Broadband Analog Aggregation for Low-Latency Federated Edge Learning

Guangxu Zhu, *Student Member, IEEE*, Yong Wang, and Kaibin Huang, *Senior Member, IEEE*

***Abstract*— To leverage rich data distributed at the network edge, a new machine-learning paradigm, called edge learning, has emerged where learning algorithms are deployed at the edge for providing intelligent services to mobile users. While computing speeds are advancing rapidly, the communication latency is becoming the bottleneck of fast edge learning. To address this issue, this work is focused on designing a low-latency multi-access scheme for edge learning. To this end, we consider a popular privacy-preserving framework, *federated edge learning* (FEEL), where a global AI-model at an edge-server is updated by aggregating (averaging) local models trained at edge devices. It is proposed that the updates simultaneously transmitted by devices over broadband channels should be analog aggregated "over-the-air" by exploiting the waveform-superposition property of a multi-access channel. Such *broadband analog aggregation* (BAA) results in dramatical communication-latency reduction compared with the conventional orthogonal access (i.e., OFDMA). In this work, the effects of BAA on learning performance are quantified targeting a single-cell random network. First, we derive two tradeoffs between communication-and-learning metrics, which are useful for network planning and optimization. The power control ("truncated channel inversion") required for BAA results in a tradeoff between the update-reliability [as measured by the receive *signal-to-noise ratio* (SNR)] and the expected update-truncation ratio. Consider the scheduling of cell-interior devices to constrain path loss. This gives rise to the other tradeoff between the receive SNR and fraction of data exploited in learning. Next, the latency-reduction ratio of the proposed BAA with respect to the traditional OFDMA scheme is proved to scale almost linearly with the device population. Experiments based on a neural network and a real dataset are conducted for corroborating the theoretical results.**

***Index Terms*— Edge intelligence, federated learning, multiple access, over-the-air computation.**

I. Introduction

THE traffic in mobile Internet is growing at a breath-taking rate due to the extreme popularity of mobile devices (e.g., smartphones and sensors). Analysis shows that there will be 80 billions of devices connected to Internet by 2025, resulting in a tenfold traffic growth compared with 2016 [1]. The availability of enormous mobile data and recent breakthroughs in *artificial intelligence* (AI) motivate researchers to develop AI technologies at the network edge. Such technologies are collectively called *edge AI* and drive the latest trend in machine learning, i.e., *edge learning*, that concerns training of edge-AI models via computation at edge servers and devices [2]–[4]. The migration of learning from central clouds towards the edge allows edge servers to have fast access to real-time data generated by edge devices for fast training of AI models. In return, downloading the models from servers to devices in proximity provision the latter intelligence to respond to real-time events. While computing speeds are growing rapidly, wireless transmission of high-dimensional data by many devices suffers from the scarcity of radio resources and hostility of wireless channels, resulting in a communication bottleneck for fast edge learning [5], [6]. This calls for the design of low-latency multi-access schemes that integrate techniques from two different areas, namely distributed learning and wireless communication.

In this work, we propose one such scheme, called *broadband analog aggregation* (BAA), for low-latency implementation of a popular distributed-learning framework, called *federated learning* [6], [7], in a wireless network, referred to as *federated edge learning* (FEEL). As illustrated in Fig. 1, a key operation of FEEL is to aggregate local models trained on devices to update the global model at a server. The BAA realizes the operation over a broadband multi-access channel by exploiting simultaneous transmission and the resultant waveform superposition. This leads to dramatic latency reduction compared with the conventional orthogonal access. We develop the BAA framework by deriving the tradeoffs between a set of communication-and-learning metrics and quantifying the latency reduction compared with the conventional design.

A. Federated Edge Learning and Multiple Access

As mentioned, FEEL is a recently developed distributed-learning framework that preserves user-privacy by avoiding direct data uploading. To this end, a typical federated-learning algorithm alternates between two phases, as shown in Fig. 1. One is to aggregate distributed updates over a multi-access channel and apply their average to update the AI-model at the edge server. The other is to broadcast the model under training to allow edge devices to compute its updates using local datasets and then transmit the updates to the server for

Manuscript received January 16, 2019; revised June 3, 2019 and August 14, 2019; accepted September 30, 2019. Date of publication October 15, 2019; date of current version January 8, 2020. This work was supported by the Hong Kong Research Grants Council under Grant 17208319, Grant 17209917, and Grant 17259416. The associate editor coordinating the review of this article and approving it for publication was C. Shen. *(Corresponding author: Kaibin Huang.)*

G. Zhu was with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong. He is now with the Shenzhen Research Institute of Big Data, Shenzhen 518000, China (e-mail: gxzhu@sribd.cn).

Y. Wang and K. Huang are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: wangyong@eee.hku.hk; huangkb@eee.hku.hk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2019.2946245

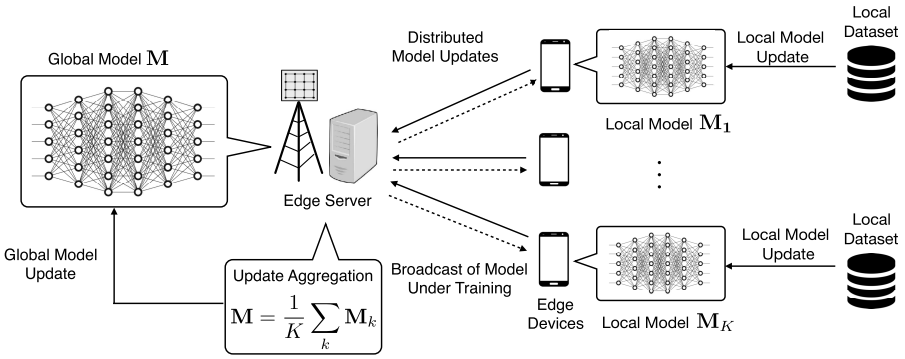


Fig. 1. Federated edge learning from wirelessly distributed data.

aggregation. The iteration continues until the global model converges and each iteration is called a *communication round*. The updates computed locally at edge devices can be either the model parameters [6] or gradient vectors [7], giving rise to two implementation approaches, i.e., *model-averaging* and *gradient-averaging*.

In view of high dimensionality in updates (each contains e.g., millions of parameters), a main theme in the FEEL research is to develop communication-efficient strategies for fast update-uploading to accelerate learning. There exist three main approaches. The first addresses the *straggler issue*, namely that the slow devices (stragglers) dominate the overall latency due to update synchronization required for aggregation. To reduce latency, a partial averaging scheme is proposed in [8] where only a portion of updates from those fast-responding devices are used for global model updating, while those from stragglers are discarded. Later, the design was enhanced by coding the updates such that the full update-averaging can be still realized using only a portion of coded updates [9]. The second approach also aims at reducing the number of transmitting devices, but the scheduling criterion is update significance instead of computation speed [10], [11]. For model averaging implementation, the update significance is measured by the *model variance* which indicates the divergence of a particular local model from the average across all local models [10]. On the other hand, for gradient averaging, the update significance is measured by *gradient divergence* that reflects the level of change on the current gradient update w.r.t. the previous one [11]. The last approach focuses on update compression by exploiting the sparsity of gradient updates [12], [13].

The prior work by computer scientists focuses on reducing the number of updating devices and compressing the information for transmission. It represents a computer-science approach for tackling the communication-latency problem in the FEEL systems. Wireless channels therein are abstracted as "bit pipes" that overlook the possibility of exploiting the channels' sophisticated properties (e.g., fading, multi-access, and spatial multiplexing) for latency reduction. Thus, a more direct and perhaps more fundamental approach for solving this communication problem is to develop wireless communication techniques to support low-latency FEEL. We adopt the new approach in this work and focus on designing a multi-access scheme for communication-efficient FEEL. The classic orthogonal-access schemes (e.g., OFDMA or TDMA) have been designed for supporting independent links. Their applications to edge learning can cause the multi-access latency to scale linearly with the number of edge devices and thus are inefficient. To overcome the drawback, we propose the low-latency BAA scheme for leveraging simultaneous broadband transmissions to implement update aggregation "over-the-air" in FEEL systems.

B. Over-the-Air Computation

The current BAA scheme builds on the classic idea of *over-the-air computation* (AirComp). The idea of AirComp can be traced back to the pioneering work studying functional computation in sensor networks [14]. The design relies on structured codes (i.e., lattice codes) to cope with channel distortion introduced by the multi-access channel. The significance of the work lies in its counter-intuitive finding that "interference" can be harnessed to help computing. It was subsequently discovered in [15] that simple analog transmission without coding but with channel pre-equalization can achieve the minimum distortion if the data sources are *independent and identically distributed* (i.i.d.) Gaussian. Nevertheless, coding can be still useful for other settings if the sources follow more complex distributions such as bivariate Gaussian [16] and correlated Gaussian [17]. The satisfactory performance and simplicity of analog AirComp has led to an active area focusing on robust design and performance analysis [18]–[21]. In particular, techniques for distributed power control and robust AirComp against channel estimation errors are proposed in [18] and [19], respectively. Theoretical analysis on the AirComp outage performance under a distortion constraint and the computation rate, defined as the number of functions computed per time slot, were provided in [20] and [21], respectively. Another vein of research focuses on transforming AirComp from theory into practice by prototyping [22] and addressing the practical issue of synchronization over sensors [23], [24]. In [23], the authors proposed to modulate the data into transmission power to relax the synchronization requirement such that only coarse block-synchronization is required for AirComp. An alternative scheme, called *AirShare*, is developed in [24] which broadcasts a shared clock to all devices to enforce synchronization.

Advancing beyond scalar-valued function computation, the latest trend in the area also explores multiple-input-mutiple-output (MIMO) techniques to enable vector-valued function computation [25]–[27], referred as MIMO AirComp. In particular, a comprehensive framework for MIMO AirComp that consists of beamforming optimization and limited-feedback design is proposed in [25]. The framework was extended in subsequent work to wirelessly-powered AirComp system [26], where the beamformer was jointly optimized with the wireless power control to further reduce the AirComp distortion, and massive MIMO AirComp system [27], where a reduced-dimension two-tier beamformer design was developed by exploiting the clustered channel structure to reduce channel-feedback overhead and signal processing complexity.

Prior work on AirComp targets sensor networks and thus focuses on narrow-band systems only. The reason is that sensor networks typically require low-rate transmission and communicate over a dedicated narrow-band in practical systems, e.g., LTE for narrow-band Internet-of-Things (NB-IoT) [28]. In contrast, we consider broadband channels due to the transmission of high-dimensional updates in the FEEL systems. However, the solutions for broadband AirComp do not exist in the existing literature. Such design as well as the study of the effects of AirComp on the edge-learning performance is an area largely uncharted. This motivates the development of the BAA scheme and its performance analysis in the current work.

