there exists k_i meeting $||g_u^{t+1} - g_u^t||^2 \leq k_i$. and $||\nabla_{g_u^t}\tilde{F}(w_u^i(t+1), Q_{c^{t+1}}(w_u^i(t+1)), g_u^t) - \frac{1}{N}\sum_{j=1}^{N}\nabla_{g_u^t}\tilde{F}(w_u^j(t+1), Q_{c^{t+1}}(w_u^j(t+1)), g_u^t)||^2 \leq k_i$.

Therefore, we derive Proposition 4, as follows:

Proposition 4. *Considering running Algorithm 1 for* T *iterations under the bandwidth and learning performance constraints for minimizing* $\tilde{F}_u^i(w_u^i, Q_{c^t}(w_u^i), g_r)$ *with* $\tau \leq T$, $\eta_1 = \frac{1}{(L + GL_{Q_1} + G_{Q_1}Ll_{Q_1}) + \lambda_1 L_b + \lambda_2(L_{D_1} + L_{DQ_1})}$, $\eta_2 = \frac{1}{(GL_{Q_2} + G_{Q_2}Ll_{Q_2}) + \lambda_1 L_{DQ_b} + \lambda_2 L_{DQ_2}}$, $\eta_3 = \frac{1}{4\lambda_2\sqrt{C_L}\sqrt{T}(L_{D_2} + L_{DQ_3})}$. *Let* $G_i^t := [\nabla_{w_u^i(t+1)}\tilde{F}(w_u^i(t+1), Q_{c^t}(w_u^i(t+1)), g_u^t)^T, \nabla_{c^{t+1}}\tilde{F}(w_u^i(t+1), Q_{c^{t+1}}(w_u^i(t+1)), g_u^t)^T, \nabla_{g_u^t}\tilde{F}(w_u^i(t+1), Q_{c^{t+1}}(w_u^i(t+1)), g_u^t)^T]^T$. *Then, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{N}\sum_{i=1}^{N}||G_i^t||^2 = \mathcal{O}\left(\frac{\tau^2\overline{k} + \overline{\Delta}F}{\sqrt{T}} + \tau^2\overline{k}\left(\frac{C_1}{T} + \frac{C_2}{T^{\frac{3}{2}}}\right) + \overline{k}\right),$$

where C_1, C_2 *are constants,* $\overline{\Delta}F = \frac{1}{N}\sum_{i=1}^{N}(L_{max}^{(i)})^2(\tilde{F}(w_u^i(0), Q_{c^0}(w_u^i(0)), g_0) - \tilde{F}(w_u^i(T), Q_{c^T}(w_u^i(T)), g_u^t))$, $C_L = 1 + \frac{\frac{1}{N}\sum_{i=1}^{N}(L_{max}^{(i)})^2}{(\min_i L_{max}^{(i)})^2}$, $\overline{k} = \frac{1}{N}\sum_{i=1}^{N}(L_{max}^{(i)})^2 k_i$, *and* $L_{max}^{(i)} = \max\{\frac{1}{2}, (L + GL_{Q_1} + G_{Q_1}Ll_{Q_1}) + \lambda_1 L_b + \lambda_2 L_{D_1} + \lambda_2 L_{DQ_1}, (GL_{Q_2} + G_{Q_2}Ll_{Q_2}) + \lambda_1 L_{DQ_b} + \lambda_2 L_{DQ_2}\}$.

Proof: The process is similar to Section 7.2.3 of [5]. The main difference is that we start with the second-order Taylor expansion of $\tilde{F}(w_u^i(t), Q_{c^t}(t), g_u^t)$ and the derivation result is based on Proposition 1-3. ∎

Convergence Result. The result in Proposition 4 achieves a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$ for finding a stationary point (w_u^i, c), when we minimize $\tilde{F}_u^i(w_u^i, Q_{c^t}(w_u^i), g_r)$ in problem (12) via Algorithm 1 for T iterations.

V. Simulations

In this section, we verify our proposed AQeD scheme using numerical simulations by (1) measuring the performance of FL settings, and (2) examining training time and bandwidth consumption. Two existing schemes are used as comparison reference: 1) GCFL: general multi-cluster (task) federated learning without quantization [13], and 2) FL8Q: multi-task FL with 8 bits quantization.

