Dynamic Scheduling for Over-the-Air Federated Edge Learning with Energy Constraints

Yuxuan Sun, *Member, IEEE,* Sheng Zhou, *Member, IEEE,*
Zhisheng Niu, *Fellow, IEEE,* Deniz Gündüz, *Senior Member, IEEE*

***Abstract*—Machine learning and wireless communication technologies are jointly facilitating an intelligent edge, where federated edge learning (FEEL) is emerging as a promising training framework. As wireless devices involved in FEEL are resource limited in terms of communication bandwidth, computing power and battery capacity, it is important to carefully schedule them to optimize the training performance. In this work, we consider an over-the-air FEEL system with analog gradient aggregation, and propose an energy-aware dynamic device scheduling algorithm to optimize the training performance within the energy constraints of devices, where both communication energy for gradient aggregation and computation energy for local training are considered. The consideration of computation energy makes dynamic scheduling challenging, as devices are scheduled before local training, but the communication energy for over-the-air aggregation depends on the l_2-norm of local gradient, which is known only after local training. We thus incorporate estimation methods into scheduling to predict the gradient norm. Taking the estimation error into account, we characterize the performance gap between the proposed algorithm and its offline counterpart. Experimental results show that, under a highly unbalanced local data distribution, the proposed algorithm can increase the accuracy by 4.9% on CIFAR-10 dataset compared with the myopic benchmark, while satisfying the energy constraints.**

***Index Terms*—Federated edge learning, over-the-air computation, energy constraints, dynamic scheduling, Lyapunov optimization.**

I. Introduction

Many emerging applications at the wireless edge, such as autonomous driving, virtual reality and Internet of things (IoT), are powered by modern machine learning (ML) techniques. Data-driven approaches also penetrate into the wireless network itself for channel estimation, encoding and decoding, resource allocation, etc. [2], [3]. The complex ML models for these applications need to be trained over massive data, while data samples are usually generated by edge devices. Centralized training methods can hardly be competent, as collecting data at one location would create network congestion, lead to extremely high transmission cost and may cause privacy concerns. On the other hand, computing capabilities of base stations (BSs) and edge devices, such as mobile phones, smart vehicles and IoT sensors, are becoming increasingly powerful, enabling intensive computations at the edge. In this context, federated learning (FL) is considered as a promising training framework that can exploit distributed data and computational resources with limited communication and privacy leakage [4], [5]. In FL, multiple devices train a shared model collaboratively with local data, and a central *parameter server (PS)* coordinates the training process.

The limited communication resource and non-independent and identically distributed (i.i.d.) data, i.e., the distribution of local data at one device is not identical with that of other devices or the global data, are the two major challenges in FL [6], [7]. Current methods to improve the communication efficiency include model compression [10]–[14], device scheduling [15], [16], and enabling multiple local iterations [13], [17], [18]. Under non-i.i.d. data, the training performance can be improved by sharing global i.i.d. data with devices [7] or the PS [19], introducing data redundancy [1], or scheduling devices based on their importance [16].

In a wireless network, FL can be carried out among wireless edge devices coordinated by a BS, called federated edge learning (FEEL). In FEEL, participating devices are often resource limited in terms of wireless bandwidth, computing capability and battery capacity. A key issue is to design device scheduling and resource allocation algorithms that optimize the training performance. Considering the communication energy constraints, an energy-efficient bandwidth allocation policy is proposed to maximize the fraction of scheduled devices in [20], while an online algorithm is designed to maximize the sum utility of scheduling in [21]. In [22], scheduling decisions are based on both the channel states and the importance of local updates. Due to the timeliness requirements of FEEL tasks at the wireless edge [23], training delay is another key performance metric. The total communication delay for training is minimized in [24], while the sum delay for computation and communication is minimized in [25]. Communication delay is combined with the importance of each update for probabilistic scheduling in [26]. A hierarchical FEEL framework is proposed in [27], where the training delay is minimized by optimizing the update interval and model compression. The trade-off between the total energy for communication and computation and the training delay is further considered in [28]–[30], yielding a joint design of local

Y. Sun, S. Zhou and Z. Niu are with the Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: sunyuxuan@tsinghua.edu.cn, sheng.zhou@tsinghua.edu.cn, niuzhs@tsinghua.edu.cn).

D. Gündüz is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2BT, UK (e-mail: d.gunduz@imperial.ac.uk).

Y. Sun, S. Zhou and Z. Niu are sponsored in part by the National Key R&D Program of China No. 2020YFB1806605, by the Nature Science Foundation of China (No. 62022049, No. 61871254, No. 91638204, No. 61861136003), by the China Postdoctoral Science Foundation No. 2020M680558, and Hitachi Ltd. (Corresponding author: Sheng Zhou)

D. Gündüz received funding from the European Research Council (ERC) under Starting Grant BEACON (grant No. 677854).

Part of this work has been presented in IEEE ICC 2020 [1].

computation speed and wireless resource allocation.

The literature above mainly focuses on the implementation of FEEL via digital wireless communications. However, the unique communication requirement of FEEL, i.e., the PS only needs the *average* of local model updates rather than each individual vector, makes the separate design of learning and communication protocol highly suboptimal [31]. A new solution called *over-the-air computation* is facilitated to further improve the communication efficiency [32]–[36], which is achieved by synchronizing the devices to transmit their local gradients or models in an analog fashion, and exploiting the superposition property of a wireless multiple access channel (MAC) to do the summation over-the-air. It is shown in [33] that over-the-air FEEL can reduce the latency of model aggregation by $O\left(\frac{N}{\log_2 N}\right)$ compared with the digital counterpart, where N is the number of devices. Power limits of devices can highly degrade the training performance, which yields the design of power allocation schemes over noisy channels [34], fading channels [35] and broadband fading channels [33]. Power control algorithms that take into account the importance of updates [37], uplink and downlink noise [38], [39], gradient statistics [40] and non-i.i.d. data [41] are further proposed.

While over-the-air computation has stringent requirements on synchronization and channel state information (CSI), there are also efforts to relax them for practical implementations. For misaligned signals, a whitened matched filtering and sampling scheme is proposed in [42]. It is shown in [43] that with multiple antennas, over-the-air computation can be realized with imperfect CSI. Based on one-bit gradient quantization and majority voting, a digital realization of over-the-air FEEL is further proposed in [44].

Existing papers on over-the-air FEEL mainly consider average power constraints for communication, but have not considered the computation energy for local model training, which is in fact non-negligible for edge devices. In this work, we aim to optimize the training performance under total energy constraints of devices by designing an energy-aware dynamic device scheduling algorithm, where energy is consumed for both communication and computation. The introduction of computation energy makes the scheduling decisions challenging due to the *causality of decision making and energy consumption*. This is because, in over-the-air FEEL, the communication energy of each device for gradient aggregation depends on the l_2-norm of its local gradient estimate, which can only be obtained *after computation*. However, online scheduling decision should be made at the start of each training round *before computation*.

Note that this issue does not arise in the existing work [28]–[30] that jointly considers communication and computation energy for FEEL with digital communication. The reason is that the transmit power of digital communication can be chosen independently of the local update. Without computation energy, this challenge does not arise in the power allocation problem [1], [33]–[35] for over-the-air FEEL system either.

The main contributions of this work include:

1) We characterize the convergence bound of the considered over-the-air FEEL system, taking into account the noise in the wireless channels and the variance of stochastic gradients. Based on the convergence analysis, we formulate a device scheduling problem to optimize the training performance under the total energy budget of each device, where both the communication energy for gradient aggregation and the computation energy for local gradient calculation are included.

2) We propose an estimated-drift-plus-penalty algorithm for energy-aware dynamic device scheduling based on Lyapunov optimization. A virtual energy queue is constructed at each device to indicate the up-to-date energy deficit and enable online decision making. Estimation methods to predict the l_2-norm of the local gradients are further incorporated in the proposed scheduling algorithm, in order to address the key challenge that communication energy is unknown when the device scheduling decisions are made.

3) We provide a theoretical performance guarantee for the proposed dynamic scheduling algorithm, by comparing its worst-case training performance and energy consumption with the offline optimal solution. Our analysis further shows the impact of energy estimation error on the performance bound.

4) Experiments on MNIST and CIFAR-10 datasets validate that the proposed dynamic device scheduling algorithm can achieve higher model accuracies compared with the myopic benchmark, while satisfying the energy limits. Under a highly-non-i.i.d. scenario, the accuracy can be increased by 4.9%. The impact of design parameters on the training performance and energy consumption are also evaluated to provide guidelines for practical implementations.

The rest of this paper is organized as follows. In Section II, we introduce the system model and problem formulation. In Section III, we carry out convergence analysis. The energy-aware dynamic device scheduling algorithm is developed in Section IV with its performance guarantee. Experimental results are shown in Section V, and conclusions are given in Section VI.

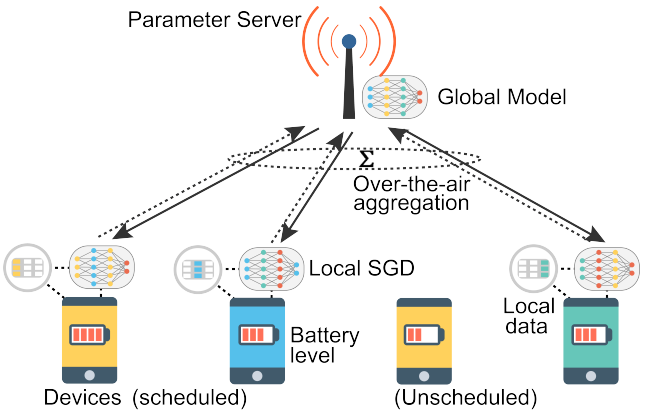


Fig. 1. Illustration of the considered over-the-air FEEL system.

II. System Model and Problem Formulation

A. System Overview

As shown in Fig. 1, we consider a FEEL system with one PS and N devices, denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$. Each device $n \in \mathcal{N}$ has a local dataset $\mathcal{D}_n$ with D_n data samples, and the

global dataset is denoted by $\mathcal{D} = \bigcup_{n=1,\ldots,N} \mathcal{D}_n$ with $D \triangleq \sum_{n=1}^{N} D_n$ data samples.

Given a single data sample $\boldsymbol{x} \in \mathcal{D}$, a loss function $f(\boldsymbol{w}, \boldsymbol{x})$ is used to measure the fitting performance of an s-dimensional model vector $\boldsymbol{w} \in \mathbb{R}^s$. At device $n \in \mathcal{N}$, the local loss function $F_n(\boldsymbol{w})$ is defined as the the average loss over local data samples, i.e.,

$$F_n(\boldsymbol{w}) \triangleq \frac{1}{D_n} \sum_{\boldsymbol{x} \in \mathcal{D}_n} f(\boldsymbol{w}, \boldsymbol{x}). \tag{1}$$

The goal of the FEEL task is to train a shared global model $\boldsymbol{w}$ that minimizes the global loss function $F(\boldsymbol{w})$, which is defined as

$$F(\boldsymbol{w}) \triangleq \frac{1}{D} \sum_{n=1}^{N} \sum_{\boldsymbol{x} \in \mathcal{D}_n} f(\boldsymbol{w}, \boldsymbol{x}) = \sum_{n=1}^{N} \frac{D_n}{D} F_n(\boldsymbol{w}). \tag{2}$$

Under the coordination of the PS, the FEEL system iterates the following three steps until the termination condition is satisfied: 1) the PS broadcasts the up-to-date global model to a subset of devices, which are scheduled to participate in the current training process; 2) the scheduled devices compute their local gradients with local datasets; and 3) the PS aggregates the local gradients over a wireless MAC and updates the global model. Each iteration consisting of these three steps is called a training round, which is indexed by t in the following. The termination conditions that are commonly used for FEEL include the convergence of the global model, or reaching a preset maximum number of training rounds. Since we consider an energy-limited wireless scenario, we set the total number of training rounds to T.

