

Federated Learning over Wireless Fading Channels

Mohammad Mohammadi Amiri and Deniz Gündüz

Abstract

We study federated machine learning at the wireless network edge, where limited power wireless devices, each with its own dataset, build a joint model with the help of a remote parameter server (PS). We consider a bandwidth-limited fading multiple access channel (MAC) from the wireless devices to the PS, and implement distributed stochastic gradient descent (DSGD) over-the-air. We first propose a digital DSGD (D-DSGD) scheme, in which one device is selected opportunistically for transmission at each iteration based on the channel conditions; the scheduled device quantizes its gradient estimate to a finite number of bits imposed by the channel condition, and transmits these bits to the PS in a reliable manner. Next, motivated by the additive nature of the wireless MAC, we propose a novel analog communication scheme, referred to as the *compressed analog* DSGD (CA-DSGD), where the devices first sparsify their gradient estimates while accumulating error from previous iterations, and project the resultant sparse vector into a low-dimensional vector. We also design a power allocation scheme to align the received gradient vectors at the PS in an efficient manner. Numerical results show that the proposed CA-DSGD algorithm converges much faster than the D-DSGD scheme and other schemes in the literature, while providing a significantly higher accuracy.

I. INTRODUCTION

As the dataset sizes and model complexity grow, distributed machine learning (ML) is becoming the only viable alternative to centralized ML, where all the dataset is gathered at a centralized server, and a joint model is trained. With the increasing amount of information collected through wireless edge devices, such centralized solutions are becoming increasingly costly, due to the limited power and bandwidth available, and less desirable due to privacy concerns. Federated

The authors are with the Information Processing and Communications Laboratory (IPC-Lab), Department of Electrical and Electronic Engineering, Imperial College London. (e-mail: m.mohammadi-amiri15@imperial.ac.uk; d.gunduz@imperial.ac.uk).

This work was supported in part by the European Research Council (ERC) Starting Grant BEACON (grant agreement no. 725731).

learning (FL) has been proposed as an alternative privacy-preserving distributed ML scheme, where each device participates in training using only locally available data with the help of a parameter server (PS) [1]. Devices exchange model parameters and their local updates with the PS, but the data never leaves the devices.

ML problems often involve the minimization of the empirical loss function

$$F(\boldsymbol{\theta}) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{u} \in \mathcal{B}} f(\boldsymbol{\theta}, \mathbf{u}) \quad (1)$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ denotes the model parameters to be optimized, \mathcal{B} is the training dataset with size $|\mathcal{B}|$ consisting of data samples and their labels, and $f(\cdot)$ is the loss function defined by the learning task. The minimization of $F(\boldsymbol{\theta})$ is typically carried out through iterative stochastic gradient descent (SGD) algorithm, in which the model parameter vector at iteration t , $\boldsymbol{\theta}_t$, is updated with a stochastic gradient

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \mathbf{g}(\boldsymbol{\theta}_t), \quad (2)$$

which satisfies $\mathbb{E}[\mathbf{g}(\boldsymbol{\theta}_t)] = \nabla F(\boldsymbol{\theta}_t)$, where η_t is the learning rate. SGD can easily be implemented across multiple devices, each of which has access to only a small fraction of the dataset. In distributed SGD (DSGD), at each iteration, device m computes a gradient vector based on the global parameter vector with respect to its local dataset, denoted by \mathcal{B}_m , and sends the result to the PS, which updates the global parameter vector according to

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \frac{1}{M} \sum_{m=1}^M \mathbf{g}_m(\boldsymbol{\theta}_t), \quad (3)$$

where M denotes the number of wireless devices, and $\mathbf{g}_m(\boldsymbol{\theta}_t) \triangleq \frac{1}{|\mathcal{B}_m|} \sum_{\mathbf{u} \in \mathcal{B}_m} \nabla f(\boldsymbol{\theta}_t, \mathbf{u})$, $m \in [M]$. In FL, each device participating in the training can also carry out model updates as in (3) locally, and share the overall difference with respect to the previous model parameters with the PS [1].