A. Simulation Settings

We consider an FL-enabled wireless network composed of two UE clusters and one BS with a cloud server as the FL model aggregator. The coverage of the BS is a circular area with a radius of 500m. The transmit power of UEs, the serving BS, and the noise power are set to 20dBm, 43dBm, and -173dBm, respectively [8]. The path loss is modeled as $g(D_1) = 34 + 40\log(D_1)$ [8] and the total amount of available wireless resources is set to 20 MHz [8]. Please note that the UEs are randomly generated in each cluster and only the UEs in the same cluster can contribute the interference to each other. Moreover, we set $\lambda_1 = 0.2$ and $\lambda_2 = 0.15$ [5].

We consider a multi-class classification task over MNIST datasets [14], where all datasets of UEs are randomly divided with 75%-25%, for training and testing respectively [15]. Moreover, we use a convolutional neural network (CNN) built over Pytorch (Python 3.8). The CNN structure for each cluster is randomly chosen in: 1) CNN1 with 2 convolution layers and 3 fully-connected layers. Specifically, the first and the second convolution layers are with 16 and 32 channels respectively, where each layer follows with 2×2 max pooling. Moreover, the fully-connected layer has 320 units where the activation function is ReLU [16]. 2) CNN2. CNN1 with an additional convolutional layer with 32 filters and 5×5 Kernal size [5]. 3) CNN3 still with 2 convolution layers and 3 fully-connected layers. The first and the second convolution layers with 32 and 64 channels respectively (2×2 max pooling), the fully-connected layer with 512 units (the activation function is ReLU) [16]. In addition, inspired by the hyperparameter analysis and the corresponding experimental results in [5], [17], we set learning rate $\eta_1 = 0.01$, $\eta_2 = 10^{-4}$, and $\eta_3 = 0.5$.

B. Simulation Results

We first verify the convergency property of our proposed AQeD scheme, while comparing the training accuracy with the other two schemes including GCFL and FL8Q. In this simulation, the number of epochs is set to 80, the number of UEs in each cluster is set to 300, and the number of samples on each UE is randomly chosen within $[200, 1200]$. Fig. 2 shows the training accuracy of the three schemes changes with the number of communication rounds. From Fig. 2, we can see that the AQeD converges faster than the other two schemes. Specifically, the training accuracy converges in 10 communication rounds for AQeD, 16 communication rounds for GCFL and 20 communication rounds for FL8Q. In addition, we find that the training accuracy of our proposed AQeD is close to that of GCFL and much higher than that of FL8Q.

Next, we examine the training loss of the three schemes. Fig. 3 shows the training loss decreases with epochs. From Fig. 3, again we see that our proposed AQeD always converges fast. This is because we introduce two distillation loss functions into

local training to ensure the behavior of local models is close to that of the global model. In addition, from Fig. 2 and Fig. 3, we can see that the training loss/accuracy of GCFL is always better than that of AQeD and FL8Q schemes when the trend of the curve converges, as quantization causes the model loss.

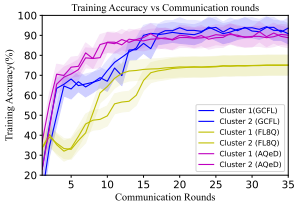

Fig. 2. Comparison of training accuracy.

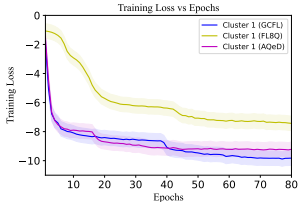

Fig. 3. Comparison of loss value.

After that, we compare the training time of the three schemes. Fig. 4 shows the training time changes with the number of UEs. From Fig. 4, we can see that the training time of AQeD is always lower than that of GCFL and FL8Q. This is because the local model training is based on the global model and distillation loss aggregation of heterogeneous models.

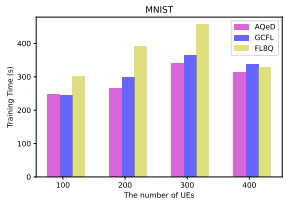

Fig. 4. Comparison of training time.