B. Local Gradient Computation

At the start of the t-th training round, the PS schedules a subset of devices $\mathcal{B}_t \subseteq \mathcal{N}$, and broadcasts the global model vector $\boldsymbol{w}_{t-1}$ obtained in the last round to these scheduled devices. Let $\beta_{n,t} \in \{0, 1\}$ be an indicator, with $\beta_{n,t} = 1$ if device n is scheduled to participate in the t-th training round, and $\beta_{n,t} = 0$ otherwise. Thus $\mathcal{B}_t = \{n | \beta_{n,t} = 1, n \in \mathcal{N}\}$. We also assume that the broadcast of $\boldsymbol{w}_{t-1}$ is error-free since the PS is a more capable node with sufficient power.

Each scheduled device $n \in \mathcal{B}_t$ computes the local gradient estimate $\tilde{\boldsymbol{g}}_{n,t}$ by running the stochastic gradient descent (SGD) algorithm on a local mini-batch $\mathcal{L}_{n,t} \subseteq \mathcal{D}_n$, according to

$$\tilde{\boldsymbol{g}}_{n,t} = \frac{1}{L_b} \sum_{\boldsymbol{x} \in \mathcal{L}_{n,t}} \nabla f\left(\boldsymbol{w}_{t-1}, \boldsymbol{x}\right), \tag{3}$$

where $L_b = |\mathcal{L}_{n,t}|$ is the batch size, and $\mathcal{L}_{n,t}$ is uniformly selected at random from the local dataset $\mathcal{D}_n$. We remark here that, following [26], [35], [39] and for simplicity, a single-iteration gradient update is considered in this work, but the proposed online scheduling framework can be extended to a more general case where multiple local iterations are carried out in each training round. Also note that, the selection of batch size is in general an empirical matter in ML. In this work, we consider the batch size as a hyper-parameter rather than an optimization variable, and set to an identical value across devices.

Local computation energy mainly depends on the complexity of the ML model, the computation frequency, as well as the batch size [29], [30]. Given the ML model, we can estimate the computation workloads in floating point operations (FLOPs) for gradient calculation. Meanwhile, we assume that the computation frequency is fixed at each device. Accordingly, for device n, the computation energy for calculating the local gradient on a single data sample is denoted by a constant e_n. The computation energy consumption $E_{n,t}^{[\text{cp}]}$ at device n in round t is given by

$$E_{n,t}^{[\text{cp}]} = e_n L_b. \tag{4}$$

C. Gradient Aggregation Over-the-Air

We assume that the devices transmit their local gradients over a noisy wireless MAC in an analog fashion for global gradient aggregation, and the PS and devices each has a single antenna. To enable the summation of local gradients over-the-air, transmissions are synchronized across all the scheduled devices, and the transmit power of each device is aligned with the others. To be specific, let $h_{n,t}$ be the wireless channel gain between device n and the PS, which is assumed to remain constant during one transmission period. Note that, the device scheduling policy designed in this work is applicable to arbitrary channel models. Moreover, as local gradient computation takes time and the wireless channel is time variant, the channel gain observed at the start of each training round may not be precise. The observation error, i.e., the difference between the observed channel gain that determines the device scheduling and its true value during transmission, will also be considered in the following. Let σ_t be the power scalar that determines the received SNR at the PS. Then the transmit power $p_{n,t}$ of each scheduled device $n \in \mathcal{B}_t$ is set to

$$p_{n,t} = \frac{\sigma_t}{h_{n,t}}, \tag{5}$$

and $p_{n,t}\tilde{\boldsymbol{g}}_{n,t}$ is transmitted from device n to the PS [33], [35]. In this way, local models can be summed over-the-air. The communication energy consumption $E_{n,t}^{[\text{tr}]}$ at device n in round t is then given by

$$E_{n,t}^{[\text{tr}]} = \left\| p_{n,t}\tilde{\boldsymbol{g}}_{n,t} \right\|_2^2 = \frac{\sigma_t^2}{h_{n,t}^2} \left\| \tilde{\boldsymbol{g}}_{n,t} \right\|_2^2, \tag{6}$$

where $\|\boldsymbol{x}\|_2$ represents the l_2-norm of vector $\boldsymbol{x}$. Therefore, if device n is scheduled in the t-th round, the total energy consumption $E_{n,t}$ for computation and communication is

$$E_{n,t} = E_{n,t}^{[\text{tr}]} + E_{n,t}^{[\text{cp}]} = \frac{\sigma_t^2}{h_{n,t}^2} \left\| \tilde{\boldsymbol{g}}_{n,t} \right\|_2^2 + e_n L_b. \tag{7}$$

Note that, our model can be extended to the case with multiple channels with minor changes, where the local gradient should be equally partitioned and transmitted in parallel through these channels [33], [35]. Then the communication energy of each device is the sum of that over all the channels.

At the PS, the received signal $\boldsymbol{y}_t$ is given by

$$\boldsymbol{y}_t = \sum_{n \in \mathcal{B}_t} h_{n,t} p_{n,t} \tilde{\boldsymbol{g}}_{n,t} + \boldsymbol{z}_t = \sigma_t \sum_{n \in \mathcal{B}_t} \tilde{\boldsymbol{g}}_{n,t} + \boldsymbol{z}_t, \tag{8}$$

where $\boldsymbol{z}_t \in \mathbb{R}^s$ is an additive white Gaussian noise vector, in which each entry is i.i.d. and follows Gaussian distribution with zero mean and variance σ_0^2.

In FEEL, we aim to update the global model vector $\boldsymbol{w}_t$ according to

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \eta_t \frac{\sum_{n\in\mathcal{B}_t} \tilde{\boldsymbol{g}}_{n,t}}{|\mathcal{B}_t|}, \tag{9}$$

where η_t is the learning rate in the t-th training round, and $|\cdot|$ denotes the cardinality of a set. Due to channel noise, the actual global model is updated according to

$$\begin{aligned}\boldsymbol{w}_t &= \boldsymbol{w}_{t-1} - \eta_t \frac{\boldsymbol{y}_t}{\sigma_t |\mathcal{B}_t|} \\ &= \boldsymbol{w}_{t-1} - \eta_t \left(\frac{\sum_{n\in\mathcal{B}_t} \tilde{\boldsymbol{g}}_{n,t}}{|\mathcal{B}_t|} + \frac{\boldsymbol{z}_t}{\sigma_t |\mathcal{B}_t|} \right). \end{aligned} \tag{10}$$

D. Problem Formulation

Given the total number of training rounds T and the initial global model vector $\boldsymbol{w}_0$, we aim to minimize the expected global loss $\mathbb{E}[F(\boldsymbol{w}_T)]$ under the energy constraints of devices, by optimizing the device scheduling $\{\beta_{n,t}\}$ and power scalar $\{\sigma_t\}$. The expectation $\mathbb{E}[F(\boldsymbol{w}_T)]$ is taken over the randomness of channel noise and data sampling for local SGD. The problem is formulated as

$$\mathcal{P}1: \min_{\{\sigma_t,\ \beta_{n,t}\}_{t=1}^T} \quad \mathbb{E}[F(\boldsymbol{w}_T)] \tag{11a}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} \beta_{n,t} E_{n,t} \le \bar{E}_n,\ \forall n, \tag{11b}$$

$$\sigma_t > 0,\ \beta_{n,t} \in \{0,1\},\ \forall n, t. \tag{11c}$$

In the first constraint (11b), $\bar{E}_n$ represents the total energy budget of device n, and the inequality indicates that for each device, the total energy consumption for both local gradient computation and wireless communication over T training rounds cannot exceed its given budget. The second constraint limits the ranges of optimization variables.

Based on the law of telescoping sums, problem $\mathcal{P}1$ can be re-written as

$$\mathcal{P}2: \min_{\{\sigma_t,\ \beta_{n,t}\}_{t=1}^T} \quad \sum_{t=1}^{T} \mathbb{E}[F(\boldsymbol{w}_t)] - \mathbb{E}[F(\boldsymbol{w}_{t-1})] \tag{12}$$

$$\text{s.t.} \quad \text{constraints (11b), (11c).}$$

There are three major challenges to solve problem $\mathcal{P}2$:

1) The inexplicit form of the objective function: Since the neural network architectures for ML might be deep and diverse, and the evolution of the model vector is very complex during the training process, it is hard to express $\mathbb{E}[F(\boldsymbol{w}_T)]$ or $\mathbb{E}[F(\boldsymbol{w}_t)] - \mathbb{E}[F(\boldsymbol{w}_{t-1})]$ in closed form. Therefore, we need an approximation of the objective function in closed-form, which can be obtained based on the convergence analysis.

2) The unavailability of future information: The optimal solution to $\mathcal{P}2$ requires the system state information of all rounds at the very beginning of training. Such information is unavailable in practice. Thus we aim to design an online scheduling algorithm, in which the scheduling decision is made at the start of each round with only the current states.

3) The causality of decision making and energy consumption: A unique characteristic of over-the-air FEEL is that the communication energy depends on the computation result to be transmitted. That is, the communication energy in (6) depends on the l_2-norm of local gradient through $\left\|\tilde{\boldsymbol{g}}_{n,t}\right\|_2^2$, which can only be acquired *after* computing the gradient in each round. However, online device scheduling decision should be made *before* gradient computation, in order not to consume computation energy at unscheduled devices, or even not to transmit global updates to these devices. As a result, the exact energy consumption in the current training round is unknown upon decision making. Note that this issue does not arise in the case of digital communication as the transmission power can be chosen independently of the local update. Moreover, the channel gain $h_{n,t}$ observed at the start of each round may not be precise depending on the computation delay.

To address these challenges, we first substitute the objective function with its upper bound based on the convergence analysis in Section III. Then in Section IV, we design an online device scheduling algorithm based on Lyapunov optimization, where the unknown instantaneous states for decision making, including the l_2-norm of local gradients and the wireless channel gains, are substituted with their estimates, and in particular, the impact of the estimation error on the performance of the proposed algorithm is analyzed.

III. Convergence Analysis and Problem Transformation

In this section, we analyze the convergence rate of the considered FEEL system, mainly to investigate how the optimization variables affect the training performance. We seek an upper bound for the objective function in problem $\mathcal{P}2$, based on which we are able to transform the original optimization problem to an approximate one with explicit expressions.

For the simplicity of notation, we define the local full gradient on device n in the t-th round as $\boldsymbol{g}_{n,t} \triangleq \nabla F_n(\boldsymbol{w}_{t-1}) = \frac{1}{D_n}\sum_{\boldsymbol{x}\in\mathcal{D}_n} \nabla f(\boldsymbol{w}_{t-1}, \boldsymbol{x})$, the global full gradient in round t as $\boldsymbol{g}_t \triangleq \nabla F(\boldsymbol{w}_{t-1}) = \sum_{n=1}^N \frac{D_n}{D}\boldsymbol{g}_{n,t}$, and the optimum loss as $F^* \triangleq \min_{\boldsymbol{w}\in\mathbb{R}^s} F(\boldsymbol{w})$.

To facilitate the convergence analysis, we make the following assumptions according to the state-of-the-art literature, including [10]–[13], [17], [33]–[35], [44], etc.

Assumption 1. *Stochastic gradient is unbiased and variance-bounded, i.e., for any device n and training round t, taking the expectation over stochastic data sampling, we have*

$$\mathbb{E}_{\boldsymbol{x}_n}\left[\nabla f\left(\boldsymbol{w}_{t-1}, \boldsymbol{x}_n\right)\right] = \mathbb{E}_{\mathcal{L}_{n,t}}\left[\tilde{\boldsymbol{g}}_{n,t}\right] = \boldsymbol{g}_t, \tag{13}$$

$$\mathbb{E}_{\boldsymbol{x}_n}\left[\left\|\nabla f\left(\boldsymbol{w}_{t-1}, \boldsymbol{x}_n\right) - \boldsymbol{g}_t\right\|_2^2\right] \le G^2, \forall n, t, \tag{14}$$

where $\boldsymbol{x}_n \in \mathcal{D}_n$ is a data sample, $\mathcal{L}_{n,t} \subseteq \mathcal{D}_n$ is a stochastic mini-batch, and G is a constant.