C. Contribution and Organization

We consider the implementation of FEEL in a single-cell wireless network. The BAA scheme is proposed to reduce the communication latency in the network, which is described as follows. Each device transmits a high-dimensional update (local model) in blocks over a broadband channel, using linear analog modulation for modulating individual parameters and *orthogonal frequency division multiplexing* (OFDM) for partitioning the channel into sub-channels. Realizing over-the-air update aggregation requires the received model parameters from different devices to have identical amplitudes, called *amplitude alignment*. This is achieved by broadband channel inversion at each device. The channel inversion and simultaneous analog OFDM transmission by a set of scheduled devices allow the server to receive the desired average of the local models/updates computed at the devices.

Based on the scheme, we develop the BAA framework using a random network model where a number of edge devices are randomly distributed in a disk area. To describe our findings, it is necessary to introduce some metrics related to the FEEL-network performance as follows:

1) *Receive SNR*: Given amplitude alignment, the receive *signal-to-noise ratios* SNRs for updates transmitted by different devices are identical. The metric is one quality measure of model update in FEEL.
2) *Truncation ratio*: This is another update-quality measure. It refers to the expected ratio of model parameters being truncated due to channel inversion at a device under a transmit-power constraint.
3) *Fraction of exploited data*: It refers to the fraction of the distributed dataset exploited in learning. Given uniform data distribution over devices, the metric is related to the fraction of scheduled devices.

The findings and the contributions from the framework development are as follows:

- **Two Communication-and-Learning Tradeoffs**: The first tradeoff as derived in closed-form is between the receive SNR and the truncation ratio, called the *SNR-truncation* tradeoff. The tradeoff also shows that the receive SNR is limited by the path-loss of the furthest device from the server, which is due to the said amplitude alignment among devices. This suggests that the receive SNR can be improved via scheduling cell-interior devices for FEEL. However, this causes data loss and thereby gives rise to the second tradeoff, namely the one between the receive SNR and the fraction of exploited data. It is referred to as *(update)-reliability-(data)-quantity tradeoff*. The above two tradeoffs are fundamental for FEEL systems with BAA and can be useful tools for further research in this direction. In addition, to improve this tradeoff by copying with data deficiency, we propose two scheduling schemes: one is to exploit mobility and the other to alternate the scheduling of cell-interior and cell-edge devices.
- **Communication-Latency Analysis**: The latency reduction of BAA is quantified with respect to (w.r.t.) the conventional multi-access scheme, namely *orthogonal frequency division multiple access* (OFDMA) with digital modulation. The latency-reduction ratio is proven to increase with K, the number of scheduled devices, as $O\left(\frac{K}{\log K}\right)$. The result shows that BAA is a promising solution for low-latency FEEL with many devices.
- **Experiments**: The FEEL system is implemented in software for an AI application of handwritten-digit recognition, where the AI-model is based on a neural network and a real image dataset. Experimental results demonstrate the derived communication-learning tradeoffs. Moreover, the results confirm the dramatic communication-latency reduction achieved by BAA w.r.t. the conventional design.

In addition, in the extended version of the work [29], extensions of the BAA scheme are also presented to address two issues, namely security against adversarial attacks and beamforming for improving link reliability of cell-edge devices.

Last, it is worth mentioning that upon the completion of this work [29], interesting parallel work [30], [31] was also reported that share the common theme of applying AirComp to update aggregation in FEEL. Nevertheless, the specific systems and designs therein differ from the current work. Both paralelled work targets narrow-band single/multi-antenna channels and focuses on algorithmic design to improve system performance. Specifically, a source-coding algorithm that exploits the sparsity in gradient update is proposed in [30] for compressed update transmission. A device-selection algorithm is developed in [31] to maximize the number of scheduled devices under a update-distortion constraint.

Organization: The remainder of the paper is organized as follows. Section II introduces the system and channel models. Section III presents the proposed BAA scheme and motivates the user scheduling problem. Practical scheduling schemes for BAA is presented in Section IV and the involved tradeoff is quantified. The latency performance of the proposed BAA is analytically compared with the digital counterpart in Section V. Section VI shows the experimental results using real dataset. Discussion on possible extensions is provided in Section VII, followed by concluding remarks in Section VIII.

II. System Model

A. Federated Edge Learning System

We consider a FEEL system comprising a single edge server and K edge devices as shown in Fig. 1. A shared AI model (e.g., a classifier), represented by the parameter set $\mathbf{w}$, is trained collaboratively across the edge devices. Each device collects a fraction of labelled training data via the interaction with its own user, constituting a local dataset.

To facilitate the learning, the loss function measuring the model error is defined as follows following the existing literatures [4]–[13], [32]. Let $\mathcal{D}_k$ denote the local dataset collected at the k-th edge device. The *local loss function* of the model vector $\mathbf{w}$ on $\mathcal{D}_k$ is given by

$$F_k(\mathbf{w}) = \frac{1}{|\mathcal{D}_k|} \sum_{(\mathbf{x}_j, y_j) \in \mathcal{D}_k} f(\mathbf{w}, \mathbf{x}_j, y_j), \tag{1}$$

where $f(\mathbf{w}, \mathbf{x}_j, y_j)$ is the sample-wise loss function quantifying the prediction error of the model $\mathbf{w}$ on the training sample $\mathbf{x}_j$ w.r.t. its ground-true label y_j. For convenience, we rewrite $f(\mathbf{w}, \mathbf{x}_j, y_j)$ as $f_j(\mathbf{w})$ and assume uniform sizes for local datasets: $|\mathcal{D}_k| \equiv D$, for all k. Then, the *global loss function* on all the distributed datasets can be written as

$$F(\mathbf{w}) = \frac{\sum_{j \in \cup_k \mathcal{D}_k} f_j(\mathbf{w})}{|\cup_k \mathcal{D}_k|} = \frac{1}{K} \sum_{k=1}^{K} F_k(\mathbf{w}). \tag{2}$$

The learning process is thus to minimize the global loss function $F(\mathbf{w})$, namely,

$$\mathbf{w}^* = \arg\min F(\mathbf{w}). \tag{3}$$

One way for computing $F(\mathbf{w})$ is to directly upload all local data, which causes the privacy issue. To tackle the issue, the FEEL framework is employed to solve the problem in (3) in a distributed manner. We focus on the model-averaging implementation in the subsequent exposition while the same principle also applies to the alternative implementation based on gradient-averaging.

For implementing FEEL, in each communication round, say the n-th round, the edge server broadcasts the current model under training $\mathbf{w}[n]$ to all edge devices. Starting from $\mathbf{w}[n]$, each device updates its own model by running τ-step ($\tau \geq 1$) *stochastic gradient descent* (SGD) towards minimizing the loss function defined in (1). Mathematically, for device k, a single-step SGD updates the local model $\mathbf{w}_k$ via:

$$\mathbf{w}_k[n+1] = \mathbf{w}[n] - \eta \nabla F_k(\mathbf{w}[n]), \tag{4}$$

where η is the step size and ∇ represents the gradient operator. Then, a τ-step SGD repeats the updating rule in (4) for τ times. Upon its completion, the local model-updates are sent to the edge server for averaging and updating the global model $\mathbf{w}$ as follows:

$$\mathbf{w}[n+1] = \frac{1}{K} \sum_{k=1}^{K} \mathbf{w}_k[n+1]. \tag{5}$$

The learning process involves the iteration between (4) and (5) until the model converges.

As observed from (5), it is only the aggregated model, i.e.,

$$\text{(Model aggregation)} \quad \mathbf{y} = \sum_{k=1}^{K} \mathbf{w}_k, \tag{6}$$

instead of individual model-updates $\{\mathbf{w}_k\}$, needed at the edge server for model averaging. This motivates the low-latency BAA scheme exploiting AirComp as presented in Section III.

B. Broadband Channel and Update Transmission

The uploading of model-updates from edge devices to the server is through a broadband multi-access channel. To cope with the frequency selective fading and inter-symbol interference, the OFDM modulation is adopted to divide the whole bandwidth B to M orthogonal sub-channels. To exploit AirComp for low-latency model aggregation, model-updates are amplitude-modulated for analog transmission. Also, each sub-channel is dedicated for one model-parameter transmission.

During the model updating phase, all devices transmit simultaneously over the whole available bandwidth. We assume symbol-level synchronization among the transmitted devices through a synchornization channel (e.g., "timing advance" in LTE systems [33]).[1] Let $\mathbf{w}_k = [w_{k,1}, \cdots, w_{k,q}]^T$ denote the $q \times 1$ local-model parameter-vector from the k-th device, where q denotes the number of model parameters. To facilitate the power-control design and reduce transmission power, the transmitted symbols, denoted by $\{\tilde{w}_{k,i}\}$, are normalized model parameters such that they have zero mean and unit variance, i.e., $\mathsf{E}(\tilde{w}_{k,i}\tilde{w}_{k,i}^*) = 1$, $\forall k, i$. Note that the normalization factor for each model dimension is uniform for all devices and can be inverted at the edge server (see Appendix A for more details). At each communication round, the model-updating duration consists of $N_s = \frac{q}{M}$ OFDM symbols. In particular, the i-th aggregated model parameter, denoted by y_i, with $i = (t-1)M + m$, received at the m-th sub-carrier, t-th OFDM symbol, and ℓ-th communication

[1]The reliability of the synchronization is proportional to the bandwidth dedicated for the synchronization channel. Particularly, the current state-of-the-art phase lock loop can achieve a synchronization offset of $0.1B_s^{-1}$, where B_s is the amount of bandwidth used for synchronization. In the existing LTE systems, the typical value of B_s is 1MHz. Thus, a sufficiently low synchronization offset of $0.1\mu sec$ can be achieved. Note that in a broadband OFDM system, as long as the synchronization offset is smaller than the length of cyclic prefix (the typical value is $5\mu sec$ in the LTE systems), the offset simply introduces a phase shift to the received symbol. The phase shift can be compensated by channel equalization, incurring no performance loss [34].

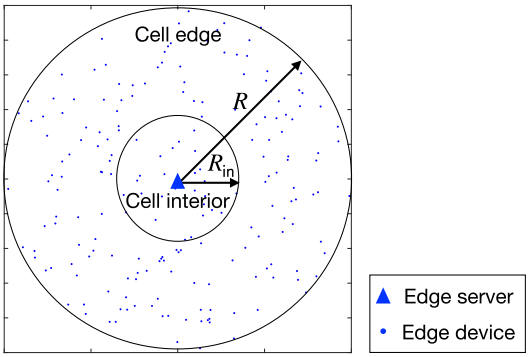


Fig. 2. A single-cell edge learning network with uniformly-distributed edge devices.

round, is given by

$$y_i[\ell]=\sum_{k=1}^{K} r_k^{-\frac{\alpha}{2}} h_k^{(m)}[t,\ell]p_k^{(m)}[t,\ell]w_{k,i}[\ell]+z^{(m)}[t,\ell], \quad \forall i \quad (7)$$

where $r_k^{-\frac{\alpha}{2}}$ captures the path-loss of the link between device k and the edge server, with r_k denoting the distance between them and α representing the path-loss exponent; the small-scale fading of the channel is captured by $h_k^{(m)}[t,\ell]$ which follows Rayleigh fading and is *identically and independently distributed* (i.i.d.) over the indexes of k, m, t, ℓ, yielding $h_k^{(m)}[t,\ell] \sim \mathcal{CN}(0,1)$. $\{p_k^{(m)}[t,\ell]\}$ are the associated power control policies on the transmitted updates to be designed in the sequel. Last, $z^{(m)}[t,\ell]$ models the i.i.d. *additive white Gaussian noise* (AWGN) following $\mathcal{CN}(0,1)$. For ease of notation, we skip the index of OFDM symbol t and that of communication round ℓ in the subsequent exposition whenever no confusion is incurred.