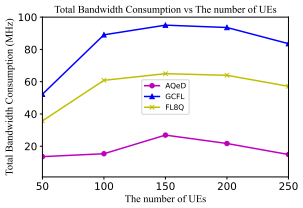

Fig. 5. Comparison of bandwidth consumption.

Finally, we examine the total bandwidth consumption of the three schemes. Fig. 5 shows that the total bandwidth consumption changes with the number of UEs. From Fig. 5, we can see that the bandwidth consumption increases in the beginning and then decreases with the number of UEs. Specifically, the total bandwidth consumption increases with the number of UEs when it is approximately below 150. When the number of UEs is approximately more than 150, the total bandwidth consumption decreases as poor channel quality causes some UEs to fail in transmitting the local models. In addition, we can see that our proposed AQeD scheme significantly outperforms GCFL and FL8Q while FL8Q significantly outperforms GCFL in terms of bandwidth consumption, as both AQeD and FL8Q introduce quantization technology while AQeD is more adaptive than FL8Q.

VI. Conclusion

In this paper, with aim to reduce communication cost while guaranteeing learning performance over heterogeneous models, we have proposed a novel adaptive quantization scheme based on ensemble distillation, called AQeD, by designing an augmented global loss function. Moreover, we have theoretically analyzed the convergence property of our AQeD scheme. Numerical results show that our proposed AQeD scheme can achieve a significant performance improvement in terms of training time and bandwidth consumption when compared with the state-of-the-art algorithms. In the future, we will continue to explore effective and efficient FL-based schemes to solve model heterogeneity as well as resource heterogeneity problems, which is still a challenging issue in wireless networks.

References

[1] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device Machine Learning: Federated Distillation and Augmentation under Non-iid Private Data," *arXiv preprint arXiv:1811.11479*, 2018.

[2] R. Gong, X. Liu, S. Jiang, T. Li, P. Hu, J. Lin, F. Yu, and J. Yan, "Differentiable Soft Quantization: Bridging Full-precision and Low-bit Neural Networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4852–4861.

[3] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via Gradient Quantization and Encoding," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1709–1720, 2017.

[4] K. Ozkara, N. Singh, D. Data, and S. Diggavi, "QuPeL: Quantized Personalization with Applications to Federated Learning," *arXiv preprint arXiv:2102.11786*, 2021.

[5] O. Kaan, S. Navjot, D. Deepesh, and D. Suhas, "QuPeD: Quantized Personalization via Distillation with Applications to Federated Learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[6] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. Doermann, and A. Innanje, "Ensemble Attention Distillation for Privacy-Preserving Federated Learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 076–15 086.

[7] L. Tao, K. Lingjing, S. S. U, and J. Martin, "Ensemble Distillation for Robust Model Fusion in Federated Learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2351–2363, 2020.

[8] Y. Liu, G. Feng, Y. Sun, S. Qin, and Y.-C. Liang, "Device association for ran slicing based on hybrid federated deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 731–15 745, 2020.

[9] 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," *arXiv preprint arXiv:2112.14387*, 2021.

[10] C. Hennig and M. Kutlukaya, "Some Thoughts About the Design of Loss Functions," *REVSTAT–Statistical Journal*, vol. 5, no. 1, pp. 19–39, 2007.

[11] Z. Allen-Zhu and Y. Li, "Towards Understanding Ensemble, Knowledge Distillation and Self-distillation in Deep Learning," *arXiv preprint arXiv:2012.09816*, 2020.

[12] R. Brooks and A. Geoffrion, "Finding Everett's Lagrange multipliers by linear programming," *Operations Research*, vol. 14, no. 6, pp. 1149–1153, 1966.

[13] S. Virginia, C. Chao-Kai, S. Maziar, and T. A. S, "Federated Multi-task Learning," *Advances in neural information processing systems*, vol. 30, 2017.

[14] Y. LeCun, "The MNIST Database of Handwritten Digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[16] M. Brendan, M. Eider, R. Daniel, H. Seth, and y Arcas Blaise Aguera, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[17] C. Darken and J. E. Moody, "Note on Learning Rate Schedules for Stochastic Optimization," in *Proceedings of the 4th International Conference on Neural Information Processing Systems*, vol. 91, 1990, pp. 832–838.