Assumption 2. *Loss functions $F_1(\boldsymbol{w}), \ldots, F_N(\boldsymbol{w})$ are l-smooth, i.e., for $\forall \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^s$ and $n \in \mathcal{N}$,*

$$F_n(\boldsymbol{v}) - F_n(\boldsymbol{w}) \le \nabla F_n^T(\boldsymbol{w})(\boldsymbol{v} - \boldsymbol{w}) + \frac{l}{2}\left\|\boldsymbol{v} - \boldsymbol{w}\right\|_2^2. \tag{15}$$

Assumption 3. *Loss functions $F_1(\boldsymbol{w}), \ldots, F_N(\boldsymbol{w})$ are μ-strongly convex, i.e., for $\forall \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^s$ and $n \in \mathcal{N}$,*

$$F_n(\boldsymbol{v}) - F_n(\boldsymbol{w}) \geq \nabla F_n^T(\boldsymbol{w})(\boldsymbol{v} - \boldsymbol{w}) + \frac{\mu}{2} \|\boldsymbol{v} - \boldsymbol{w}\|_2^2. \quad (16)$$

A. Convergence Analysis

Based on the assumptions above, we provide a single-round convergence guarantee in the following lemma by characterizing the upper bound of $\mathbb{E}[F(\boldsymbol{w}_t)] - \mathbb{E}[F(\boldsymbol{w}_{t-1})]$. Our convergence analysis jointly considers the transmit power and channel noise in the over-the-air FEEL system, as well as the variance of the stochastic gradients.

Lemma 1. *Given the global model vector $\boldsymbol{w}_{t-1}$ and the set of scheduled devices $\mathcal{B}_t$ at the beginning of round t, the single-round convergence is upper-bounded by*

$$\mathbb{E}[F(\boldsymbol{w}_t)] - \mathbb{E}[F(\boldsymbol{w}_{t-1})] \leq -\eta_t \left(1 - \frac{l\eta_t}{2}\right) \|\boldsymbol{g}_t\|_2^2 + \frac{l\eta_t^2}{2} \left(\frac{G^2}{L_b|\mathcal{B}_t|} + \frac{\sigma_0^2 s}{\sigma_t^2 |\mathcal{B}_t|^2}\right), \quad (17)$$

where the expectation is taken over the randomness of channel noise and SGD.

Proof. See Appendix A. □

According to Lemma 1, we can see that the number of devices $|\mathcal{B}_t|$ scheduled in each round makes a key contribution to the convergence rate of training. The power scalar σ_t should also be chosen carefully to reduce the impact of noise while satisfying the energy constraints of devices.

Based on Lemma 1, the convergence performance of over-the-air FEEL after T training rounds is given in the following theorem.

Theorem 1. *Given the global model vector $\boldsymbol{w}_0$ and any device scheduling sequence $\{\mathcal{B}_t, t = 1, \ldots, T\}$, after T rounds of training,*

$$\mathbb{E}[F(\boldsymbol{w}_T)] - F^* \leq (\mathbb{E}[F(\boldsymbol{w}_0)] - F^*) \prod_{i=1}^{T} (1 - \mu\eta_i) + \sum_{i=1}^{T-1} A_i \prod_{j=i+1}^{T} (1 - \mu\eta_i) + A_T, \quad (18)$$

where $A_t \triangleq \frac{\eta_t}{2} \left(\frac{G^2}{L_b|\mathcal{B}_t|} + \frac{\sigma_0^2 s}{\sigma_t^2 |\mathcal{B}_t|^2}\right)$ and the learning rate satisfies $\eta_t \leq \min\{\frac{1}{l}, 1\}, \ \forall t$.

Proof. See Appendix B. □

We remark here that, instead of maximizing the weighted average number of devices scheduled over time as in the existing papers [1], [20], [21], we provide a more reasonable objective function based on Lemma 1 and Theorem 1 in the following, which directly minimizes the upper bound on $\mathbb{E}[F(\boldsymbol{w}_t)] - \mathbb{E}[F(\boldsymbol{w}_{t-1})]$. Another remark is that, we make the three assumptions above to enable the convergence analysis. In particular, Assumption 1 indicates i.i.d. local data, and Assumption 3 assumes strong convexity of loss functions. While the algorithm we propose in the following is based on the obtained convergence behavior, it can also work well under non-i.i.d. data and non-convex loss functions, as being validated in the experiments in Section V.

B. Problem Transformation

As discussed in Section II-D, the objective function $\sum_{t=1}^{T} \mathbb{E}[F(\boldsymbol{w}_t)] - \mathbb{E}[F(\boldsymbol{w}_{t-1})]$ in problem $\mathcal{P}2$ cannot be expressed explicitly. Let

$$U_t \triangleq \frac{l\eta_t^2}{2} \left(\frac{G^2}{L_b \sum_{n=1}^{N} \beta_{n,t}} + \frac{\sigma_0^2 s}{\sigma_t^2 \left(\sum_{n=1}^{N} \beta_{n,t}\right)^2}\right), \quad (19)$$

where we recall that σ_t and $\beta_{n,t}$ are the power scalar and worker scheduling indicator, respectively. To make the optimization problem tractable, we substitute the objective function with its convergence bound according to Lemma 1, and formulate an alternative optimization problem:

$$\mathcal{P}3: \min_{\{\sigma_t, \ \beta_{n,t}\}_{t=1}^{T}} \sum_{t=1}^{T} -\eta_t \left(1 - \frac{l\eta_t}{2}\right) \|\boldsymbol{g}_t\|_2^2 + U_t \quad (20)$$
$$\text{s.t.} \quad \text{constraints (11b), (11c).}$$

Moreover, due to the unavailability of future system states, we aim to design an online algorithm to solve problem $\mathcal{P}3$, and ignore the impact of current decision on the future system states. As the global full gradient $\boldsymbol{g}_t$ defined on the whole dataset is fixed given the global model vector $\boldsymbol{w}_{t-1}$ at the start of training round t, and the learning rate η_t and smoothness parameter l are hyper-parameters, the first term in (20) is a constant. Therefore, we ignore this term and transform the optimization problem to

$$\mathcal{P}4: \min_{\{\sigma_t, \beta_{n,t}\}_{t=1}^{T}} \sum_{t=1}^{T} U_t \quad (21)$$
$$\text{s.t.} \quad \text{constraints (11b), (11c).}$$

IV. Energy-Aware Dynamic Device Scheduling Algorithm

In this section, we propose an energy-aware dynamic device scheduling algorithm that solves problem $\mathcal{P}4$ in an online fashion. To address the challenge brought by the causality of decision making and communication energy consumption, we first propose two heuristics to estimate the l_2-norm of local gradient estimates. Then, we design an online scheduling algorithm based on Lyapunov optimization, and characterize the worst-case performance of the proposed algorithm, which takes the error of energy estimation into consideration. Finally, we provide some practical considerations for real implementations.

A. Estimating the l_2-Norm of Local Gradients

We propose two heuristics in the following to estimate the l_2-norm of local gradients $\left\|\tilde{\boldsymbol{g}}_{n,t}\right\|_2^2$ at the start of each training round t.

1) Compute the l_2-norm of local gradients with a smaller mini-batch (EST-C):

An additional step is introduced at the start of each training round. To be specific, the PS broadcasts the up-to-date global model $\boldsymbol{w}_{t-1}$ to all the devices. Then, each device randomly selects a mini-batch $\mathcal{L}'_{n,t} \subseteq \mathcal{D}_n$ with batch size L_e to calculate a local gradient estimate $\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}$:

$$\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]} = \frac{1}{L_e} \sum_{\boldsymbol{x} \in \mathcal{L}'_{n,t}} \nabla f\left(\boldsymbol{w}_{t-1}, \boldsymbol{x}\right). \tag{22}$$

The computation energy should be modified as $E_{n,t}^{[\text{cp}]} = \beta_{n,t} e_n (L_b - L_e) + e_n L_e$.

We further assume that each device can upload the value of $\left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2^2$ to the PS with negligible cost, which is used as the estimation of the l_2-norm of local gradient for device n in round t.

Following similar proof of Lemma 1 as (39), the exact value of gradient norm and its estimation have the bounded expectations as follows:

$$\begin{aligned} \mathbb{E}\left[\left\|\tilde{\boldsymbol{g}}_{n,t}\right\|_2^2\right] &\leq \left\|\boldsymbol{g}_{n,t}\right\|_2^2 + \frac{G^2}{L_b}, \\ \mathbb{E}\left[\left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2^2\right] &\leq \left\|\boldsymbol{g}_{n,t}\right\|_2^2 + \frac{G^2}{L_e}. \end{aligned} \tag{23}$$

As L_b is typically much larger than L_e, the expressions above indicate that the estimation may suffer from a large deviation due to the gradient variance.

2) Estimate with past information (EST-P):

A simpler and more straightforward way is to use the most recent l_2-norm of local gradient to estimate the current one at each device. Let $t_n = \arg\max_t \{t | \beta_{n,t} = 1\}$ be the most recent round in which device n is scheduled. The estimated l_2-norm of the current local gradient estimate is:

$$\left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2^2 = \left\|\tilde{\boldsymbol{g}}_{n,t_n}\right\|_2^2. \tag{24}$$

We will show in Fig. 2 and Fig. 3 in Section V that, under our considered datasets, estimation by EST-P is more accurate due to the strong temporal correlation of gradients, while the EST-C method does not perform well due to the high variance of the stochastic gradients. Moreover, compared with EST-C that requires additional computation and communication, EST-P method is computation-free and only needs each device to report the l_2-norm of local gradients when it is scheduled. Therefore, we use EST-P to estimate the l_2-norm of local gradient in the following device scheduling algorithm. On the other hand, if the variance of the stochastic gradient is small while the temporal correlation of the gradient norm is weak, the EST-C method can be incorporated.

B. Energy-Aware Dynamic Device Scheduling Algorithm

To enable online scheduling without any future information while satisfying the total energy constraints of devices, we construct a virtual queue $q_{n,t}$ for each device n to indicate the gap between the cumulative energy consumption till round t and the budget, evolved as

$$q_{n,t+1} = \max\left\{q_{n,t} + \beta_{n,t} E_{n,t} - \frac{\bar{E}_n}{T}, 0\right\}, \tag{25}$$

with initial value $q_{n,1} = 0$, $\forall n \in \mathcal{N}$.

Recall that the causality of device scheduling and energy consumption leads to the unawareness of $E_{n,t}$ at the start of round t. Based on the estimated l_2-norm of local gradient $\left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2^2$ by EST-P and the wireless channel gain $\tilde{h}_{n,t}$ observed at the beginning of round t, the estimated energy consumption of device n at round t, denoted by $\tilde{E}_{n,t}$, is given by

$$\tilde{E}_{n,t} = \frac{\sigma_t^2}{\tilde{h}_{n,t}^2} \left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2^2 + e_n L_b. \tag{26}$$

Inspired by the drift-plus-penalty algorithm of Lyapunov optimization [45], the online scheduling aims to solve the following problem:

$$\mathcal{P}5: \min_{\{\sigma_t,\ \beta_{n,t}\}} V U_t + \sum_{n=1}^{N} \beta_{n,t} q_{n,t} \tilde{E}_{n,t} \tag{27a}$$

$$\text{s.t.} \quad \sigma_t > 0,\ \beta_{n,t} \in \{0,1\},\ \forall n, \tag{27b}$$

where V is an adjustable weight parameter to balance the loss U_t and energy consumption, and recall that $U_t = \frac{l\eta_t^2}{2}\left(\frac{G^2}{L_b \sum_{n=1}^N \beta_{n,t}} + \frac{\sigma_0^2 s}{\sigma_t^2 \left(\sum_{n=1}^N \beta_{n,t}\right)^2}\right)$.

Compared to the classical drift-plus-penalty algorithm where all the states in the current round are known, the drift term $q_{n,t}\tilde{E}_{n,t}$ in $\mathcal{P}5$ is an approximation, and thus we call it *estimated-drift-plus-penalty* algorithm.