The power allocation over sub-channels, $\{p_k^{(m)}\}$, will be adapted to the channel coefficients, $\{h_k^{(m)}\}$, for implementing the BAA as presented in the sequel. The transmission of each device is subject to the long-term transmission power constraint:

$$\mathsf{E}[|\sum_{m=1}^{M} |p_k^{(m)}(h_k^{(m)})|^2|] \leq P_0, \quad \forall k, \quad (8)$$

where the the expectation is taken over the random channel coefficients. Since channel coefficients are i.i.d. over different sub-channels, the above power constraint reduces to

$$\mathsf{E}[|[p_k^{(m)}(h_k^{(m)})|^2|] \leq \tfrac{P_0}{M}, \quad \forall k. \quad (9)$$

C. Network Topology

We consider a single-cell network distribution in a disk. Specifically, the edge devices are i.i.d. distributed over a disk centred at the edge server with a cell-radius R. Thus the *probability density function* (PDF) of the distance r_k is

$$f_{r_k}(r) = \frac{2r}{R^2}, \quad 0 \leq r \leq R. \quad (10)$$

Fig. 2 illustrates one realization of the random network. The cell is divided into two non-overlapping parts: cell-interior and cell-edge. Specifically, the area within a range of distance R_{in} from the server is referred to as the *cell-interior* while that outside the range as the *cell-edge*.

III. Broadband Analog Aggregation: Scheme and Properties

In this section, the proposed BAA scheme for FEEL is first presented. Then the resultant SNR-truncation tradeoff is derived via analyzing the receive SNR and update-truncation ratio.

A. The Scheme of Broadband Analog Aggregation

1) Transmitter Design: To enable BAA, the transmitter design for edge devices is shown in Fig. 3(a). As highlighted, the design differs from the classic OFDM transmitter by replacing digital modulation (e.g., QAM) with analog one and adding channel-inversion power control.

The new signal-processing operations in the transmitter are described as follows. The local-model parameters are first amplitude-modulated into symbols. The long symbol sequence is divided into blocks. Each is transmitted in a single OFDM symbol with one parameter over one frequency sub-channel. Assuming perfect CSI at the transmitter, sub-channels are inverted by power control so that model parameters transmitted by different devices are received with identical amplitudes, achieving amplitude alignment at the receiver as required for BAA. Nevertheless, a brute-force approach is inefficient if not impossible under a power constraint since some sub-channels are likely to encounter deep fades. To overcome the issue, we adopt the more practical *truncated channel inversion*. To be specific, a sub-channel is inverted only if its gain exceeds a so called *power-cutoff threshold*, denoted by g_{th}, or otherwise allocated zero power. Then the transmission power of the k-th device on the m-th sub-channel, $p_k^{(m)}$, is given by

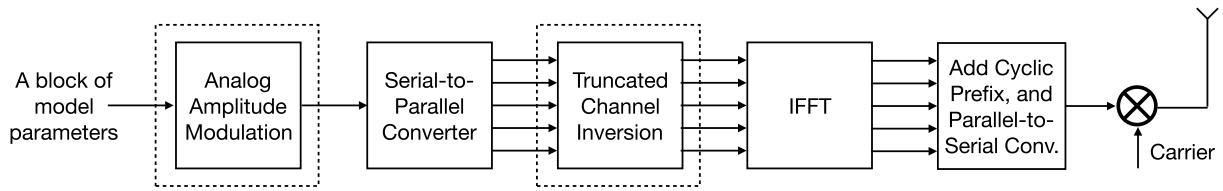
$$p_k^{(m)} = \begin{cases} \frac{\sqrt{\rho_k}}{r_k^{-\frac{\alpha}{2}} h_k^{(m)}}, & |h_k^{(m)}|^2 \geq g_{\text{th}} \\ 0, & |h_k^{(m)}|^2 < g_{\text{th}}, \end{cases} \quad (11)$$

where ρ_k is a scaling factor set for ensuring the average-transmit-power constraint in (9). One can see from (7) that, ρ_k also determines the receive SNR of the model-update from each device.

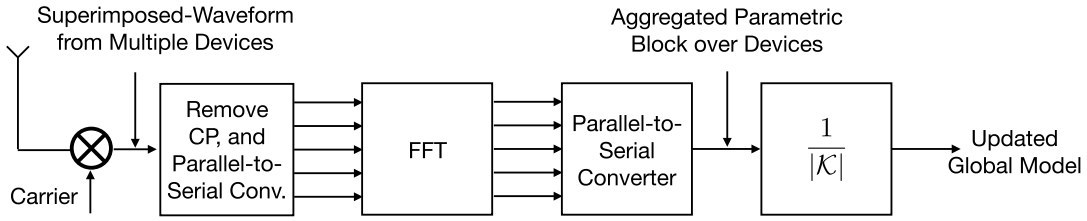
We remark that the policy can cause the loss of those model parameters that are mapped to the truncated sub-channels. To measure the loss, we define the truncation ratio as $\zeta = \frac{\text{\# of truncated parameters}}{\text{\# of total model-update parameters}}$ and analyze it in the sequel. Other operations in Fig. 3(a) follow the conventional design. Their details are omitted for brevity.

2) Receiver Design: Fig. 3(b) shows the receiver design for edge server. It has the same architecture as the conventional OFDM receiver except that the digital demodulator is replaced with a post-processor that scales the received signals to obtain the average model. However, the received signals are different between the current and conventional designs as described below.

Consider an arbitrary communication round ℓ and a set of devices scheduled by the server to transmit their local models, which are represented by the index set $\mathcal{K}$. Given their



(a) Transmitter design for edge devices



(b) Receiver design for edge server

Fig. 3. Transceiver design for broadband analog aggregation.

simultaneous transmission, the server receives superimposed waveforms. By substituting the truncated-channel-inversion policy in (11) into (7), the server obtains the aggregated local-model block, denoted by a $M\times 1$ vector $\mathbf{y}[t,\ell]$, at the parallel-to-serial converter output [see Fig. 3(b)] as:

$$\mathbf{y}[t,\ell]=\sum_{k\in\mathcal{K}}\sqrt{\rho_k}\mathbf{w}_k^{(\mathsf{Tr})}[t,\ell]+\mathbf{z}[t,\ell], \tag{12}$$

where t and ℓ are the indexes of local-model block (OFDM symbol) and communication round as defined in (7). $\mathbf{w}_k^{(\mathsf{Tr})}[t,\ell]$ is a truncated version of $\mathbf{w}_k[t,\ell]=[w_{k,(t-1)M+1},\cdots,w_{k,tM}]^T$ with the truncated elements determined by the channel realizations according to (11) and represented by zeros. Note from (12) that $\{\sqrt{\rho_k}\}$ should be aligned to enforce the said amplitude alignment required for aggregation. Next, cascading all the N_s blocks and scale the result by the factor $\frac{1}{|\mathcal{K}|}$ gives the desired updated global model. Then, the server initiates the next communication round by broadcasting the model to all devices or complete the learning if the model converges.

Remark 1: It is worth pointing out that truncating model parameter at random in BAA may be beneficial to the training process as it can potentially prevent overfitting the model to the training data. The truncation resembles the celebrated anti-overfitting technique, called "dropout" [35], that truncates a function of neurons of the model at random during training. The technique is particularly useful for training a deep neural network (DNN). The rationale behind dropout is that a DNN usually consists of a large amount of neurons which may learn interdependent *features from the data. The interdependency among neurons makes the resultant model overfitted to the training data and thus reduces its generalization capability. By cutting redundant neurons, dropout can reduce interdependent learning among neurons and thus improve the learning performance.*

B. SNR-Truncation Tradeoff

Targeting the BAA scheme, we show in this sub-section that there exists a tradeoff between the receive SNR (identical for all devices) and the truncation ratio defined in the preceding sub-section, which is regulated by the power-cutoff threshold in (11).

First, substituting (11) into (9), yields the maximum receive SNR of a model-update as follows.

Lemma 1 *(Maximum receive SNR): Consider the k-th edge device with the propagation distance r_k, the maximum receive SNR of the update transmitted by the device is bounded as*

$$\rho_k\leq\frac{P_0}{Mr_k^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}, \tag{13}$$

where $\mathsf{Ei}(x)$ is the exponential integral function defined as $\mathsf{Ei}(x)=\int_x^{\infty}\frac{1}{t}\exp(-t)dt$. The equality is achieved when the device transmits with the maximum average power P_0.

Proof: Let $g_k=|h_k|^2$ denote the channel gain of the k-th link. Since the channel coefficient is Rayleigh distributed $h_k\sim\mathcal{CN}(0,1)$, it yields that g_k follows the exponential distribution with unit mean. Then substituting (11) into (9) gives

$$\frac{\rho_k}{r_k^{-\alpha}}\int_{g_{\mathsf{th}}}^{\infty}\frac{1}{g}\exp(-g)dg\leq\frac{P_0}{M}. \tag{14}$$

The desired result follows by invoking the definition of the exponential integral function. ■

Lemma 1 indicates that the maximum receive SNR for a model-update is limited by the propagation distance. For BAA implementation, near devices need lower their transmission power to achieve amplitude alignment with far devices. This results in a receive SNR identical for all devices, denoted as ρ_0. It follows from Lemma 1 that

$$\text{(Receive SNR)}\quad\rho_0=\frac{P_0}{Mr_{\max}^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}, \tag{15}$$

where $r_{\max}=\max_k\{r_1,r_2,\cdots,r_K\}$ denotes the distance from the edge server to the furthest active device. The result suggests the need of limiting the distances of devices by scheduling, which is explored in Section IV.