What distinguishes FL from conventional ML is the large number of devices that participate in the training, and the low-capacity and unreliable links that connect these devices to the PS. Therefore, there have been significant research efforts to reduce the communication requirements in FL [1]–[24]. However, these and follow-up studies consider orthogonal channels from the participating devices to the PS, and ignore the physical layer aspects of wireless connections, even though FL has been mainly motivated for mobile devices.

In this paper, we consider DSGD *over-the-air*; that is, we consider a wireless shared medium from the devices to the PS as it has been considered independently in the parallel works, [25]–[27]. To emphasize the limitations of wireless channels, we note that the dimension of some of the recent ML models, which also determines the size of the gradient estimates that must be transmitted to the PS at each iteration, can be extremely large, e.g., the 50-layer ResNet network has ~ 26 million weight parameters, while the VGGNet architecture has approximately 138 million parameters. On the other hand, available channel bandwidth is typically small due to the bandwidth and latency limitations; for example 1 LTE frame of 5MHz bandwidth and duration 10ms can carry only 6000 complex symbols. In principle, we can treat each iteration of the DSGD algorithm as a distributed over-the-air computation problem. FL over a static Gaussian MAC is studied in [25], where both a digital scheme, which separates computation and communication, and an analog over-the-air computation scheme are introduced. While the digital scheme exploits gradient quantization followed by independent channel coding at the participating wireless devices, the analog scheme exploits the additive nature of the wireless channel and gradient sparsification and random linear projection for dimensionality reduction. In [26] the authors consider a fading MAC, and also apply analog transmission, where each entry of a gradient vector at each of the devices is scheduled for transmission depending on the corresponding channel condition. A multi-antenna PS is considered in [27], where receive beamforming is used to maximize the number of devices scheduled for transmission at each iteration.

Here, we extend our previous work in [25], and study DSGD over a wireless fading MAC. We first consider the *separate* computation and communication approach, and propose a digital DSGD (D-DSGD) scheme, in which only a single device is opportunistically scheduled for transmission at each iteration of DSGD. The scheduled device quantizes its gradient estimate to a finite number of bits while accumulating the error from previous iterations (this will be clarified later), and employs a capacity achieving channel code to transmit the bits over the available bandwidth-limited channel to the PS. We then study analog transmission from the devices to the PS motivated by the signal-superposition property of the wireless MAC. At first, we extend the scheme in [26] by introducing error accumulation, which is shown to improve the performance numerically. We then propose a novel scheme, inspired by the random projection used in [25] for dimensionality reduction, which we will refer to as the *compressed analog DSGD* (CA-DSGD). With CA-DSGD, each device projects its gradient estimate to a low-dimensional

vector and transmits only the important gradient entries while accumulating the error. CA-DSGD scheme provides the flexibility of adjusting the dimension of the gradient estimate sent by each device, which is particularly important for bandwidth-limited wireless channels. A power allocation scheme is also designed, which aligns the vectors sent by different devices at the PS while satisfying the average power constraint. We implement these schemes for the MNIST classification task, where the training dataset is distributed randomly across wireless devices. Numerical results show that the proposed CA-DSGD scheme substantially improves both the training and test accuracies compared to the D-DSGD as well as the analog DSGD schemes under consideration with the same average power constraint and bandwidth resources. In addition to the significant benefits of the proposed CA-DSGD scheme we make the following observations:

- 1) The improvement of analog over-the-air computation compared to the D-DSGD scheme is particularly striking in the low power regime. This is mainly due to the “beamforming” effect of simultaneously transmitting highly correlated gradient estimates.
- 2) While the accuracy of the D-DSGD scheme increases drastically with the available average power, the performance of the analog schemes improve marginally. This highlights the energy efficiency of over-the-air computation, and makes it particularly attractive for FL across low-power IoT sensors.
- 3) Increasing the number of devices improves the accuracy for all the schemes even if the total dataset size and total power consumption remain the same. This “diversity gain” is much more limited for the analog schemes, and diminishes further as the training duration increases.
- 4) We observe that the performance of the CA-DSGD scheme improves if we reduce the bandwidth used at each iteration, and increase the number of DSGD iterations instead.