Notice that problem $\mathcal{P}5$ is a mixed integer non-linear programming problem, which is still very difficult to solve. Meanwhile, existing work has shown that the convergence performance of FEEL with over-the-air gradient aggregation is not very sensitive to the power scalar σ_t, as long as the received SNR or the power limit of each device is larger than a threshold [34]. Therefore, we further decouple the optimization variables in $\mathcal{P}5$ by considering the power scalar σ_t as a hyper-parameter, and then develop the optimal solution to the online device scheduling problem.

1) Received SNR and Power Scalar σ_t

The power scalar σ_t is chosen as follows. In the t-th round, the expected received SNR at the PS side is denoted by γ_t, given by

$$\gamma_t = \mathbb{E}\left[\frac{\left\|\sigma_t \sum_{n\in\mathcal{B}_t} \tilde{\boldsymbol{g}}_{n,t}\right\|_2^2}{\left\|\boldsymbol{z}_t\right\|_2^2}\right] = \frac{\sigma_t^2}{\sigma_0^2 s}\left\|\sum_{n\in\mathcal{B}_t} \tilde{\boldsymbol{g}}_{n,t}\right\|_2^2. \tag{28}$$

Let γ_0 be a pre-defined SNR threshold. The power scalar is set according to

$$\sigma_t = \frac{\gamma_0 \sigma_0^2 \sqrt{s}}{\min_{n\in\mathcal{N}} \left\|\tilde{\boldsymbol{g}}_{n,t}\right\|_2} \approx \frac{\gamma_0 \sigma_0^2 \sqrt{s}}{\min_{n\in\mathcal{N}} \left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2}, \tag{29}$$

such that the expectation of the received SNR can meet the threshold γ_0 even in the worst case when a single device is scheduled. Recall that $\left\|\tilde{\boldsymbol{g}}_{n,t}\right\|_2$ is unknown and thus approximated by $\left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2$ according to the EST-P method.

2) Optimal Online Device Scheduling

Algorithm 1 Optimal Online Device Scheduling to $\mathcal{P}6$

1: Sort $\mathcal{C}_t = \left\{q_{n,t}\tilde{E}_{n,t}, \forall n\right\}$ and let $C_t^{[m]}$ be the m-th smallest value of $\mathcal{C}_t$.
2: **for** $k = 1, \ldots, N$ **do**
3: Calculate $v_t(k)$ according to (31).
4: **end for**
5: Get $k^* = \arg\min_k \{v_t(k) \mid k = 1, \ldots, N\}$.
6: **for** $n = 1, \ldots, N$ **do**
7: Let $\beta_{n,t} = 1$ if $q_{n,t}\tilde{E}_{n,t} \leq C_t^{[k^*]}$, and $\beta_{n,t} = 0$ otherwise.
8: **end for**

Given the power scalar σ_t, the device scheduling $\{\beta_{n,t}\}$ in the t-th round aims to solve

$$\mathcal{P}6: \min_{\{\beta_{n,t}\}} \; VU_t + \sum_{n=1}^{N} \beta_{n,t} q_{n,t} \tilde{E}_{n,t} \tag{30a}$$

$$\text{s.t.} \quad \beta_{n,t} \in \{0,1\}, \; \forall n. \tag{30b}$$

An optimal solution to problem $\mathcal{P}6$ is shown in Algorithm 1. In Line 1, we sort $\mathcal{C}_t = \left\{q_{n,t}\tilde{E}_{n,t}, \forall n\right\}$ in the ascending order, and let $C_t^{[m]}$ be the m-th smallest value of $\mathcal{C}_t$. Many sorting algorithms such as Heapsort or Mergesort can be used, with worst-case complexity $O(N \log N)$. In Lines 2-4, we iterate over the possible number of scheduled devices $k \in \{1, \ldots, N\}$, and calculate the corresponding minimum estimated-drift-plus-penalty $v_t(k)$ according to

$$v_t(k) \triangleq \frac{l\eta_t^2}{2}\left(\frac{G^2}{L_b k} + \frac{\sigma_0^2 s}{\sigma_t^2 k^2}\right) + \sum_{n=1}^{k} C_t^{[k]}. \tag{31}$$

The optimal number of devices k^* to be scheduled is obtained by finding the minimum $v_t(k)$ according to Line 5, and k^* devices with smallest estimated drift $q_{n,t}\tilde{E}_{n,t}$ are scheduled, as shown Lines 6-8. Besides Line 1, all the other steps are with complexity $O(N)$. Therefore, the complexity of making a device scheduling decision in a single round is $O(N \log N)$.

3) The Complete Algorithm

The proposed energy-aware dynamic device scheduling algorithm is summarized in Algorithm 2. In the t-th training round, the PS makes device scheduling decision by solving $\mathcal{P}6$ based on the estimated energy consumption and the virtual queue, which is run in an online fashion without any future information. The weight parameter V and the virtual queue states $\{q_{n,t}, \forall n\}$ jointly balance the training gain of the FEEL task and the energy consumption of devices. In particular, a larger V puts more emphasis on scheduling more devices so as to accelerate the convergence rate. Meanwhile, a larger $q_{n,t}$ indicates that the cumulative energy consumption of device n till the current round far exceeds the budget, so that the device tends to save energy. As shown in Algorithm 1, the optimal solution to $\mathcal{P}6$ also indicates that devices with smaller values of $q_{n,t}\tilde{E}_{n,t}$ are always scheduled first, as their energy is relatively sufficient.

Then, the up-to-date global model vector $\boldsymbol{w}_{t-1}$ is broadcast to the scheduled devices, who run local SGD to compute local gradients $\tilde{\boldsymbol{g}}_{n,t}$ in parallel. After computation, local gradients are aggregated over-the-air and the global model is updated by the PS. Finally, the PS collects the actual energy consumption of each scheduled device, which also contains the information of local gradient norm $\left\|\tilde{\boldsymbol{g}}_{n,t}\right\|_2$, and updates the virtual queue states for all the devices to guide the scheduling decision in the next training round. The state information is exchanged with reliable point-to-point communication, and the transmission cost is neglected.

Algorithm 2 Energy-Aware Dynamic Device Scheduling Algorithm

1: **Initialization**: initialize global model $\boldsymbol{w}_0$. Each device n runs local SGD according to (3) to report $\left\|\tilde{\boldsymbol{g}}_{n,0}\right\|_2^2$ to the PS, and let $q_{n,1} = 0$.
2: **for** $t = 1, \ldots, T$ **do**
3: The PS set σ_t according to (29), acquires channel gains $\tilde{h}_{n,t}$ and calculates the estimated energy consumption $\tilde{E}_{n,t}$ according to (26) for all devices.
4: The PS schedules a subset of devices $\mathcal{B}_t$ by solving $\mathcal{P}6$ according to Algorithm 1.
5: The PS broadcasts $\boldsymbol{w}_{t-1}$ and σ_t to the scheduled devices $n \in \mathcal{B}_t$.
6: Each scheduled device $n \in \mathcal{B}_t$ updates local gradient $\tilde{\boldsymbol{g}}_{n,t}$ according to (3), and transmits $\frac{\sigma_t}{\tilde{h}_{n,t}}\tilde{\boldsymbol{g}}_{n,t}$ simultaneously with all the other scheduled devices.
7: The PS receives $\boldsymbol{y}_t$ and updates the global model $\boldsymbol{w}_t$ according to (10).
8: Each scheduled device $n \in \mathcal{B}_t$ reports $E_{n,t}$ to the PS, and the PS updates the virtual queue $q_{n,t}$ for all devices according to (25).
9: **end for**

We remark that, while Algorithm 2 is designed based on the convergence analysis above, the proposed online device scheduling framework has a wide applicability. Once convergence analysis is carried out in any other system setting, such as multiple local iterations or non-convex loss functions, we can then substitute U_t with the derived convergence upper bound while still using the estimated-drift-plus-penalty algorithm for dynamic scheduling.

C. Performance Analysis

The performance of the proposed dynamic device scheduling algorithm is characterized by comparing with its optimal offline counterpart

$$\begin{aligned} \min_{\{\beta_{n,t}\}_{t=1}^T} \quad & \sum_{t=1}^{T} U_t \\ \text{s.t.} \quad & \sum_{t=1}^{T} \beta_{n,t} E_{n,t} \leq \bar{E}_n, \; \forall n, \\ & \beta_{n,t} \in \{0,1\}, \; \forall n, t. \end{aligned}$$

This is in fact the device scheduling problem of $\mathcal{P}4$, regarding $\{\sigma_t\}_{t=1}^T$ as a pre-defined hyper-parameter sequence. Let $\{\beta_{n,t}^*\}_{t=1}^T$ be the offline optimal device schedule obtained by solving the problem above, and $\sum_{t=1}^T U_t^*$ the corresponding

offline optimal loss. Define $\sum_{t=1}^{T} U_t^{\ddagger}$ as the cumulative loss of the proposed algorithm, which is achieved by solving the online device scheduling problem $\mathcal{P}6$ in each round. To enable the theoretical analysis, we neglect the impact of current scheduling decision on the future gradient norm for the offline counterpart. Meanwhile, we assume that the wireless channel state is independent across time, but do not make any assumption on its distribution. The performance guarantee of the proposed algorithm is shown in the following theorem.

Theorem 2. *Compared to the offline optimal solution, the cumulative loss of Algorithm 2 can be bounded by*

$$\sum_{t=1}^{T} U_t^{\ddagger} \leq \sum_{t=1}^{T} U_t^{*} + \frac{\theta_0 T^2 + T(T-1)\delta_0 \sum_{n=1}^{N} \theta_n}{V}, \quad (32)$$

and the total energy consumption of Algorithm 2 can be bounded by

$$\sum_{t=1}^{T} \beta_{n,t} E_{n,t} \leq \bar{E}_n + \sqrt{2V \sum_{t=1}^{T} U_t^{*} + 2\theta_0 T^2 + 2T(T-1)\delta_0 \sum_{n=1}^{N} \theta_n}, \quad (33)$$

where $\delta_0 \triangleq \max_{\{n,t\}} \left\{ \left| \tilde{E}_{n,t} - E_{n,t} \right| \right\}$, $\theta_0 \triangleq \sum_{n=1}^{N} \frac{1}{2}\theta_n^2$ *and* $\theta_n \triangleq \max_t \left\{ \left| E_{n,t} - \frac{\bar{E}_n}{T} \right| \right\}$.

Proof. See Appendix C. □

Theorem 2 shows that, the training performance of the proposed energy-aware dynamic device scheduling algorithm can be bounded with respect to its optimal offline counterpart, while the deviation between the cumulative energy consumption of each device and its budget is also bounded. The worst-case performance can be improved by reducing the upper bound of the energy usage bias θ_n and the maximum energy estimation error δ_0. Moreover, the trade-off between the training performance of the FEEL task and maximum energy consumption of each device can be balanced by the weight parameter V. In practice, we should use the energy in a balanced manner to avoid large θ_n, and carefully select V to optimize the training performance within the energy limits.

We also remark here that, compared to the state-of-the-art that also applies Lyapunov optimization to solve scheduling problem under energy constraints [1], [21], [46], our analysis further shows the impact of estimation error on the performance bound.

D. Implementation Issues

To enable the efficient implementation of the proposed algorithm in s real system, we provide some practical considerations as follows.

1) Communication Rescheduling

The key motivation of rescheduling is to avoid using significantly more energy than expected when the estimation error of $\tilde{E}_{n,t}$ is large. To be specific, after local gradient computation, each scheduled device can learn its exact energy consumption $E_{n,t}$ by calculating the local gradient norm $\left\|\tilde{\boldsymbol{g}}_{n,t}\right\|_2^2$ and acquiring the accurate channel gain $h_{n,t}$. If $E_{n,t} - \tilde{E}_{n,t} \leq \delta_h$, where $\delta_h > 0$ is a given threshold, then the device is scheduled for gradient aggregation. Otherwise, the device backs off from the communication step.