Besides affecting the receive SNR, the power-cutoff threshold g_{th} also regulates the truncation ratio. In particular, when

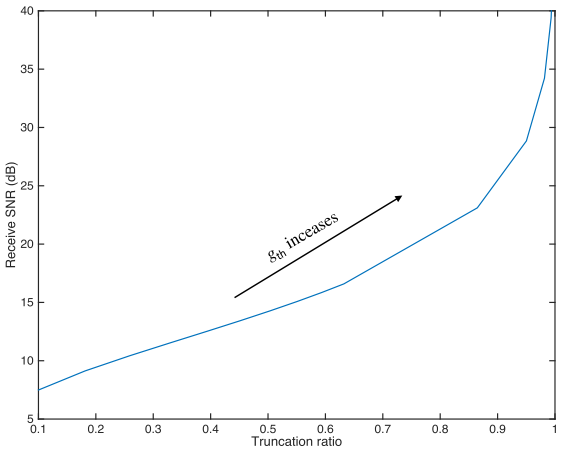


Fig. 4. Illustration of the SNR-truncation tradeoff with $P_0 = 0.1$(W), $M = 1000$, $r_{\max} = 100$, $N_0 = -80$ dBm and $\alpha = 3$.

an update contains sufficiently many parameters, by law of large number, its truncation ratio is equal to the corresponding channel cutoff probability as derived below.

Lemma 2 *(Truncation Ratio): When the model-update dimension $q \to \infty$, the truncation ratio ζ is equal to the channel-cutoff probability:*

$$\zeta \to \mathsf{Pr}(|h_k|^2 < g_{\mathsf{th}}) = 1 - \exp(-g_{\mathsf{th}}), \quad q \to \infty \tag{16}$$

Proof: The result immediately follows from the exponential distribution of the channel gain. ∎

Combining Lemmas 1 and 2, we derive the said SNR-truncation tradeoff as follows.

Proposition 1 *(SNR-Truncation Tradeoff): Given the BAA scheme, the relationship between the receive SNR ρ_0 and the truncation ratio ζ is specified by the following equation:*

$$\rho_0 = \frac{P_0}{M r_{\max}^{\alpha} \mathsf{Ei}(-\ln(1-\zeta))}, \quad q \gg 1. \tag{17}$$

Remark 2 (SNR-Truncation Tradeoff and Power-Cutoff Threshold): An exemplary SNR-truncation tradeoff curve for typical system settings is plotted in Fig. 4. The said tradeoff is controlled by the power-cutoff threshold g_{th}. Particularly, an increasing g_{th} tends to increase the receive SNR at the cost of more truncated parameters in model updates and vice versa. Since both affect the receive-update quality at the server, the threshold being too high or too low degrades the learning performance. Thus it is necessary to optimize the threshold, which is a design issue warranting further investigation. In experiments, the power-cutoff threshold is optimized numerically to optimize the learning performance by a grid search.

IV. Broadband Analog Aggregation: Scheduling

The preceding result in Proposition 2 shows that the bottleneck of the receive SNR of model updates is the device with the longest propagation distance. Then to ensure the update reliability, it is desirable to constrain the distance of active devices from the server. This motivates the following scheduling scheme.

> ***Scheme 1*** *(Cell-Interior Scheduling): The edge server schedules only the cell-interior edge devices whose distances are no larger than a distance threshold R_{in}.*

For the purpose of comparison, consider the baseline scheme of simply scheduling all available devices, called *all-inclusive scheduling*. Compared with the baseline scheme, even though cell-interior scheduling improves the update reliability, it has the drawback of *data deficiency* since the resultant model-training fails to exploit data at cell-edge devices. As a basic property of cell-interior scheduling, we derive in the sequel a tradeoff between the SNR gain (w.r.t. all-inclusive scheduling) and the fraction of exploited data, called the (update)-reliability-(data)-quantity tradeoff, by analyzing the two metrics. In the last subsection, schemes for coping with data deficiency are discussed.

A. Fraction of Exploited Data

Assuming equal data partitioning among all edge devices, we can first establish the relationship between the fraction of exploited data, denoted by F_{DAT}, and the fraction of scheduled users, denoted by F_{USE}, as follows:

$$F_{\mathsf{DAT}} = (1-\beta) F_{\mathsf{USE}}, \tag{18}$$

where the factor $(1-\beta)$ is added to exclude the extreme event that a scheduled device is totally truncated (dose not transmit any parameter) due to poor channel conditions and thus the data in the device is not exploited, with β denoting the *device truncation probability* given by

$$\beta = [1 - \exp(-g_{\mathsf{th}})]^q. \tag{19}$$

Given (18), we can turn the derivation of F_{DAT} to that of F_{USE}. To derive F_{USE}, it requires the distribution of the number of scheduled devices within the range of R_{in}, denoted K_{in}. The distribution is parameterized by R_{in}, R, and K and derived below.

Lemma 3 *(Distribution of the Number of Scheduled Devices): In cell-interior scheduling, given the distance threshold R_{in}, the number of scheduled users follows a Binomial distribution with the* probability mass function *(PMF) given by:*

$$\mathsf{Pr}(K_{\mathsf{in}} = k) = \binom{K}{k} \left(\frac{R_{\mathsf{in}}^2}{R^2}\right)^k \left(1 - \frac{R_{\mathsf{in}}^2}{R^2}\right)^{K-k}, \tag{20}$$

Proof: See Appendix B. ∎

By Lemma 3 and (18), one can easily obtain the expected fraction of exploited data as follows:

Proposition 2 *(Expected Fraction of Exploited Data): Given cell-interior scheduling with the distance threshold R_{in}, the expected fraction of exploited-data is given by*

$$F_{\mathsf{DAT}} = (1-\beta)\mathsf{E}\left(\frac{K_{\mathsf{in}}}{K}\right) = (1-\beta)\left(\frac{R_{\mathsf{in}}}{R}\right)^2. \tag{21}$$

B. Receive SNR Gain

To characterize the receive SNR gain, the expected receive SNRs for all-inclusive scheduling and cell-interior scheduling are analyzed. Their ratio gives the desired SNR gain.

1) All-Inclusive Scheduling: In order to derive $\mathsf{E}(\rho_0)$ with ρ_0 defined in (15), the distribution of $r_{\max} = \max_k\{r_1, r_2, \cdots, r_K\}$ is required which is provided as follows.

Lemma 4 *(Distribution of Maximum Distance): The PDF of the maximum distance $r_{\max}$ under the uniform user distribution in* (10) *is given by*

$$f_{r_{\max}}(r) = \frac{2K}{R^{2K}} r^{2K-1}. \tag{22}$$

The result follows straightforwardly from (10) *and the proof is omitted.*

Using Lemma 4, the expected receive SNR for all-inclusive scheduling is derived below.

Lemma 5 *(Expected Receive SNR for All-Inclusive Scheduling): By employing all-inclusive scheduling, the resultant expected receive SNR is given by*

$$\mathsf{E}(\rho_0) = \frac{2K}{2K-\alpha} \frac{P_0}{MR^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}, \quad K > \frac{\alpha}{2}. \tag{23}$$

Proof: See Appendix C. ∎

Since the path-loss exponent $\alpha \in [3, 4]$ in practice, the requirement of $K > \frac{\alpha}{2}$ for the above result can be easily satisfied by having the number of edge devices $K > 2$.

2) Cell-Interior Scheduling: For cell-interior scheduling, the following result can be derived.

Lemma 6 *(Expected Receive SNR for Cell-Interior Scheduling): By employing cell-interior scheduling, the resultant expected receive SNR is given by*

$$\mathsf{E}[\rho_0(R_{\mathsf{in}})] = \frac{c(R_{\mathsf{in}})P_0}{MR_{\mathsf{in}}^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}, \tag{24}$$

where $c(R_{\mathsf{in}})$ is a bounded scaling factor depending on R_{in} and K with

$$c(R_{\mathsf{in}}) = \sum_{k=2}^{K} \frac{2k}{2k-\alpha} \binom{K}{k} \left(\frac{R_{\mathsf{in}}^2}{R^2}\right)^k \left(1 - \frac{R_{\mathsf{in}}^2}{R^2}\right)^{K-k}.$$

Particularly, for the typical case that $\alpha = 3$, we can show that $1 \le c(R_{\mathsf{in}}) \le 4$.

Proof: See Appendix D. ∎

A direct comparison between Lemma 5 and 6 yields the SNR gain of cell-interior scheduling over the all-inclusive counterpart as shown below.

Proposition 3 *(SNR Gain of Cell-Interior Scheduling): Given the distance threshold R_{in}, the cell-interior scheduling can attain the following receive SNR gain over the all-inclusive scheduling:*

$$G_{\mathsf{SNR}} = \frac{\mathsf{E}[\rho_0(R_{\mathsf{in}})]}{\mathsf{E}(\rho_0)} = a\left(\frac{R}{R_{\mathsf{in}}}\right)^{\alpha}, \tag{25}$$

where $a = \frac{2K-\alpha}{2K} c(R_{\mathsf{in}})$ is a bounded scaling factor, with $c(R_{\mathsf{in}})$ given in Lemma 6.

Note from Propositions 2 and 3 that both the fraction of exploited data and the SNR gain of the cell-interior scheduling are non-linear functions of the range ratio $\frac{R_{\mathsf{in}}}{R}$, but with different exponent scalings: the former is the square power law while the latter being a power law with the exponent equal to the path-loss exponent α.

C. Reliability-Quantity Tradeoff

Based on Proposition 2 and 3, the mentioned tradeoff between update reliability and data quantity can be derived as follows.

Proposition 4 *(Reliability-Quantity Tradeoff): When cell-interior scheduling is employed, the tradeoff between the SNR gain and the fraction of exploited data for model training is given by*

$$G_{\mathsf{SNR}} = a\left(\frac{1-\beta}{F_{\mathsf{DAT}}}\right)^{\frac{\alpha}{2}}. \tag{26}$$

Proposition 4 suggests that the path-loss exponent α plays a crucial role in determining how much SNR gain can be attained at the cost of losing a fraction $(1 - F_{\mathsf{DAT}})$ of training data. The larger the value of α is, the higher the cost is. Next, the result also provides a guideline for the selection of the distance threshold R_{in}. Generally speaking, for larger α, the learning performance is more *SNR-limited*. It is thus desired to have a smaller R_{in} to alleviate the SNR penalty due to the scheduling of cell-edge devices. In contrast, for smaller α, the learning performance is limited by the size of training dataset, and thus *data-limited*. Thereby, it is preferable to increase R_{in} to include more remote data at the edge to the training dataset with a degraded but acceptable receive SNR (in terms of its effect to the learning performance).

We note that the exact convergence rate w.r.t. to the received SNR and the number of scheduled devices is challenging to derive as how the received SNR and the number of scheduled users affect the convergence rate has a complex dependence on the data distribution, learning task and the learning model. The challenge will be formally tackled in a follow-up paper. Some key insights are listed as follows:

- (**Convergence guarantee**). The convergence of FEEL by BAA is guaranteed as long as the loss function of the model is a smooth function which is the case even for neural networks.
- (**Effect of channel noise and fading**). The channel noise and fading slow down the model convergence by introducing a up-scaling and a positive bias terms to the gradient norm. Due to the increased gradient norm, more communication rounds are needed for convergence.
- (**Effect of number of scheduled devices**). Having more scheduled devices can accelerate the model convergence (reduce the gradient variance) by a scaling law of $O(1/K)$.