Notations: \mathbb{R} and \mathbb{C} represent the sets of real and complex values, respectively. For two vectors \mathbf{x} and \mathbf{y} with the same dimension, $\mathbf{x} \cdot \mathbf{y}$ returns their inner product. For a vector $\mathbf{z} \in \mathbb{C}^i$, $\text{Re}\{\mathbf{z}\} \in \mathbb{R}^i$ and $\text{Im}\{\mathbf{z}\} \in \mathbb{R}^i$ return the entry-wise real and imaginary components of \mathbf{z} , respectively. Also, $[\mathbf{v}, \mathbf{w}]$ represents the concatenation of two row vectors \mathbf{v} and \mathbf{w} . We denote a zero-mean normal distribution with variance σ^2 by $\mathcal{N}(0, \sigma^2)$, and $\mathcal{CN}(0, \sigma^2)$ represents a complex normal distribution with independent real and imaginary terms each distributed according to $\mathcal{N}(0, \sigma^2/2)$. For positive integer i , we let $[i] \triangleq \{1, \dots, i\}$. We denote the cardinality of set \mathcal{A} by $|\mathcal{A}|$, and l_2 norm of vector \mathbf{x} by $\|\mathbf{x}\|_2$. The imaginary unit is represented by $\sqrt{-1}$.

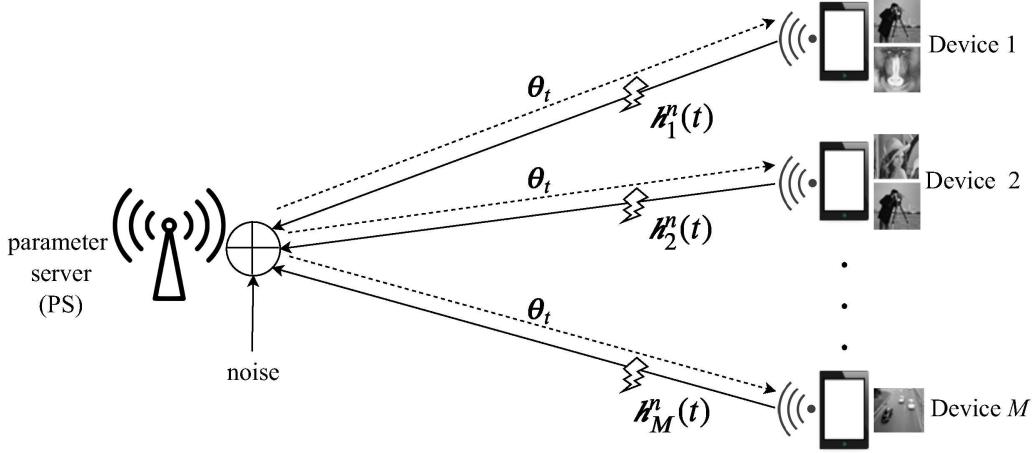


Fig. 1: Illustration of wireless FL architecture. The PS sends the updated parameter vector to all the wireless devices over an error-free ideal multi-cast channel, while the gradient estimates, computed by each device using only the available local dataset, are transmitted to the PS over the fading uplink channel.