2) Peak Power Constraint

In practice, the power used to transmit each entry of $p_{n,t}\tilde{\boldsymbol{g}}_{n,t}$ cannot exceed the peak power limit of a device. While the proposed algorithm does not schedule a device with high communication energy in general, it cannot guarantee that all transmitted entries satisfy the peak power constraint. Inspired by the idea of gradient clipping [47], we can simply truncate $p_{n,t}\tilde{\boldsymbol{g}}_{n,t}$ at the peak power. This generally does not introduce significant loss unless the peak power is highly limited.

3) Minimum Value of Virtual Queue

The typical evolution of virtual queue is given in (25), in which the minimum queue value is set to 0. In problem $\mathcal{P}6$, $q_{n,t} = 0$ indicates that the energy consumption is not considered in the current scheduling round, and thus the device is scheduled. However, the energy consumption $E_{n,t}$ might be large, leading to a large deviation θ_n and thus a poor worst-case performance. To avoid such cases, we instead set $q_{\min} > 0$ as the minimum value of the virtual queue in practice.

4) Estimations of Smoothness Parameter l and Variance Bound G^2

Our algorithm is designed based on the convergence analysis under Assumptions in Section III. These hyper-parameters should be estimated in practice. According to the definition of smoothness, l is estimated by the maximum value of $\frac{\left\|\tilde{\boldsymbol{g}}_{n,t} - \tilde{\boldsymbol{g}}_{n,t-1}\right\|}{\left\|\boldsymbol{w}_{t-1} - \boldsymbol{w}_{t-2}\right\|}$ during training, while each device can count the variance of local gradients to set a reasonable variance bound G^2.

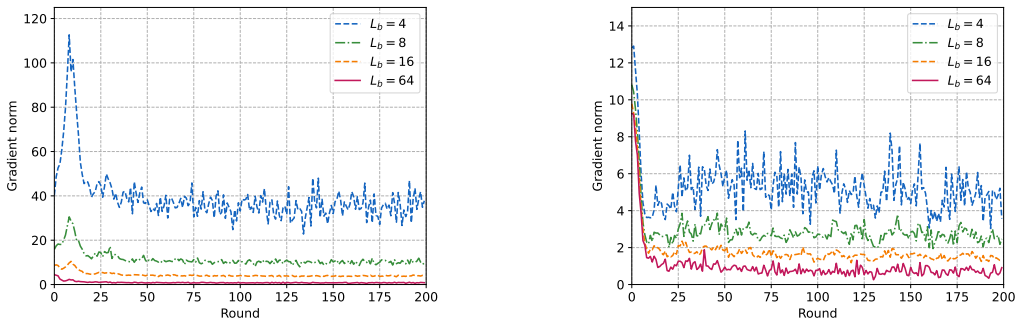
V. Experiments

In this section, we evaluate the proposed energy-aware dynamic device scheduling algorithm for an image classification task using both MNIST[1] and CIFAR-10[2] datasets. We consider $N = 10$ devices and both i.i.d. and non-i.i.d. datasets on devices. For the i.i.d. case, the training dataset of MNIST with 60000 samples (or CIFAR-10 with 50000 samples) is randomly partitioned into N disjoint subsets, and each device holds one subset. For the non-i.i.d. case, we sort the data samples by their labels, and each device holds a disjoint subset of data with m labels (represented by 'non-i.i.d. (m)' in the following). Note that the data distributions are more skewed for smaller m, and they become i.i.d. when m is equal to the total number of classes in the dataset.

For MNIST, we train a multilayer perceptron (MLP) which has a 784-unit input layer with ReLU activation, a 64-unit hidden layer, and a 10-unit softmax output layer, with 50890 parameters in total. The total number of rounds is set to $T = 200$, and 10 local iterations are carried out per round with batch size $L_b = 64$. In each round, the total computation energy is 1J for each device. For CIFAR-10, we train a convolutional neural network (CNN) with the following structure:

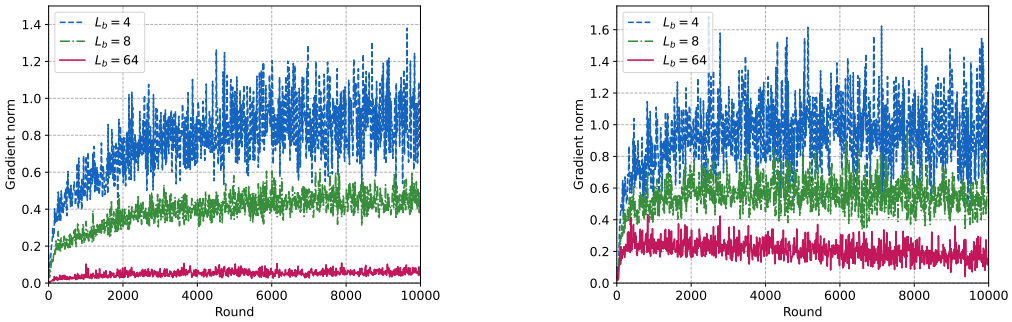
[1]http://yann.lecun.com/exdb/mnist/

[2]https://www.cs.toronto.edu/ kriz/cifar.html



(a) I.i.d. local data. (b) Non-i.i.d. ($m = 1$).

Fig. 2. The l_2-norm of local gradients and their estimated values on the MNIST dataset.



(a) I.i.d. local data. (b) Non-i.i.d. ($m = 2$).

Fig. 3. The l_2-norm of local gradients and their estimated values on the CIFAR-10 dataset.

two 3×3 convolution layers each with 32 channels and followed by a 2×2 max pooling layer, two 3×3 convolution layers each with 64 channels and followed by a 2×2 max pooling layer, a fully connected layer with 120 units, and finally a 10-unit softmax output layer. Each convolution or fully connected layer is activated by ReLU, and the total number of model parameters is 258898. We train the model for $T = 10000$ rounds, and one mini-batch is run per round with batch size $L_b = 64$. Local computation energy per round per device is set to 10J.

For both MNIST and CIFAR-10, the learning rate η_t is set to 0.05, $\forall t$, a momentum of 0.9 is adopted, and cross entropy is adopted as the loss function. The wireless channel follows Rayleigh fading with scale parameter 1, and by default we assume that the accurate channel gain can be observed, i.e., $\tilde{h}_{n,t} = h_{n,t}$. The variance of channel noise is $\sigma_0^2 = 10^{-6}$. The power scalar is selected according to (29), where the default SNR threshold is $\gamma_0 = 7$dB. For the dynamic scheduling algorithm, the minimum value of virtual queue is $q_{\min} = 0.1$, and the maximum estimation error $\delta_h = 0.5\tilde{E}_{n,t}$ is allowed for communication reschedule.

A. l_2-Norm of Local Gradients

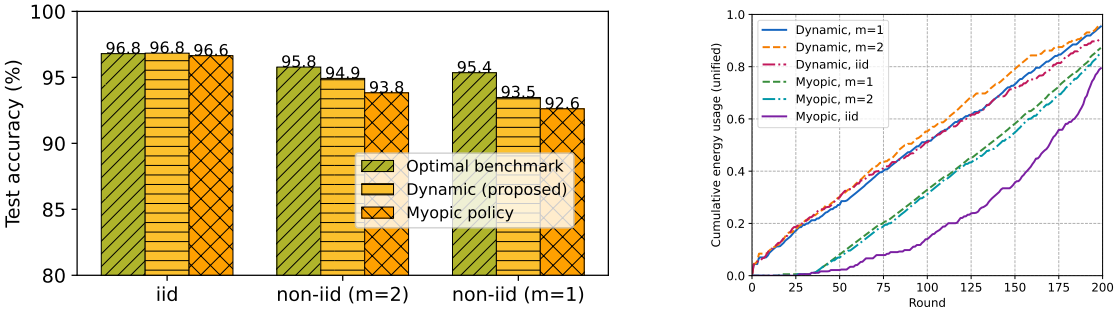
In Fig. 2 and Fig. 3, we first evaluate the EST-C and EST-P methods proposed in Section IV-A that estimate the l_2-norm of local gradient, by observing the temporal variations of the gradients. To eliminate the impact of device scheduling, we do not limit the energy consumption and all devices are scheduled. The batch size L_b used for the model training is 64. In each round, each device further computes its local gradient with smaller batch sizes $L_b = 4$, 8 and 16 and records the corresponding estimated gradient norm, which is adopted by the EST-C method as $\left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2^2$. For the EST-P method, $\left\|\tilde{\boldsymbol{g}}_{n,t}^{[\text{est}]}\right\|_2^2$ will be given the value of the l_2-norm of gradients with $L_b = 64$ at a certain round before t. Each curve is averaged over 50 and 20 runs for MNIST and CIFAR-10, respectively.

As shown in Fig. 2 and Fig. 3, the gradient norms achieved by different batch sizes are highly varying, and a smaller batch size yields a higher l_2-norm of gradient due to the non-negligible gradient variance, which is consistent with the analysis in (23). This result indicates that the EST-C method cannot provide an accurate estimation of gradient norm. Meanwhile, with a fixed batch size, such as $L_b = 64$, the gradient norm has a strong temporal correlation. Therefore, the EST-P method can provide a much better estimate of the gradient norm, which is embedded in the proposed dynamic device scheduling algorithm.

B. Performance of the Proposed Device Scheduling Algorithm

We compare the performance of the proposed scheduling algorithm with two benchmarks:

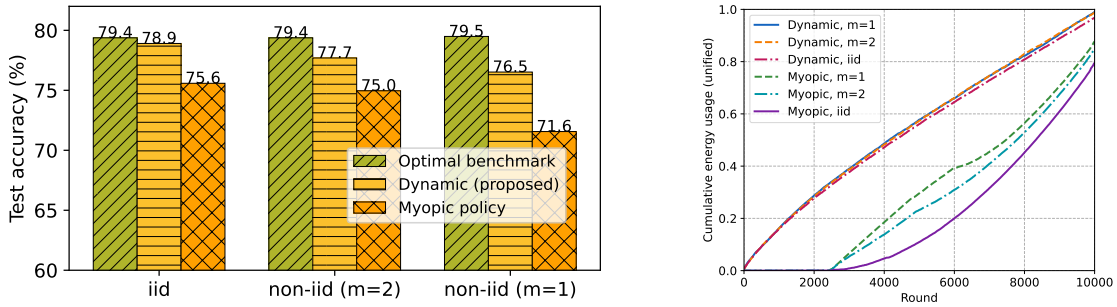
1) Optimal benchmark: Devices do not have energy limitations, so that all of them participate in each training round.



(a) Accuracy on test dataset.

(b) Unified cumulative energy usage.

Fig. 4. Performance of the proposed dynamic scheduling algorithm and benchmarks on MNIST.



(a) Accuracy on test dataset.

(b) Unified cumulative energy usage.

Fig. 5. Performance of the proposed dynamic scheduling algorithm and benchmarks on CIFAR-10.

Channel fading and noise still exist, and thus this benchmark provides the optimal performance under the same wireless settings of the FEEL system.

2) Myopic policy: For each device n, the maximum energy that can be used in round t is given by the remaining energy divided by the remaining number of rounds, i.e., $\frac{\bar{E}_n - \sum_{\tau=1}^{t-1} \beta_{n,t} E_{n,t}}{T-t+1}$.