D. Coping With Data Deficiency

The data deficiency of cell-interior scheduling, namely, the failure of exploiting cell-edge data, may lead to learning performance degradation. Two methods for addressing the issue are discussed in the sequel.

1) High-Mobility Networks: Consider the scenario where devices have high mobility and their locations change rapidly over time. In this scenario, the cell-interior scheduling is known as *opportunistic scheduling*. Given high mobility, the

scheme can automatically cope with the data deficiency since an cell-edge device can enter the cell-interior in a subsequent communication round and be scheduled. For tractability, by assuming the locations of all devices are i.i.d. over communication rounds following the existing literature, e.g., [36]–[38], we can quantify this fact as follows.

Proposition 5*: In a high-mobility network with K devices and given a training period consisting of N_{CR} communication rounds, the probability that all data is exploited for learning is given by*

$$p_{\mathsf{all}} = \left(1-(1-p_{\mathsf{in}})^{N_{\mathsf{CR}}}\right)^K \approx 1 - K(1-p_{\mathsf{in}})^{N_{\mathsf{CR}}}, \quad N_{\mathsf{CR}} \to \infty, \tag{27}$$

where $p_{\mathsf{in}} = (1-\beta)\left(\frac{R_{\mathsf{in}}}{R}\right)^2$ denotes the probability that a device lies in the cell-interior and the device is not truncated, with β being the device truncation probability defined in (19).

Proof: The proof is straightforward by noting that the event that all data is exploited during N_{CR} communication rounds is equivalent to that all devices are *ever* in the cell-interior in N_{CR} communication rounds and not truncated. The detailed derivation is omitted for brevity. ■

Note that, as N_{CR} increases, the probability p_{all} approaches to 1 at an exponential rate. This justifies our claim that the opportunistic (cell-interior) scheduling can efficiently cope with the data deficiency issue by simply increasing N_{CR} to exploit the device mobility.

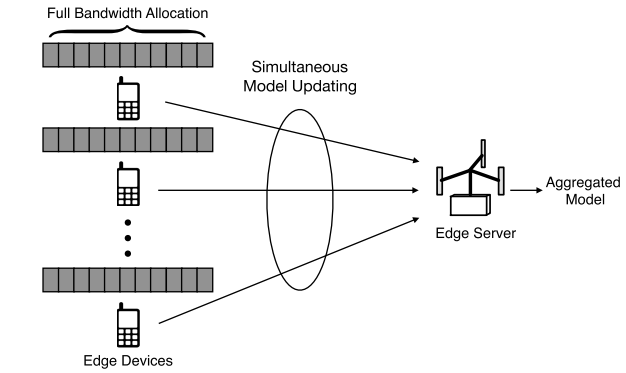
2) Low-Mobility Networks: However, in the low-mobility networks, the fraction of exploited data by cell-interior scheduling remains unchanged over communication rounds. For this reason, we propose the following alternating-scheduling scheme to exploit the cell-edge data.

> ***Scheme 2*** *(Edge-Interior Alternating scheduling): The edge server alternates between cell-interior schedulings (Scheme 1) and all-inclusive scheduling.*

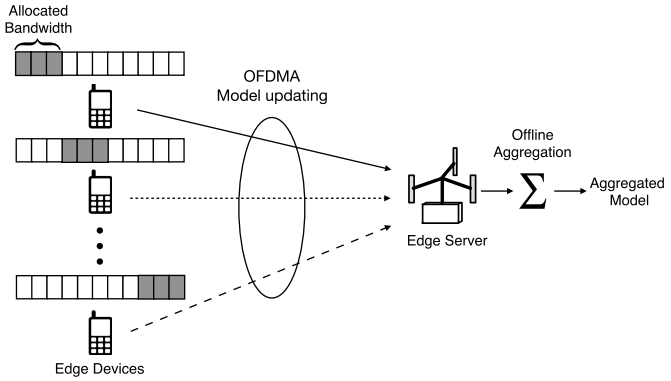
By alternating cell-interior and all-inclusive scheduling, the current scheme strikes a balance between the advantages/disadvantages of the two sub-schemes. In particular, alternating scheme can exploit all data for learning while achieving an effective receive SNR averaging those of the two sub-schemes. The alternating frequency between the two modes could be optimized to balance the reliability-quantity tradeoff for improving learning performance. As a result, the alternating scheduling can outperform both the cell-interior and all-inclusive schedulings.

V. Latency Analysis: BAA v.s. Broadband Digital Aggregation

The key advantage of the proposed BBA w.r.t. the conventional digital OFDMA is the significant reduction in communication latency. As illustrated in Fig. 5, the fundamental reason for the latency reduction is the difference in how the two schemes allocate the spectrum to devices. BAA allows the complete reuse of the whole bandwidth to exploit "interference" for direct aggregation. OFDMA orthogonalizes the bandwidth allocation to avoid interference for offering reliable communication. As a result, the bandwidth per device reduces with the number of devices. In this section, we analyze the latency reduction of BAA w.r.t. OFDMA.



(a) Broadband analog aggregation by AirComp



(b) Broadband digital aggregation by OFDMA

Fig. 5. Broadband analog aggregation versus broadband digital aggregation.

A. Latency Analysis of Broadband Analog Aggregation

For BAA, each model parameter is amplitude-modulated to a single analog symbol and each sub-channel is dedicated for a single parameter transmission. Thus, to upload a model update of dimension q, the total number of analog symbols to be transmitted is calculated as

$$D_{\mathsf{ana}} = q \text{ (symbols)}. \tag{28}$$

Since all devices transmit their model-updates simultaneously using all available sub-channels. One can easily derive that the number of OFDM symbols required for transmitting the whole update is equal to $\frac{q}{M}$. Thus, we obtain the following result.

Proposition 6 *(Latency for BAA): The latency per communication round for BAA is given by*

$$T_{\mathsf{ana}} = \frac{q}{M} T_s, \tag{29}$$

where T_s is the symbol duration of an OFDM symbol.

Two key observations can be made from the result in (29) as follows:

- Due to the complete reuse of radio resource (e.g., time and frequency) among devices, the resultant latency is

thus independent of the number of accessing devices, making it particularly promising in dense edge-learning network.
- The latency of the analog aggregation is a deterministic value independent of the channel realizations, which is in contrast to the digital counterpart whose latency is a random variable due to the channel-dependent transmission rate as will be shown in (34).

B. Latency Analysis of Broadband Digital Aggregation

For broadband digital aggregation (OFDMA), each parameter is first quantized into a fixed number of bits, denoted as Q. Then, for a device to upload a model update of dimension q, the total data amount to be transmitted is

$$D_{\mathsf{dig}} = qQ \text{ (bits)}. \tag{30}$$

During update aggregation, all K edge devises communicate with the edge server based on OFDMA to avoid inter-device interference. For simplicity, we assume that the total available bandwidth is evenly divided and assigned to K devices, so each device uploads its local model via an equal portion of allocated sub-channels [see Fig. 5(b)]. Thus the number of sub-channels allocated to device k is given by

$$M_k = \frac{M}{K}. \tag{31}$$

Thus the received signals from device k on the m-th sub-channel can be rewritten from (7) as

$$y_k^{(m)} = r_k^{-\frac{\alpha}{2}} h_k^{(m)} p_k^{(m)} x_k^{(m)} + z^{(m)}. \tag{32}$$

where the notations follow those in (7) and x_k denotes the quantized version of the model-update parameter. Since only a fraction of spectrum is used by each device, the power constraint in (9) is modified as follows:

$$\mathsf{E}\left[|p_k^{(m)}(h_k^{(m)})|^2\right] \leq \tfrac{KP_0}{M}, \quad \forall k. \tag{33}$$

In order to derive the model updating latency, the transmission rate of the system is needed. To this end, we consider the practical adaptive QAM modulation scheme [39]. It is well known that the optimal power control for such a scheme follows "water-filling" over channel realizations. The resultant transmission rate has no closed-form, making latency analysis intractable. The difficulty can be overcome by considering the sub-optimal power control, truncated channel inversion in (11). Then based on the result from [39], given a target *bit error rate* (BER), the resultant instantaneous transmission rate for device k on sub-channel m is:

$$R_k^{(m)} = \begin{cases} B_{\mathsf{sub}} \log_2\left(1 + \dfrac{-1.5\rho_k}{\ln(5\mathrm{BER})}\right), & |h_k^{(m)}|^2 \geq g_{\mathsf{th}} \\ 0, & |h_k^{(m)}|^2 < g_{\mathsf{th}}, \end{cases} \tag{34}$$

where $B_{\mathsf{sub}} = \frac{B}{M}$ denotes the sub-carrier spacing in the OFDM system and the receive SNR ρ_k can be easily derived by substituting (11) into (33):

$$\rho_k = \frac{KP_0}{Mr_k^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}. \tag{35}$$

By taking expectation of (34) w.r.t. sub-channel coefficient $h_k^{(m)}$ and summing over all the allocated sub-channel indeces $\{m\}_{m=1}^{M_k}$, the expected sum transmission rate for device k can be computed as follows:

$$\begin{aligned} R_k &= \mathsf{E}\left(\sum_{m=1}^{M_k} R_k^{(m)}\right) \\ &= M_k B_{\mathsf{sub}} \log_2\left(1 + \frac{-1.5\rho_k}{\ln(5\mathrm{BER})}\right) \mathsf{Pr}(|h_k^{(m)}|^2 \geq g_{\mathsf{th}}). \end{aligned} \tag{36}$$

Since $h_k^{(m)} \sim \mathcal{CN}(0,1)$, we have $\mathsf{Pr}(|h_k^{(m)}|^2 > g_{\mathsf{th}}) = \exp(-g_{\mathsf{th}})$. Thus (36) boils down to

$$R_k = M_k B_{\mathsf{sub}} \log_2\left(1 + \frac{-1.5KP_0}{\ln(5\mathrm{BER})Mr_k^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}\right)\exp(-g_{\mathsf{th}}). \tag{37}$$

Given (30) and (37), we derive the expected update communication latency for device k as

$$\begin{aligned} T_k &= \frac{D_{\mathsf{dig}}}{R_k} \\ &= \frac{KqQ}{MB_{\mathsf{sub}} \log_2\left(1 + \frac{-1.5KP_0}{\ln(5\mathrm{BER})Mr_k^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}\right)\exp(-g_{\mathsf{th}})}. \end{aligned} \tag{38}$$

Since the model aggregation is performed offline by the edge server after all local models are reliably received, the communication latency is determined by that of the slowest device, which is known as the *straggler effect*. Accordingly, we can establish the main result in the current sub-section as follows.