II. SYSTEM MODEL

We consider FL across M wireless devices, each with its own local dataset, which employ DSGD with the help of a remote PS. We model the channel from the devices to the PS as a wireless fading MAC, and OFDM is employed for transmission. The system model is illustrated in Fig. 1. The parameter vector at iteration t is denoted by θ_t , and we assume that it is delivered from the PS to the devices over an error-free shared link. We denote the set of data samples available at device m by \mathcal{B}_m , and the stochastic gradient computed by device m with respect to local data samples by $\mathbf{g}_m(\theta_t) \in \mathbb{R}^d$, $m \in [M]$. At the t -th iteration of the DSGD algorithm in (3), the local gradient estimates of the devices are sent to the PS over a wireless fading MAC using s subchannels for a total of N time slots, where $s \leq d$ (in practice, we typically have $s \ll d$). We denote the length- s channel input vector transmitted by device m at the n -th time slot of the t -th iteration of the DSGD by $\mathbf{x}_m^n(t) = [x_{m,1}^n(t) \cdots x_{m,s}^n(t)]^T \in \mathbb{C}^s$. The i -th entry of channel output $\mathbf{y}^n(t) \in \mathbb{C}^s$ received by the PS at the n -th time slot of the t -th iteration, $n \in [N]$, is given by

$$y_i^n(t) = \sum_{m \in \mathcal{M}_i^n(t)} h_{m,i}^n(t) x_{m,i}^n(t) + z_i^n(t), \quad i \in [s], \quad (4)$$

where $\mathcal{M}_i^n(t) \subset [M]$, $h_{m,i}^n(t) \in \mathbb{C}$ is the i -th entry of the vector of channel gains $\mathbf{h}_m^n(t)$ from device m to the PS, and is assumed to be independent and identically distributed (i.i.d.) according

to $\mathcal{CN}(0, \sigma_m^2)$, e.g., Rayleigh fading, and $z_i^n(t) \in \mathbb{C}$ is the i -th entry of complex Gaussian noise vector $\mathbf{z}^n(t)$ and is i.i.d. according to $\mathcal{CN}(0, 1)$. The channel input vector of device m at the n -th time slot of iteration t , $n \in [N]$, is a function of the channel gains $\mathbf{h}_m^n(t)$, current parameter vector $\boldsymbol{\theta}_t$, the local dataset \mathcal{B}_m , and the current gradient estimate at device m , $\mathbf{g}_m(\boldsymbol{\theta}_t)$, $m \in [M]$. We assume that, at each time slot, the channel state information (CSI) is known by the devices and the PS. For a total of T iterations of the DSGD algorithm, the following total average transmit power constraint is imposed at device m :

$$\frac{1}{NT} \sum_{t=1}^T \sum_{n=1}^N \mathbb{E} [\|\mathbf{x}_m^n(t)\|_2^2] \leq \bar{P}, \quad \forall m \in [M], \quad (5)$$

where the expectation is taken over the randomness of the channel gains.

The goal is to recover $\frac{1}{M} \sum_{m=1}^M \mathbf{g}_m(\boldsymbol{\theta}_t)$ at the PS, which then updates the model parameter as in (3) after N time slots. However, due to the pre-processing performed at each device and the distortion caused by the wireless channel, the PS uses a noisy estimate to update the model parameter. Having defined $\mathbf{y}(t) \triangleq [\mathbf{y}^1(t)^T \cdots \mathbf{y}^N(t)^T]^T$, we have $\boldsymbol{\theta}_{t+1} = \phi(\boldsymbol{\theta}_t, \mathbf{y}(t))$ for some update function $\phi : \mathbb{R}^d \times \mathbb{C}^{Ns} \rightarrow \mathbb{R}^d$. The updated model parameter is then multicast to the devices by the PS through an error-free shared link, so the devices receive a consistent parameter vector for their computations in the next iteration.

We note that the goal of the devices is to transmit their local gradient estimates to the PS over the wireless channel, which is a joint source-channel coding problem. Moreover, the PS is not interested in the individual estimates of the devices, but in their average; hence, we have a distributed lossy computation problem. We will consider both a digital approach separating computations and communication and an analog transmission approach, where the gradients are transmitted simultaneously over the wireless MAC in an uncoded fashion.

III. DIGITAL DSGD

We first consider DSGD with digital transmission of the gradient estimates by the devices over the wireless fading MAC, referred to as the digital DSGD (D-DSGS) scheme. For D-DSGD, we consider $N = 1$, i.e., the parameter vector is updated after each time slot, and drop the dependency on time slot parameter n .