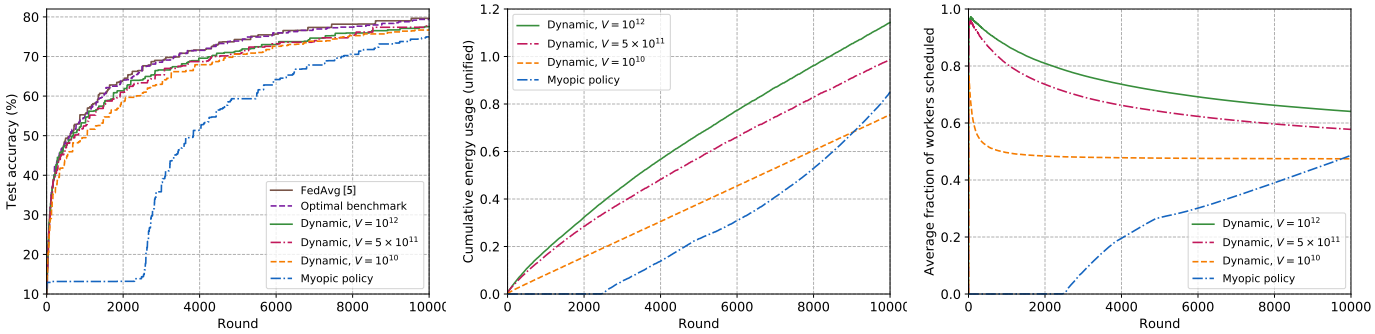
In Fig. 4, we compare the training performance and energy consumption of the proposed dynamic scheduling algorithm with the optimal and myopic benchmarks on MNIST. Let $\bar{E} = 1$J be the energy budget per round, and the total energy budget of each device is $\bar{E}_n = T\bar{E}, \forall n$. For non-i.i.d. data with 1 label per device, the weight parameter is $V = 5 \times 10^7$, while for the other two cases, $V = 10^8$. The training performance is characterized by the accuracy of the MLP model on the test dataset, as shown in Fig. 4(a). Results show that our proposed dynamic scheduling algorithm achieves the optimal accuracy under i.i.d. data, and always outperforms the myopic policy.

To show the energy usage during the training process, we plot $\max_{n\in\mathcal{N}} \frac{\sum_{\tau=1}^{t} \beta_{n,\tau} E_{n,\tau}}{\bar{E}_n}$ which represents the maximum value of the unified cumulative energy usage across devices till the t-th round. As devices may have different energy budgets, directly plotting their maximum energy usage is not informative. Instead, this unified metric reflects how much energy has been used by the devices. For the myopic policy, the energy required for computation and communication exceeds the budget at the beginning of training, thus no devices can be scheduled. However, our proposed algorithm enables devices to use energy in a more flexible way, thus improving the training performance.

Similar comparison is made on CIFAR-10 dataset in Fig. 5, where $\bar{E} = 8$J and $V = 5 \times 10^{11}$. Note that compared to the local computation energy required per round (10J), the energy budget is relatively limited, and the advantage of the proposed dynamic scheduling over the myopic policy is more prominent in such a scenario. In particular, under the highly non-i.i.d. case with $m = 1$, dynamic scheduling improves the accuracy by 4.9% compared to the myopic policy, by utilizing 10% more energy in a more balanced manner. We can also see that our proposed algorithm can satisfy the energy constraints of devices under both datasets (at the end of training, the unified energy usage is smaller than 1).

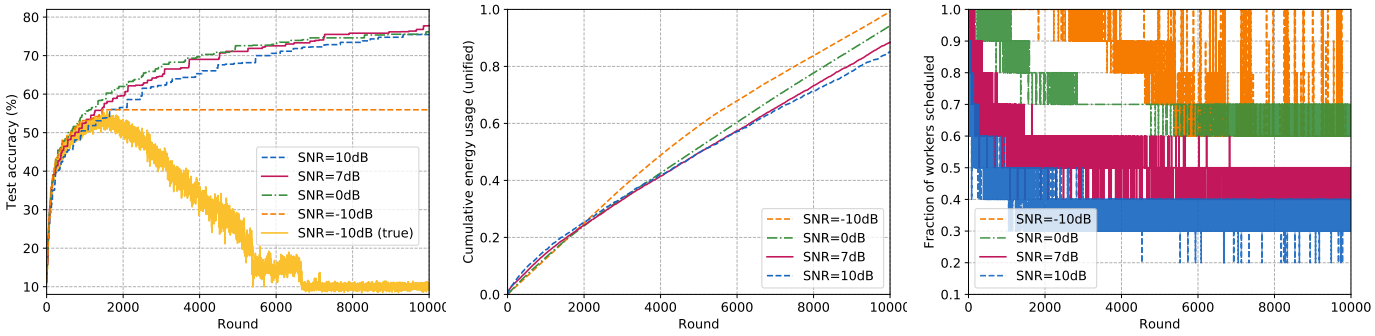
In the following, we further explore the impact of key parameters on the training performance and energy consumption with CIFAR-10, as it is more challenging than MNIST. We focus on the non-i.i.d. case, where each device has a local subset with $m = 2$ labels.

Fig. 6 validates that the weight parameter V can balance the trade-off between the training performance and energy consumption, where $\bar{E} = 8$J. As V increases, devices use energy in a more aggressive manner, leading to a higher energy usage and more scheduled devices, so as to accelerate the convergence. However, if V is too large, such as $V = 10^{12}$, energy is not given enough attention and finally the limit is violated. In practical systems, V should be judiciously selected to optimize the training performance while satisfying the energy constraints. We further plot the accuracy of the



(a) Accuracy on test dataset. (b) Unified cumulative energy usage. (c) Scheduled devices.

Fig. 6. Performance of the proposed algorithm under different weight parameter V on CIFAR-10.



(a) Accuracy on test dataset. (b) Unified cumulative energy usage. (c) Scheduled devices.

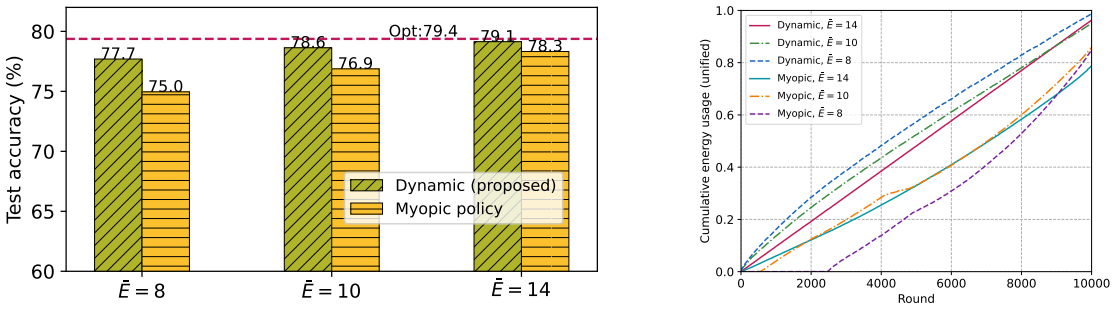
Fig. 7. Performance of the proposed algorithm under different received SNR thresholds on CIFAR-10.

classical federated averaging (FedAvg) algorithm in [5] without considering channel fading and noise, while the training parameters remain the same, and the local data is i.i.d. All the devices are scheduled in each round, and their models can be uploaded precisely for global averaging. FedAvg in this ideal setting provides a baseline accuracy for the considered FL system, regardless of the wireless channel environments. Compared to FedAvg, we can see that our proposed algorithm with a proper V can achieve a similar convergence rate, while the gap of the accuracy is mainly due to the energy constraints and channel noise.

The impact of the received SNR threshold γ_0 on the training performance and energy consumption with the proposed dynamic scheduling algorithm is shown in Fig. 7, where $\bar{E} = 8$J, and $V = 2.5 \times 10^{11}$. In Fig. 7(a), the curve marked with 'true' plots the actual evolution of the model accuracy during the training process with SNR threshold -10dB, while the other curves present the best test accuracy up-to-date. The maximum cumulative energy usage and instantaneous fraction of devices that are scheduled in each round is shown in Fig. 7(b) and Fig. 7(c), respectively. Clearly, a smaller SNR threshold helps to save communication energy, and thus more devices can be scheduled in each round. However, the cumulative noise might degrade the accuracy or even diverge the training if the SNR is too low, for instance when SNR $= -10$dB. On the other hand, a larger SNR, such as SNR $= 10$dB, makes communication more energy-consuming, which also degrades the training performance due to fewer participants. A proper value of the received SNR threshold should be given to balance the negative impact of noise and the energy consumption. As shown in Fig. 7, SNR $= 7$dB is the best choice under our simulation setting.

We compare our proposed algorithm with optimal and myopic benchmarks under different energy budgets in Fig. 8. For $\bar{E} = 14$J, we set $V = 10^{10}$, and for $\bar{E} = 8$J or 10J, we let $V = 5 \times 10^{11}$. Our proposed dynamic scheduling algorithm always outperforms the myopic benchmark by achieving higher accuracy and utilizing energy more efficiently, and approaches the optimal accuracy as $\bar{E}$ increases. Moreover, the accuracy gap between the proposed algorithm and myopic policy is 2.7%, 1.7% and 0.8% for $\bar{E} = 8$, 10 and 14, respectively, indicating that the dynamic scheduling algorithm is particularly promising under the energy-limited regime.

We also evaluate the robustness of the proposed dynamic scheduling algorithm by introducing channel observation errors, where $\bar{E} = 8$J and $V = 5 \times 10^{11}$. In Fig. 9, the first group of bars are obtained without channel observation error, i.e., $\tilde{h}_{n,t} = h_{n,t}$. The second to the forth group of bars suffer inaccurate channel observations. For example, if the error is 20%, then $\tilde{h}_{n,t}$ is uniformly distributed within $[0.8h_{n,t}, 1.2h_{n,t}]$. A larger observation error further leads to less accurate energy estimations $\tilde{E}_{n,t}$. However, our main finding is that the proposed algorithm only suffers a tiny accuracy degradation, which validates its robust training performance in the practical scenarios. We also mention that the myopic policy also performs well under different observation errors



(a) Accuracy on test dataset.

(b) Unified cumulative energy usage.

Fig. 8. Performance of the proposed algorithm and benchmarks under different energy budgets on CIFAR-10.

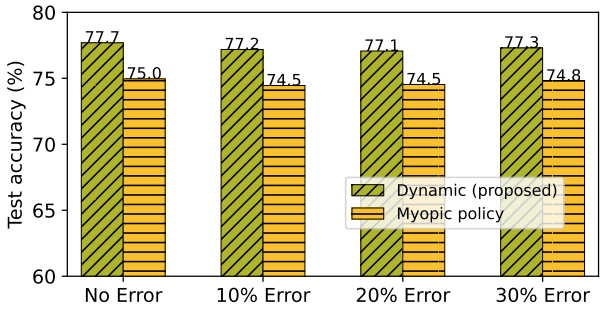


Fig. 9. Performance of the proposed algorithm and myopic benchmark under different channel estimation errors on CIFAR-10.

compared to the error-free case. Nevertheless, the proposed algorithm still beats the myopic policy by a significant margin in all the scenarios.

Finally, we consider a heterogeneous scenario where a total number of $N = 40$ devices have different energy constraints, and use different batch sizes for training. In specific, there are 20 devices with batch size $L_b = 64$ and per round energy budget $\bar{E} = 8$J, 10 devices with $L_b = 128$ and $\bar{E} = 16$J, and the other 10 devices with $L_b = 256$ and $\bar{E} = 32$J. The weight parameter V is set to 10^{13} for the i.i.d. case, and 9×10^{12} for the non-i.i.d. cases. Note that, to implement the proposed algorithm with different batch sizes, we need to set the smallest batch size to L_b in problem $\mathcal{P}6$, so that the convergence upper bound still holds. As shown in Fig. 10, the proposed algorithm still outperforms the myopic benchmark, with over 3% accuracy improvement under the highly non-i.i.d. case. At the same time, the energy constraints of devices are satisfied.

VI. Conclusions

We have investigated the device scheduling problem for FEEL with over-the-air gradient aggregation, aiming to optimize the training performance under joint communication and computation energy limits of devices. Convergence analysis has been carried out showing the importance of device participation to the training performance, and an energy-aware dynamic device scheduling algorithm has been developed. In particular, we have noticed the existence of unobservable states, mainly the l_2-norm of local gradients, for online decision making in over-the-air FEEL, and proposed an estimated-drift-plus-penalty solution based on the Lyapunov optimization framework accordingly. We have characterized a theoretical guarantee for the proposed dynamic scheduling algorithm by taking the deviation of estimated states into consideration. Experiments on MNIST and CIFAR-10 datasets have been carried out to validate the theoretical findings. Compared to the myopic benchmark, we have shown a significant 4.9% accuracy improvement on CIFAR-10 for a highly non-i.i.d. data distribution and stringent energy constraints.