Proposition 7 *(Expected Latency for Broadband Digital Aggregation): The expected latency per communication round for broadband digital model aggregation is given by*

$$\begin{aligned} T_{\mathsf{dig}} &= \max_k\{T_k\} \\ &= \frac{KqQ}{M \log_2\left(1 + \frac{-1.5KP_0}{\ln(5BER)Mr_{\max}^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}\right)\exp(-g_{\mathsf{th}})} T_s, \end{aligned} \tag{39}$$

where $r_{\max} = \max_k\{r_1, r_2, \cdots, r_K\}$ *denotes the distance to the furthest user, and* $T_s = \frac{1}{B_{\mathsf{sub}}}$ *is the symbol duration of an OFDM symbol.*

Several key observations can be made from (39) as summarized below:
- The latency of the scheme scales approximately linearly with the number of devices K.
- Due to the straggler effect, the latency of the scheme is bottlenecked by the distance to the furthest user in the network r_{max}. The level of latency penalty for scheduling a far-away user is determined by the path-loss exponent α.
- The latency can be controlled by the target BER. Lower BER can accelerate the update aggregation but at a cost of degraded update-reliability and vice versa.

C. Latency Comparison Between Analog and Digital Aggregation

Combining Propositions 6 and 7, we are ready to derive the latency-reduction ratio of BAA w.r.t. the digital counterpart, defined as $\gamma = \frac{T_{\mathsf{dig}}}{T_{\mathsf{ana}}}$, as follows.

TABLE I
COMPARISON BETWEEN ANALOG AND DIGITAL AGGREGATION

	Broadband analog aggregation	**Broadband digital aggregation**
Effect of channel condition	Receive SNR and truncation ratio [see (15) & (16)]	Transmission rate [see (34)]
Distance dependency	Receive SNR depends on furthest user [see (15)]	Latency depends on furthest user [see (39)]
Latency scaling with # of devices	Independent [see (29)]	Approximately linear scaling [see (39)]
Update reliability guarantee	Loose guarantee by scheduling [see (25)]	Strict guarantee specified by target BER

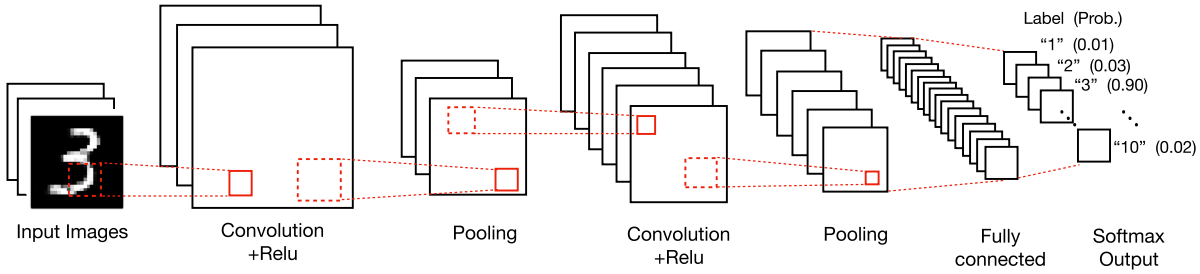


Fig. 6. Architecture of the convolutional neural network used in experiments.

Proposition 8 *(Latency Reduction): The latency-reduction ratio of the BAA over its digital counterpart, is given by*

$$\gamma = \frac{KQ}{\log_2\left(1 + \frac{-1.5KP_0}{\ln(5BER)Mr_{\max}^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})}\right)\exp(-g_{\mathsf{th}})}. \quad (40)$$

Based on Proposition 8, the following insights can be derived.

- The latency-reduction ratio scales linearly with the quantization resolution Q and approximately linearly with the number of devices K. More precisely, we have the following scaling law w.r.t. K:

$$\gamma = O\left(\frac{K}{\log_2 K}\right), \quad K \to \infty. \quad (41)$$

- The latency-reduction ratio can keep increasing unboundedly as $r_{\max} \to \infty$ and the increasing rate depends on the path-loss exponent α.
- The latency-reduction ratio is a decreasing function of the target BER. Particularly, as BER $\to 0$, the ratio grows unboundedly, i.e., $\gamma \to \infty$, since no practical modulation scheme can achieve zero BER.
- For the digital scheme, the power-cutoff threshold has double effects on the latency-reduction ratio via affecting the receive SNR and channel-cutoff probability. On one hand, increasing g_{th} leads to a higher receive SNR of the digital scheme as reflected in (35), which reduces the latency-reduction ratio. On the other hand, a large g_{th} incurs a high channel-cutoff probability, that reduces the expected transmission rate of the digital scheme [see (37)] and thus increases the latency-reduction ratio.

A comprehensive comparison between analog and digital aggregation is shown in Table I.

VI. EXPERIMENTAL RESULTS

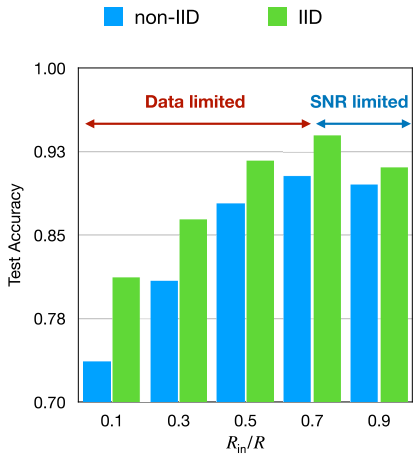
A. Experiment Settings

Consider a FEEL system with one edge server and $K = 200$ edge devices. The simulation parameters are set as follows unless specified otherwise. The cell radius is $R = 100$, the path loss exponent $\alpha = 3$, the number of sub-channels $M = 1000$, the average transmission power constraint per device $P_0 = 0.1(W)$, and the noise variance $N_0 = -80$ dBm. Besides, we set the power-cutoff threshold as $g_{\mathsf{th}} = 0.5$ and the number of iterations per communication round $\tau = 5$.

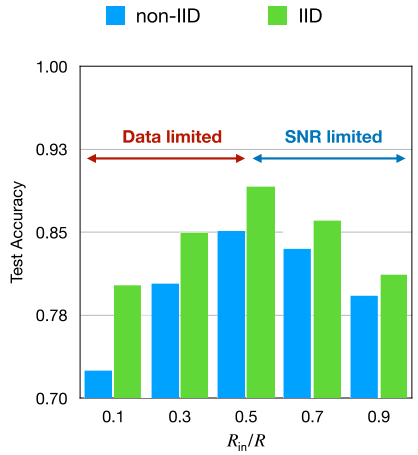
For exposition, we consider the learning task of handwritten-digit recognition using the well-known MNIST dataset that consists of 10 categories ranging from digit "0" to "9" and a total of 60000 labeled training data samples. To simulate the distributed mobile data, we consider two types of data partitions, i.e., the **IID** setting and **non-IID** one. For the former setting, we randomly partition the training samples into 200 equal shares, each of which is assigned to one particular device. While for the latter setting, we first sort the data by digit label, divide it into 400 shards of size 150, and assign each of 200 clients 2 shards. As illustrated in Fig. 6, the classifier model is implemented using a 6-layer convolutional neural network (CNN) that consists of two 5x5 convolution layers with ReLu activation (the first with 32 channels, the second with 64), each followed with 2x2 max pooling, a fully connected layer with 512 units and ReLu activation, and a final softmax output layer (582,026 parameters in total).

B. Tradeoff in User Scheduling

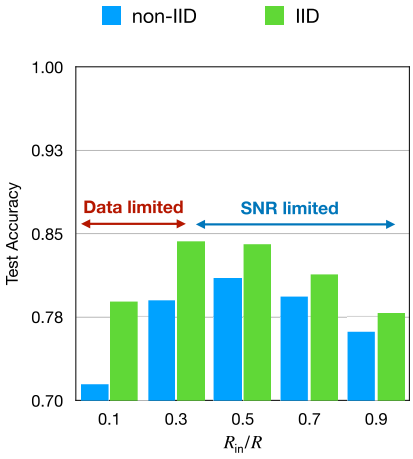
The tradeoff inherent in the user scheduling problem is first shown in Fig. 7. Consider the cell-interior scheduling in Scheme 1. The bar-figure shows the ultimate test accuracy of the learned model (after 1000 communication rounds) against the selection of the normalized distance threshold R_{in}/R. Three plots under varying values of the path-loss component α are provided where both the cases of IID and non-IID data-partition are experimented. It can be observed from all plots that, as the more devices included in the aggregation by increasing R_{in}, the test accuracy first increases then decreases



(a) $\alpha = 2.5$

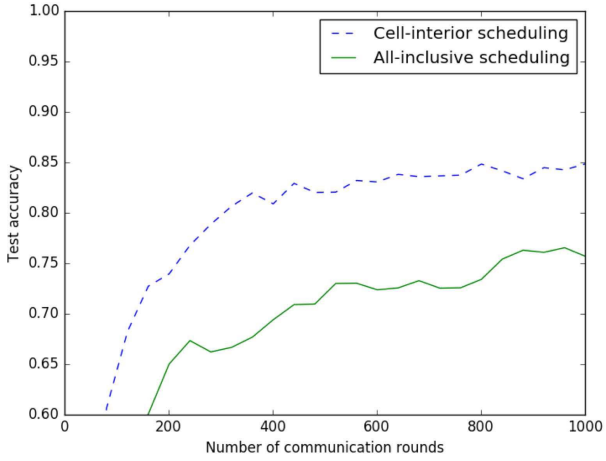


(b) $\alpha = 3$

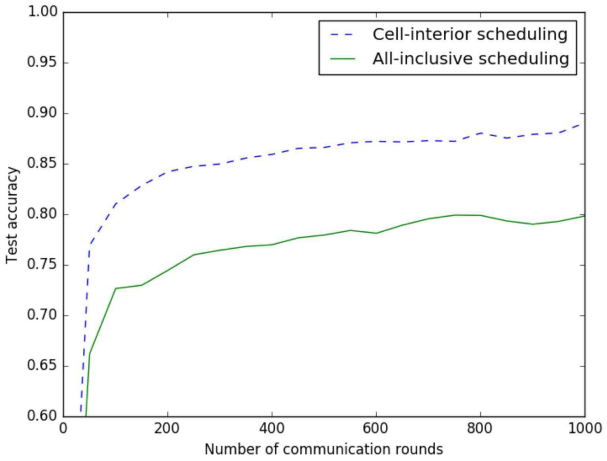


(c) $\alpha = 3.5$

Fig. 7. Test accuracy versus distance threshold in cell-interior scheduling.



(a) non-IID data distribution



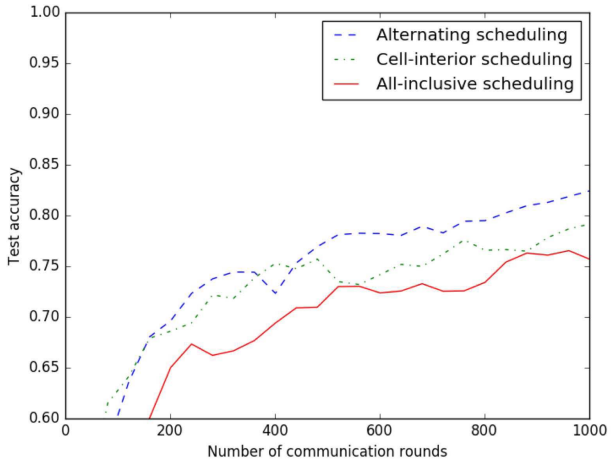
(b) IID data distribution

Fig. 8. Performance comparison between different scheduling schemes in the high-mobility scenario.