As future directions, heterogeneous data distributions across devices can be considered, where local datasets represent different number of classes. We would like to observe if data diversity of a device should be taken into account for the scheduling decision. The trade-off between training delay and energy consumption in over-the-air FEEL is worth further investigation, as well as the impact of the non-convexity of loss functions on the convergence rate and the device schedule.

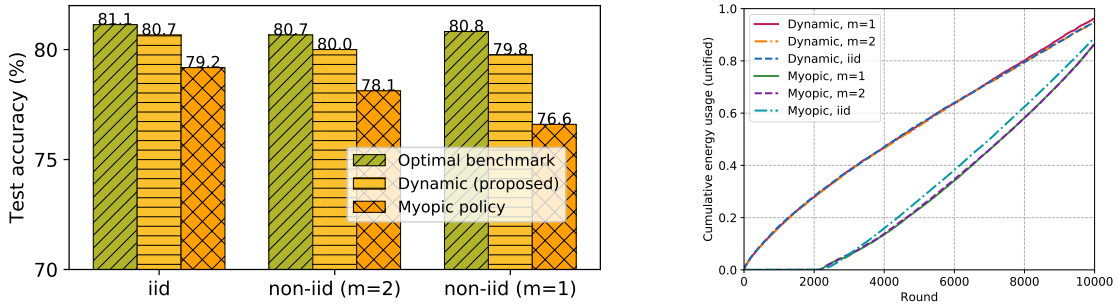
Appendix A
Proof of Lemma 1

For the simplicity of notation, let $\tilde{\boldsymbol{g}}_t \triangleq \frac{\sum_{n\in\mathcal{B}_t} \tilde{\boldsymbol{g}}_{n,t}}{|\mathcal{B}_t|}$ be the average of the local gradients of scheduled devices, and $\tilde{\boldsymbol{z}}_t \triangleq \frac{\boldsymbol{z}_t}{\sigma_t |\mathcal{B}_t|}$ the noise received at the PS. The global model is updated according to

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \eta_t(\tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{z}}_t). \tag{34}$$

According to Assumption 2, the gap of loss between two adjacent rounds can be bounded by

$$F(\boldsymbol{w}_t) - F(\boldsymbol{w}_{t-1})$$



(a) Accuracy on test dataset.

(b) Unified cumulative energy usage.

Fig. 10. Performance of the proposed dynamic scheduling algorithm and benchmarks on CIFAR-10 under different batch sizes and heterogeneous energy constraints.

$$\leq \nabla F(\boldsymbol{w}_{t-1})^{\mathrm{T}}(\boldsymbol{w}_t - \boldsymbol{w}_{t-1}) + \frac{l}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_2^2$$
$$= -\eta_t \boldsymbol{g}_t^{\mathrm{T}}(\tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{z}}_t) + \frac{l\eta_t^2}{2}\|\tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{z}}_t\|_2^2, \tag{35}$$

where $\boldsymbol{g}_t = \nabla F(\boldsymbol{w}_{t-1})$ is the global full gradient in round t.

Based on the facts that the channel noise and local gradients are independent, and each entry in the noise vector $\boldsymbol{z}_t$ follows Gaussian distribution with zero mean and variance σ_0^2, we first take the expectation over the received noise vector $\tilde{\boldsymbol{z}}_t$, and get

$$\mathbb{E}_{\boldsymbol{z}_t}\left[\|\tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{z}}_t\|_2^2\right] = \|\tilde{\boldsymbol{g}}_t\|_2^2 + \mathbb{E}_{\boldsymbol{z}_t}\left[\|\tilde{\boldsymbol{z}}_t\|_2^2\right]$$
$$= \|\tilde{\boldsymbol{g}}_t\|_2^2 + \frac{\sigma_0^2 s}{\sigma_t^2|\mathcal{B}_t|^2}, \tag{36}$$

Substituting (36) into (35), we obtain

$$\mathbb{E}_{\boldsymbol{z}_t}\left[F(\boldsymbol{w}_t) - F(\boldsymbol{w}_{t-1})\right]$$
$$\leq -\eta_t \boldsymbol{g}_t^{\mathrm{T}}\tilde{\boldsymbol{g}}_t + \frac{l\eta_t^2}{2}\|\tilde{\boldsymbol{g}}_t\|_2^2 + \frac{l\eta_t^2}{2}\frac{\sigma_0^2 s}{\sigma_t^2|\mathcal{B}_t|^2}. \tag{37}$$

Then we consider the relation between $\tilde{\boldsymbol{g}}_t$ and $\boldsymbol{g}_t$ by taking the expectation over stochastic data sampling. Based on Assumption 1, we get

$$\mathbb{E}_{\mathcal{L}_{n,t}}\left[\tilde{\boldsymbol{g}}_t\right] = \mathbb{E}_{\mathcal{L}_{n,t}}\left[\frac{\sum_{n\in\mathcal{B}_t}\tilde{\boldsymbol{g}}_{n,t}}{|\mathcal{B}_t|}\right] = \boldsymbol{g}_t, \tag{38}$$

$$\mathbb{E}_{\mathcal{L}_{n,t}}\left[\|\tilde{\boldsymbol{g}}_t\|_2^2\right] = \mathbb{E}\left[\left\|\frac{\sum_{n\in\mathcal{B}_t}\sum_{\boldsymbol{x}\in\mathcal{L}_{n,t}}\nabla f\left(\boldsymbol{w}_{t-1},\boldsymbol{x}\right)}{L_b|\mathcal{B}_t|}\right\|_2^2\right]$$
$$\leq \|\boldsymbol{g}_t\|_2^2 + \frac{G^2}{L_b|\mathcal{B}_t|}. \tag{39}$$

Finally, taking the expectation over noise and SGD on the left hand side of (37), and substituting its right hand side with (38) and (39), we have

$$\mathbb{E}[F(\boldsymbol{w}_t) - F(\boldsymbol{w}_{t-1})]$$
$$\leq -\eta_t\|\boldsymbol{g}_t\|_2^2 + \frac{l\eta_t^2}{2}\left(\|\boldsymbol{g}_t\|_2^2 + \frac{G^2}{L_b|\mathcal{B}_t|}\right) + \frac{l\eta_t^2}{2}\frac{\sigma_0^2 s}{\sigma_t^2|\mathcal{B}_t|^2}$$
$$= -\eta_t\left(1 - \frac{l\eta_t}{2}\right)\|\boldsymbol{g}_t\|_2^2 + \frac{l\eta_t^2}{2}\left(\frac{G^2}{L_b|\mathcal{B}_t|} + \frac{\sigma_0^2 s}{\sigma_t^2|\mathcal{B}_t|^2}\right).$$

Since $\mathbb{E}[F(\boldsymbol{w}_t) - F(\boldsymbol{w}_{t-1})] = \mathbb{E}[F(\boldsymbol{w}_t)] - \mathbb{E}[F(\boldsymbol{w}_{t-1})]$, Lemma 1 is proved.

Appendix B
Proof of Theorem 1

By the μ-strong convexity of the loss functions (Assumption 3), the Polyak-Lojasiewicz inequality holds

$$\|\boldsymbol{g}_t\|_2^2 \geq 2\mu(F(\boldsymbol{w}_{t-1}) - F^*). \tag{40}$$

Substituting (40) into Lemma 1, and assuming that $\eta_t \leq \frac{1}{l}$ (thus $1 - \frac{l\eta_t}{2} \geq \frac{1}{2}$), we can obtain

$$\mathbb{E}[F(\boldsymbol{w}_t) - F(\boldsymbol{w}_{t-1})]$$
$$\leq -\eta_t\left(1 - \frac{l\eta_t}{2}\right)\|\boldsymbol{g}_t\|_2^2 + \frac{l\eta_t^2}{2}\left(\frac{G^2}{L_b|\mathcal{B}_t|} + \frac{\sigma_0^2 s}{\sigma_t^2|\mathcal{B}_t|^2}\right)$$
$$\leq -\eta_t\mu(\mathbb{E}[F(\boldsymbol{w}_{t-1})] - F^*) + \frac{\eta_t}{2}\left(\frac{G^2}{L_b|\mathcal{B}_t|} + \frac{\sigma_0^2 s}{\sigma_t^2|\mathcal{B}_t|^2}\right). \tag{41}$$

Let $A_t \triangleq \frac{\eta_t}{2}\left(\frac{G^2}{L_b|\mathcal{B}_t|} + \frac{\sigma_0^2 s}{\sigma_t^2|\mathcal{B}_t|^2}\right)$, and thus (41) can be re-written as

$$\mathbb{E}[F(\boldsymbol{w}_t)] - F^* \leq (1 - \mu\eta_t)(\mathbb{E}[F(\boldsymbol{w}_{t-1})] - F^*) + A_t. \tag{42}$$

With recursion, we can prove Theorem 1:

$$\mathbb{E}[F(\boldsymbol{w}_t)] - F^* \leq (1 - \mu\eta_t)(\mathbb{E}[F(\boldsymbol{w}_{t-1})] - F^*) + A_t$$
$$\leq (1 - \mu\eta_t)(1 - \mu\eta_{t-1})(\mathbb{E}[F(\boldsymbol{w}_{t-2})] - F^*)$$
$$+ (1 - \mu\eta_t)A_{t-1} + A_t$$
$$\leq \cdots \leq (\mathbb{E}[F(\boldsymbol{w}_0)] - F^*)\prod_{i=1}^{t}(1 - \mu\eta_i)$$
$$+ \sum_{i=1}^{t-1} A_i \prod_{j=i+1}^{t}(1 - \mu\eta_i) + A_t. \tag{43}$$

Appendix C
Proof of Theorem 2

Let $y_{n,t} \triangleq \beta_{n,t}E_{n,t} - \frac{\bar{E}_n}{T}$, and $\tilde{y}_{n,t} \triangleq \beta_{n,t}\tilde{E}_{n,t} - \frac{\bar{E}_n}{T}$. Define the error of estimated energy consumption at device n in the t-th round as $\delta_{n,t} \triangleq \beta_{n,t}\tilde{E}_{n,t} - \beta_{n,t}E_{n,t} = \tilde{y}_{n,t} - y_{n,t}$, with maximum absolute value $\delta_0 \triangleq \max_{\{n,t\}}\left\{\left|\tilde{E}_{n,t} - E_{n,t}\right|\right\}$. According to the evolution of the virtual queue, which is defined in (25), it is easy to prove that $q_{n,t+1}^2 \leq (q_{n,t} + y_{n,t})^2$ and $y_{n,t} \leq q_{n,t+1} - q_{n,t}$.

Define the Lyapunov function as $L(t) \triangleq \sum_{n=1}^{N} \frac{1}{2} q_{n,t}^2$, and the Lyapunov drift of a single round as $\Delta_1(t) \triangleq L(t+1) - L(t)$, which is given by

$$\Delta_1(t) = L(t+1) - L(t) = \sum_{n=1}^{N} \left(\frac{1}{2} q_{n,t+1}^2 - \frac{1}{2} q_{n,t}^2 \right) \leq \sum_{n=1}^{N} \left(\frac{1}{2} y_{n,t}^2 + q_{n,t} y_{n,t} \right) \leq \theta_0 + \sum_{n=1}^{N} q_{n,t} y_{n,t}, \quad (44)$$

where $\theta_0 \triangleq \sum_{n=1}^{N} \frac{1}{2} \theta_n^2$ and $\theta_n \triangleq \max_t \{|y_{n,t}|\}$. By adding VU_t on both sides of (44), an upper bound on the single-round drift-plus-penalty function is given by

$$\begin{aligned} \Delta_1(t) + VU_t &\leq \theta_0 + \sum_{n=1}^{N} q_{n,t} y_{n,t} + VU_t \\ &= \theta_0 + \sum_{n=1}^{N} q_{n,t} \left(\beta_{n,t} E_{n,t} - \frac{\bar{E}_n}{T} \right) + VU_t \quad (45) \\ &= \theta_0 + \sum_{n=1}^{N} q_{n,t} \left(\tilde{y}_{n,t} - \delta_{n,t} \right) + VU_t \\ &= \theta_0 + \sum_{n=1}^{N} q_{n,t} \left(\beta_{n,t} \tilde{E}_{n,t} - \delta_{n,t} - \frac{\bar{E}_n}{T} \right) + VU_t. \quad (46) \end{aligned}$$

The classical drift-plus-penalty algorithm of Lyapunov optimization aims to minimize the upper bound of $\Delta_1(t) + VU_t$, as shown in (45). Since we do not have the exact value of $E_{n,t}$, we instead minimize the estimated-drift-plus-penalty, as shown in (46).