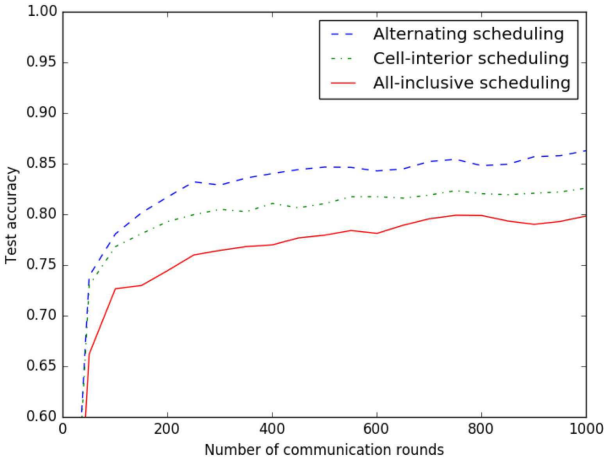
after a certain point, passing through a data-limited regime towards a SNR-limited regime. The phenomenon verifies the existence of the said reliability-quantity tradeoff in user scheduling. In addition, as the path-loss exponent increases, the learning performance is found to be more suffered from SNR-limited than data-limited, suggesting a decreasing choice of R_{in} to reduce the SNR penalty due to the scheduling of cell-edge devices. Last, it is also noted that the non-IID setting is in general more data-hungry than the IID one, thereby preferring a higher value of R_{in} even when the path-loss exponent is high [see Fig. 7(c)]. The observations align with our previous discussions in Section IV-C.

C. Performance Comparison Between Different Scheduling Schemes

The performance of the cell-interior scheduling scheme is evaluated in Figs. 8 and 9, for high-mobility and low-mobility scenarios, respectively. The difference between the two scenarios is that, in the former setting, the devices lying within the cell-interior change rapidly over communication rounds, while in the latter case, the device-locations remain unchanged throughout the entire training process. For all curves, the distance threshold R_{in} is optimized numerically by a fine-grained grid search within the range of $[0.1R, 0.9R]$ for the best test accuracy. It is observed that the cell-interior scheduling outperforms the naive all-inclusive scheduling by a remarkable gap in the high-mobility scenario where the learning performance is SNR-limited. On the other hand, in the low-mobility scenario where the cell-interior scheduling suffers from data-deficiency, the proposed alternating scheduling

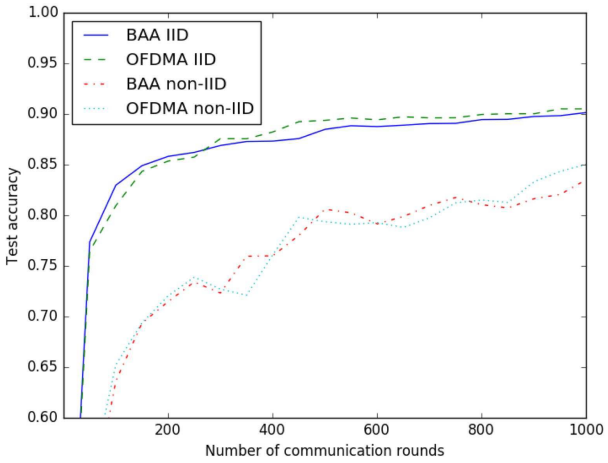


(a) non-IID data distribution

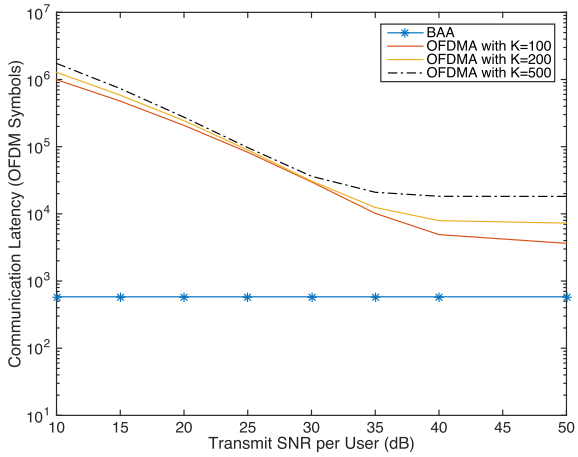


(b) IID data distribution

Fig. 9. Performance comparison between different scheduling schemes in the low-mobility scenario.



(a) Test accuracy



(b) Communication latency

Fig. 10. Performance comparison between BAA and OFDMA.

can enhance the learning performance further by occasionally exploiting the data in the cell-edge devices.

D. Performance Comparison: Analog Aggregation v.s. Digital Aggregation

The learning accuracy and communication latency of the BAA and the digital OFDMA are compared in Fig. 10 under the same transmit SNR per user and a fixed user scheduling scheme with $\frac{R_{\text{in}}}{R} = 0.5$. For the digital OFDMA, model-update parameters are quantized into bit sequence with 16-bit per parameter, and the adaptive MQAM modulation is used to maximize the spectrum efficiency under a target BER of 10^{-3}. As shown at Fig. 10(a), although BAA is expected to be more vulnerable to channel noise, it is interesting to see that the two schemes are comparable in learning accuracy (for both the IID and non-IID settings). Such accurate learning of BAA is partly due to the high expressiveness of the deep neural network which makes the learnt model robust against perturbation by channel noise. The result has a profound and refreshing implication that reliable communication may not be the primary concern in edge learning. Essentially, BAA exploits this relaxation on communication reliability to trade for a low communication latency as shown at Fig. 10(b). The latency gap between the two schemes is remarkable. Without compromising the learning accuracy, BAA can achieve a significant latency reduction ranging from 10x to 1000x. In general, the superiority in latency of BAA over OFDMA is more pronounced in the low SNR regime and dense-network scenarios.

VII. Concluding Remarks

In this paper, we have presented the framework of BAA for low-latency FEEL. The design exploits the wave-from superposition property of a multi-access channel for communication-efficient update aggregation. The significance of the work lies in the finding of two communication-and-learning tradeoffs, namely the *SNR-truncation tradeoff* resulting from the amplitude alignment required for aggregation, and the *reliabilty-quantity tradeoff* due to the scheduling of cell-interior devices for constraining path loss. The tradeoffs are fundamental for FEEL network with BAA and can provide useful guidelines for network planning and optimization. Moreover, we prove that the latency reduction of the proposed

BAA w.r.t. the traditional OFDMA scale almost linearly with the device population, justifying the low-latency property.

At a higher lever, the current work represents an initial but important steps towards the fusion of communication and computation/learning. It opens several directions for further investigation. One direction is to further enhance the aggregation performance of BAA by exploiting the clustering structure in device distribution for scheduling. Another interesting direction is to integrate the BAA design with the sparsity-induced update-compression techniques for further reducing the communication overhead. Last, robust BAA against the synchronization error is also an important topic to be addressed.

APPENDIX

A. Model-Parameter Normalization and De-Normalization

Since the statistics of the model parameters may change over communication rounds, to ensure proper normalization, the required model-parameter statistics (i.e, mean and variance) are estimated per communication round using the locally learnt model parameters at the round. The procedure for the edge server to access the information involves four main steps as described below.

- **Step 1 (Local Statistics Estimation at Devices)**: At each communication round, each device estimates the mean and variance of the locally learnt model parameter, denoted by $\bar{w}_k$ and σ_k^2 respectively, as follow:

$$\bar{w}_k = \frac{1}{q}\sum_{i=1}^{q} w_{k,i}, \quad \sigma_k^2 = \frac{1}{q}\sum_{i=1}^{q}(w_{k,i}-\bar{w}_k)^2. \quad (42)$$

  Then the locally estimated mean and variance will be transmitted to the edge server for global model-parameter statistics estimation by averaging.
- **Step 2 (Global Statistics Estimation at Edge Server)**: Upon receiving $\{\bar{w}_k\}$ and $\{\sigma_k^2\}$ at the edge server, the estimation of the model-parameter statistics can be refined by averaging over all the local estimates as follows:

$$\bar{w} = \frac{1}{K}\sum_{k=1}^{K} \bar{w}_k, \quad \sigma^2 = \frac{1}{K}\sum_{k=1}^{K} \sigma_k^2. \quad (43)$$

  Now the estimated $\bar{w}$ and σ^2 are the estimates of the mean and variance of the model parameters, which will be broadcast back to the edge devices and used for normalization.
- **Step 3 (Transmit Signal Normalization)** After receiving the normalization factors $\bar{w}$ and σ^2, each edge device perform transmit signal normalization as follows:

$$\tilde{w}_{k,i} = \frac{w_{k,i}-\bar{w}}{\sigma} \quad (44)$$

- **Step 4 (Receive Signal De-normalization)** At the edge server, the desired aggregation of the model parameters, $w_i = \sum_{k=1}^{K} w_{k,i}$, can be recovered from the aggregated receive signal, $\tilde{w}_i = \sum_{k=1}^{K} \tilde{w}_{k,i}$ by inverting the normalization as follows:

$$w_i = \sigma\tilde{w}_i + K\bar{w}. \quad (45)$$

B. Proof of Lemma 3

By definition, we can establish the following event equivalence.

$$\mathsf{Pr}(K_{\mathsf{in}}=k) = \mathsf{Pr}[k \text{ devices lie in the range of } R_{\mathsf{in}} \text{ while } (K-k) \text{ ones out of the range of } R_{\mathsf{in}}]. \quad (46)$$

Since the device-locations are i.i.d. distributed, the events defined on the right hand side in (46) follows a Binomial distribution with the success probability equal to $\mathsf{Pr}(r_k \le R_{\mathsf{in}})$, i.e., the probability that a device lie in the range of R_{in}:

$$\mathsf{Pr}(K_{\mathsf{in}}=k) = \binom{K}{k}\left[\mathsf{Pr}(r_k \le R_{\mathsf{in}})\right]^k \left[1-\mathsf{Pr}(r_k \le R_{\mathsf{in}})\right]^{K-k}. \quad (47)$$

Then, according to the uniform distribution presented in (10), we have

$$\mathsf{Pr}(r_k \le R_{\mathsf{in}}) = \int_0^{R_{\mathsf{in}}} \frac{2r}{R^2}dr = \frac{R_{\mathsf{in}}^2}{R^2}. \quad (48)$$

Thereby, by substituting (48) into (48), the desired result is obtained.

C. Proof of Lemma 5

By using Lemma 4 and (15), the expected receive SNR of all-inclusive scheme can be computed by

$$\mathsf{E}(\rho_0) = \int_0^R \frac{P_0}{Mx^{\alpha}\mathsf{Ei}(g_{\mathsf{th}})} f_{r_{\max}}(x)dx \quad (49)$$

$$= \frac{P_0}{M\mathsf{Ei}(g_{\mathsf{th}})}\frac{2K}{R^{2K}}\int_0^R x^{2K-\alpha-1}dx. \quad (50)$$

To ensure that the integral in (49) converges, it requires that $2K-\alpha-1 \ge 0$. The assumption always holds in practice as mentioned earlier. Under the assumption for convergence, by completing the integral, we can have the desired result.

D. Proof of Lemma 6

For the cell-interior scheduling, the expectation on the receive SNR is more challenging to derive, as the number of scheduled devices is now a random variable, adding an additional layer of randomness to the receive SNR besides the randomly distributed device-distance.

To overcome the challenge, we find it convenient to tackle the two-layer randomness sequentially using the trick of conditional expectation. Particularly, the expected aligned received power can be computed using the following formula.

$$\mathsf{E}(\rho_0) = \mathsf{E}[\mathsf{E}(\rho_0 \mid K_{\mathsf{in}}=k)], \quad (51)$$

where the first expectation is taken over the k-th furthest distance to the edge server conditioned on $K_{\mathsf{in}}=k$, while the second expectation is over the variable K_{in} whose PMF is given in Lemma 3. Then (51) can be explicitly written as

$$\mathsf{E}(\rho_0) = \sum_{k=0}^{K} \mathsf{E}(\rho_0 \mid K_{\mathsf{in}}=k)\mathsf{Pr}(K_{\mathsf{in}}=k). \quad (52)$$

For simplicity, we consider the typical case that $\alpha = 3$. Note that the first term in (51) is equal to zero, i.e., $\mathsf{E}(\rho_0 \mid$

$K_{\mathsf{in}} = 0) = 0$, and the second term is negligible when K is sufficiently large since $\mathsf{Pr}(K_{\mathsf{in}} = 1) \to 0$ and $\mathsf{E}(\rho_0 \mid K_{\mathsf{in}} = k)$ should be bounded.

Then remaining task is to compute $\mathsf{E}(\rho_0 \mid K_{\mathsf{in}} = k)$ for $k \geq 2$. Note that given $K_{\mathsf{in}} = k$, the k scheduled devices also follow i.i.d. uniform distribution over the cell-interior within the distance of R_{in}. As a result, by following simular steps in the proof of Lemma 5, for $k \geq 2$, one can easily derive that,

$$\mathsf{E}(\rho_0 \mid K_{\mathsf{in}} = k) = \frac{2k}{2k-\alpha} \frac{P_0}{M R_{\mathsf{in}}^{\alpha} \mathsf{Ei}(g_{\mathsf{th}})}, \tag{53}$$

Substituting (53) into (52), it follows that

$$\mathsf{E}(\rho_0) = \frac{P_0}{M R_{\mathsf{in}}^{\alpha} \mathsf{Ei}(g_{\mathsf{th}})} \times \underbrace{\sum_{k=2}^{K} \frac{2k}{2k-\alpha} \binom{K}{k} \left(\frac{R_{\mathsf{in}}^2}{R^2}\right)^k \left(1 - \frac{R_{\mathsf{in}}^2}{R^2}\right)^{K-k}}_{c(R_{\mathsf{in}})}, \tag{54}$$

which gives the derived result in (24).

Also note that the scaling factor $c(R_{\mathsf{in}})$ is essentially a weighted average for the term $\frac{2k}{2k-\alpha}$ from $k = 2$ to K. Given that $\alpha = 3$, and K is sufficiently large, we note that $\frac{2k}{2k-\alpha}$ monotonically ranges from 1 to 4. Since a weighted average for the values from a range will not exceed the range, it gives the conclusion that $1 \leq c(R_{\mathsf{in}}) \leq 4$, which completes the proof.

REFERENCES

[1] N. Poggi. (2017). *3 Key Internet of Things Trends to Keep Your Eye on in 2017*. [Online]. Available: https://preyproject.com/blog/en/3-key-internet-of-things-trends-to-keep-your-eye-on-in-2017/

[2] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Towards an intelligent edge: Wireless communication meets machine learning," 2018, *arXiv:1809.00343*. [Online]. Available: https://arxiv.org/abs/1809.00343

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[4] S. Wang *et al.*, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE Int. Conf. Comput. Comnun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 63–71.

[5] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *J. Amer. Stat. Assoc.*, vol. 114, no. 526, pp. 668–681, Feb. 2018. doi: 10.1080/01621459.2018.1429274.

[6] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Int. Statist. (AISTATS)*, Apr. 2017, pp. 1–10.

[7] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2017, *arXiv:1610.05492*. [Online]. Available: https://arxiv.org/abs/1610.05492

[8] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," 2017, *arXiv:1604.00981*. [Online]. Available: https://arxiv.org/abs/1604.00981

[9] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Sydney, NSW, Australia, Aug. 2017, pp. 3368–3376.

[10] M. Kamp, L. Adilova, J. Sicking, F. Hüger, P. Schlicht, T. Wirtz, and S. Wrobel, "Efficient decentralized deep learning by dynamic model averaging," 2018, *arXiv:1807.03210*. [Online]. Available: https://arxiv.org/abs/1807.03210

[11] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, Dec. 2018, pp. 5055–5065.

[12] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Copenhagen, Denmark, Sep. 2017, pp. 440–445.

[13] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, May 2018, pp. 1–14.

[14] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.

[15] M. Gastpar, "Uncoded transmission is exactly optimal for a simple Gaussian 'sensor' network," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 5247–5251, Nov. 2008.

[16] A. B. Wagner, S. Tavildar, and P. Viswanath, "Rate region of the quadratic Gaussian two-encoder source-coding problem," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 1938–1961, May 2008.

[17] R. Soundararajan and S. Vishwanath, "Communicating linear functions of correlated Gaussian sources over a MAC," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1853–1860, Mar. 2012.

[18] J. J. Xiao, S. Cui, Z. Q. Luo, and A. J. Goldsmith, "Linear coherent decentralized estimation," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 757–770, Feb. 2008.

[19] M. Goldenbaum and S. Stańczak, "On the channel estimation effort for analog computation over wireless multiple-access channels," *IEEE Wireless Commun. Lett.*, vol. 3, no. 3, pp. 261–264, Jun. 2014.

[20] C.-H. Wang, A. S. Leong, and S. Dey, "Distortion outage minimization and diversity order analysis for coherent multiaccess," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6144–6159, Dec. 2011.

[21] M. Goldenbaum, S. Stańczak, and H. Boche, "On achievable rates for analog computing real-valued functions over the wireless channel," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 4036–4041.

[22] O. Abari, H. Rahul, and D. Katabi, "Over-the-air function computation in sensor networks," *CoRR*, pp. 1–8, Dec. 2016. [Online]. Available: http://arxiv.org/abs/1612.02307

[23] M. Goldenbaum and S. Stanczak, "Robust analog function computation via wireless multiple-access channels," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3863–3877, Sep. 2013.

[24] O. Abari, H. Rahul, D. Katabi, and M. Pant, "AirShare: Distributed coherent transmission made seamless," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 1742–1750.

[25] G. Zhu and K. Huang, "MIMO over-the-air computation for high-mobility multimodal sensing," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6089–6103, Aug. 2018.

[26] X. Li, G. Zhu, Y. Gong, and K. Huang, "Wirelessly powered data aggregation for IoT via over-the-air function computation: Beamforming and power control," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3437–3452, Jul. 2019.

[27] D. Wen, G. Zhu, and K. Huang, "Reduced-dimension design of MIMO over-the-air computing for data aggregation in clustered IoT networks," *IEEE Trans. Wireless Commun.*, to be published. doi: 10.1109/TWC.2019.2934956.

[28] M. Chen, Y. Miao, Y. Hao, and K. Hwang, "Narrow band Internet of Things," *IEEE Access*, vol. 5, pp. 20557–20577, 2017.

[29] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning (extended version)," 2018, *arXiv:1812.11494*. [Online]. Available: http://arxiv.org/abs/1812.11494

[30] M. M. Amiri and D. Gunduz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air,"2019, *arXiv:1901.00844*. [Online]. Available: https://arxiv.org/abs/1901.00844

[31] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," 2018, *arXiv:1812.11750*. [Online]. Available: https://arxiv.org/abs/1812.11750

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[33] 3GPP Specifications. *Timing Advance (TA) in LTE*. Accessed: Sep. 2010. [Online]. Available: http://4g5gworld.com/blog/timing-advance-ta-lte

[34] G. Arunabha, J. Zhang, J. G. Andrews, and R. Muhamed, *Fundamentals of LTE* (The Prentice Hall Communications Engineering and Emerging Technologies Series). Upper Saddle River, NJ, USA: Prentice-Hall, 2010.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jun. +2014.
[36] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.
[37] X. Lin, R. K. Ganti, P. J. Fleming, and J. G. Andrews, "Towards understanding the fundamentals of mobility in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1686–1698, Apr. 2013.
[38] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, Sep. 2002.
[39] A. J. Goldsmith and S.-G. Chua, "Variable-rate variable-power MQAM for fading channels," *IEEE Trans. Commun.*, vol. 45, no. 10, pp. 1218–1230, Oct. 1997.

Guangxu Zhu (S'14) received the B.S. and M.S. degrees from Zhejiang University and the Ph.D. degree from The University of Hong Kong, all in electronic and electrical engineering. He is currently a Research Scientist with the Shenzhen Research Institute of Big Data. His research interests include edge intelligence, distributed machine learning, and 5G technologies, such as massive MIMO, mmWave communication, and wirelessly powered communication. He was a recipient of the Hong Kong Postgraduate Fellowship (HKPF) and the Best Paper Award from WCSP 2013.

Yong Wang received the B.S. degree from Nanjing University in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interests include the machine learning, deep learning, and natural language processing.

Kaibin Huang (S'05–M'08–SM'13) received the B.Eng. degree (Hons.) and the M.Eng. degree from the National University of Singapore and the Ph.D. degree from The University of Texas at Austin (UT Austin), all in electrical engineering. He is currently an Associate Professor with the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interests include wireless transmission of information and its applications in different areas, such as mobile edge computing, distributed machine learning, and 5G systems. He was a recipient the Best Article Award from the IEEE GLOBECOM 2006, the 2015 IEEE ComSoc Asia Pacific Outstanding Article Award, the 2019 IEEE ComSoc Best Tutorial Article Award, the Outstanding Teaching Award from Yonsei, the Motorola Partnerships in Research Grant, and the University Continuing Fellowship from UT Austin. He was an Editor of the IEEE WIRELESS COMMUNICATIONS LETTERS from 2011 to 2016 and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS series on *Green Communications and Networking* from 2015 to 2016. He is an Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.