Define the T-round drift as $\Delta_T \triangleq L(T+1) - L(1) = \sum_{n=1}^{N} \frac{1}{2} q_{n,T+1}^2$. Then the T-round drift-plus-penalty function can be bounded by:

$$\begin{aligned} &\Delta_T + V \sum_{t=1}^{T} U_t \\ &\leq \sum_{t=1}^{T} \left(\theta_0 + \sum_{n=1}^{N} q_{n,t} (\tilde{y}_{n,t} - \delta_{n,t}) \right) + V \sum_{t=1}^{T} U_t \\ &= \theta_0 T + \sum_{t=1}^{T} \left(\sum_{n=1}^{N} q_{n,t} \tilde{y}_{n,t} + VU_t - \sum_{n=1}^{N} q_{n,t} \delta_{n,t} \right) \quad (47) \end{aligned}$$

We use superscript * to represent the optimal offline solution of $\mathcal{P}4$ (σ_t is not an optimization variable), superscript † to represent the classical drift-plus-penalty algorithm, i.e., $\min_{\{\beta_{n,t}\}} VU_t + \sum_{n=1}^{N} \beta_{n,t} q_{n,t} E_{n,t}$, and ‡ to represent our proposed estimated-drift-plus-penalty algorithm that solves $\mathcal{P}6$.

The T-round drift-plus-penalty is bounded by:

$$\begin{aligned} &\Delta_T^\ddagger + V \sum_{t=1}^{T} U_t^\ddagger \\ &\leq \theta_0 T + \sum_{t=1}^{T} \left(\sum_{n=1}^{N} q_{n,t} \tilde{y}_{n,t}^\ddagger + VU_t^\ddagger - \sum_{n=1}^{N} q_{n,t} \delta_{n,t}^\ddagger \right) \\ &\overset{(a)}{\leq} \theta_0 T + \sum_{t=1}^{T} \left(\sum_{n=1}^{N} q_{n,t} \tilde{y}_n^\dagger(t) + VU_t^\dagger - \sum_{n=1}^{N} q_{n,t} \delta_{n,t}^\ddagger \right) \\ &= \theta_0 T + \sum_{t=1}^{T} \left(\sum_{n=1}^{N} q_{n,t} \left(y_{n,t}^\dagger + \delta_{n,t}^\dagger \right) + VU_t^\dagger - \sum_{n=1}^{N} q_{n,t} \delta_{n,t}^\ddagger \right) \\ &= \theta_0 T + \sum_{t=1}^{T} \left(\sum_{n=1}^{N} q_{n,t} y_{n,t}^\dagger + VU_t^\dagger + \sum_{n=1}^{N} q_{n,t} \left(\delta_{n,t}^\dagger - \delta_{n,t}^\ddagger \right) \right) \\ &\overset{(b)}{\leq} \theta_0 T + \sum_{t=1}^{T} \left(\sum_{n=1}^{N} q_{n,t} y_{n,t}^* + VU_t^* + 2\delta_0 \sum_{n=1}^{N} q_{n,t} \right). \quad (48) \end{aligned}$$

Inequality (a) holds because optimally solving $\mathcal{P}6$ yields a minimum value $\sum_{n=1}^{N} q_{n,t} \tilde{y}_{n,t}^\ddagger + VU_t^\ddagger$ for each t. Inequality (b) holds since the drift-plus-penalty algorithm achieves the minimum value of $\sum_{n=1}^{N} q_{n,t} y_{n,t} + VU_t$, and thus plugging in the optimal offline policy on the right-hand-side increases the value.

Now we bound the right-hand-side of (48). Note that $q_{n,t+1} - q_{n,t} \leq \theta_n, \forall t, n$, and thus

$$q_{n,t} = q_{n,t} - q_{n,1} = \sum_{\tau=1}^{t-1} (q_{n,\tau+1} - q_{n,\tau}) \leq (t-1)\theta_n, \quad (49)$$

$$q_{n,t} y_{n,t}^* = (q_{n,t} - q_{n,1}) y_{n,t}^* \leq (t-1)\theta_n^2. \quad (50)$$

Substituting (49) and (50) into (48) yields

$$\begin{aligned} &\Delta_T^\ddagger + V \sum_{t=1}^{T} U_t^\ddagger \\ &\leq \theta_0 T + V \sum_{t=1}^{T} U_t^* + \sum_{t=1}^{T} \sum_{n=1}^{N} (t-1)\theta_n^2 + 2\delta_0 \sum_{t=1}^{T} \sum_{n=1}^{N} (t-1)\theta_n \\ &= \theta_0 T + V \sum_{t=1}^{T} U_t^* + \theta_0 T(T-1) + T(T-1)\delta_0 \sum_{n=1}^{N} \theta_n \\ &= V \sum_{t=1}^{T} U_t^* + \theta_0 T^2 + T(T-1)\delta_0 \sum_{n=1}^{N} \theta_n. \quad (51) \end{aligned}$$

Notice that $\Delta_T^\ddagger \geq 0$, (32) in Theorem 2 can be derived from (51) by dividing both sides by V. As $U_t > 0$, and for $\forall n$, $\frac{1}{2} q_{n,T+1}^2 \leq \Delta_T$, we get

$$\begin{aligned} &\sum_{t=1}^{T} y_{n,t} = \sum_{t=1}^{T} \beta_{n,t} E_{n,t} - \bar{E}_n \leq \sum_{t=1}^{T} q_{n,t+1} - q_{n,t} = q_{n,T+1} \\ &\leq \sqrt{2\Delta_T} = \sqrt{2V \sum_{t=1}^{T} U_t^* + 2\theta_0 T^2 + 2T(T-1)\delta_0 \sum_{n=1}^{N} \theta_n}. \end{aligned}$$

Thus eq. (33) in Theorem 2 is proved.

REFERENCES

[1] Y. Sun, S. Zhou, and D. Gunduz, "Energy-aware analog aggregation for federated learning with redundant data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020.

[2] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. -C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," in *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98-105, Apr. 2017.

[3] D. Gunduz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy and M. van der Schaar, "Machine learning in the air," in *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2184-2199, Oct. 2019.

[4] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *NIPS Workshop on Private Multi-Party Machine Learning*, Oct. 2016.

[5] B. McMahan, E. Moore, D. Ramage, et al. "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artificial Intelligence and Statistics (AIStats)*, Apr. 2017.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," in *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

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

[8] W. Y. B. Lim, et al., "Federated learning in mobile edge networks: A comprehensive survey," in *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031-2063, thirdquarter 2020.

[9] M. Chen, D. Gunduz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *arXiv preprint arXiv:2104.02151*, Apr. 2021.

[10] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Dec. 2017.

[11] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Advances in Neural Information Processing Systems (NIPS)*, Dec. 2018.

[12] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," in *IEEE Trans. Signal Process.*, vol. 68, pp. 2128-2142, Mar. 2020.

[13] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics (AIStats)*, Jun. 2020.

[14] E. Ozfatura, K. Ozfatura, D. Gunduz, "Time-correlated sparsification for communication-efficient federated learning," *arXiv preprint arXiv:2101.08837*, Jan. 2021.

[15] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," in *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317-333, Jan. 2020.

[16] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning", in *Proc. 32nd Conference on Neural Information Processing Systems (NeurIPS)*, Montreal, Canada, Dec. 2018.

[17] Z. Qu, K. Lin, J. Kalagnanam, Z. Li, J. Zhou, Z. Zhou, "Federated learning's blessing: Fedavg has linear speedup," in *Proc. ICLR Workshop on Distributed and Private Machine Learning (DPML)*, May 2021.

[18] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," in *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205-1221, Jun. 2019.

[19] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto and R. Yonetani, "Hybrid-FL for wireless networks: Cooperative learning mechanism using non-iid data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020.

[20] Q. Zeng, Y. Du, K. K. Leung, and K. Huang, "Energy-efficient radio resource allocation for federated edge learning," in *Proc. IEEE Int. Conf. Commun. Workshops*, Dublin, Ireland, Jun. 2020.

[21] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," in *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188-1200, Feb. 2021.

[22] M. Mohammadi Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," in *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643-3658, Jun. 2021.

[23] Y. Sun, W. Shi, X. Huang, S. Zhou and Z. Niu, "Edge learning with timeliness constraints: Challenges and solutions," in *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 27-33, Dec. 2020.

[24] M. Chen, H. V. Poor, W. Saad and S. Cui, "Convergence time optimization for federated learning over wireless networks," in *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457-2471, Apr. 2021.

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

[26] J. Ren, Y. He, D. Wen, G. Yu, K. Huang and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," in *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690-7703, Nov. 2020.

[27] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, Barcelona, Spain, May 2020.

[28] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM*, Paris, France, May 2019.

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

[30] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," in *Journal of Communications and Information Networks*, vol. 6, no. 2, pp. 110-124, Jul. 2021.

[31] D. Gunduz, D. B. Kurka, M. Jankowski, M. Mohammadi Amiri, E. Ozfatura and S. Sreekumar, "Communicate to learn at the edge," in *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 14-19, Dec. 2020.

[32] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," in *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19-25, Jan. 2020.

[33] G. Zhu, Y. Wang, and K. Huang, "Low-latency broadband analog aggregation for federated edge learning," in *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491-506, Jan. 2020.

[34] M. Mohammadi Amiri and D. Gunduz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," in *IEEE Trans. Signal Process.*, vol. 68, pp. 2155-2169, Apr. 2020.

[35] M. Mohammodi Amiri and D. Gunduz, "Federated learning over wireless fading channels," in *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546-3557, May 2020.

[36] G. Zhu, J. Xu and K. Huang, "Over-the-air computing for 6G: Turning air into a computer," *arXiv preprint arXiv:2009.02181*, Sept. 2020.

[37] H. Guo, A. Liu and V. K. N. Lau, "Analog gradient aggregation for federated learning over wireless networks: Customized design and convergence analysis," in *IEEE Internet Things J.*, vol. 8, no. 1, pp. 197-210, Jan. 2021.

[38] X. Wei and C. Shen, "Federated learning over noisy channels: Convergence analysis and design examples," *arXiv preprint arXiv:2101.02198*, Jan. 2021.

[39] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Convergence of federated learning over a noisy downlink," in *IEEE Trans. Wireless Commun.*, early access, Aug. 2021.

[40] N. Zhang and M. Tao, "Gradient statistics aware power control for over-the-air federated learning in fading channels," in *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5115-5128, Aug. 2021.

[41] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, "Over-the-air federated learning from heterogeneous data," in *IEEE Trans. Signal Process.*, vol. 69, pp. 3796-3811, 2021.

[42] Y. Shao, D. Gunduz, S. C. Liew, "Federated edge learning with misaligned over-the-air computation," *arXiv preprint arXiv:2102.13604*, Feb. 2021.

[43] M. Mohammodi Amiri, T. M. Duman, D. Gunduz, S. R. Kulkarni, H. V. Poor, "Blind federated edge learning," in *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5129-5143, Aug. 2021.

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

[45] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.

[46] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," in *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637-2646, Nov. 2017.

[47] Moritz Hardt, Ben Recht, Yoram Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *Proc. 33rd International Conference on Machine Learning (ICML)*, Jun. 2016.