The goal here is to schedule devices and employ power allocation across time slots such that devices can transmit to the PS their local gradient estimates as accurately as possible. A possible approach is to schedule all the devices at all the iterations; however, due to the interference among

the devices this will result in each device sending a very coarse description of its local gradient estimate, and it is observed in [25] that the performance degrades with the increasing number of devices. Instead here we will schedule the devices opportunistically according to their channel states.

In particular, with the knowledge of channel state information (CSI), at each iteration t , we select the device which has the largest value of $\sum_{i=1}^s |h_{m,i}|^2$, $m \in [M]$. Accordingly, the index of the transmitting device at iteration t is given by:

$$m^*(t) = \arg \max_{m \in [M]} \left\{ \sum_{i=1}^s |h_{m,i}|^2 \right\}. \quad (6)$$

At each iteration, we denote the probability of selecting device m by π_m , which is time invariant due to the statistics of the channel gains, and $\sum_{m=1}^M \pi_m = 1$. The power allocated to device m at the t -th iteration is given by $P_m(t)$, which, for an average transmit power \bar{P} , should satisfy

$$\frac{\pi_m}{T} \sum_{t=1}^T P_m(t) \leq \bar{P}, \quad \text{for } m \in [M]. \quad (7)$$

Next, we derive the capacity of the communication channel from device $m^*(t)$ to the PS at the t -th iteration of D-DSGD, which provides an upper bound on the number of bits device $m^*(t)$ can transmit. The i -th entry of the channel output at the t -th iteration, which is the result of transmission from device $m^*(t)$, is given by

$$y_i(t) = h_{m^*(t),i}(t) x_{m^*(t),i}(t) + z_i(t), \quad i \in [s], \quad (8)$$

which is equivalent to a wireless fast fading channel with a limited number of channel uses s where the CSI is known at the transmitter and the receiver. In the following, we provide an upper bound on the capacity of the communication channel given in (8) with finite channel uses s . We consider a scenario, referred to as the *infinite fixed channel* scenario, where the channel realization $h_{m^*(t),i}$, $\forall i \in [s]$, is repeated over infinitely many time iterations. The capacity per iteration of this communication channel provides an upper bound on that of given in (8). For a transmit power $P_{m^*(t)}(t)$, the capacity per iteration of the wireless channel with the infinite fixed channel scenario is the result of the following optimization problem [28, Section 5.4.6]:

$$\begin{aligned} & \max_{P_1, \dots, P_s} \sum_{i=1}^s \log_2 \left(1 + P_i |h_{m^*(t),i}(t)|^2 \right), \\ & \text{subject to } \sum_{i=1}^s P_i = P_{m^*(t)}(t). \end{aligned} \quad (9)$$

Optimization problem (9) is solved through waterfilling, and the optimal power allocation is given by

$$P_i^* = \max \left\{ \frac{1}{\zeta} - \frac{1}{|h_{m^*(t),i}(t)|^2}, 0 \right\}, \quad (10)$$

where ζ is determined such that $\sum_{i=1}^s P_i^* = P_{m^*(t)}(t)$. Having calculated P_1^*, \dots, P_s^* , the capacity of the wireless channel in (8) is given by

$$R(t) = \sum_{i=1}^s \log_2 \left(1 + P_i^* |h_{m^*(t),i}(t)|^2 \right), \quad (11)$$

which provides an upper bound on the capacity of the communication channel between device $m^*(t)$ and the PS.

Remark 1. *An alternative device selection criteria rather than the one given in (6) is to select the device resulting in the highest channel capacity. Here we do not employ this selection criteria due to the overhead introduced by the computational complexity at the PS.*

We adopt the D-DSGD scheme proposed in [25, Section III], which is an extension of the one proposed in [18], for digital transmission. With the D-DSGD scheme, gradient estimate $\mathbf{g}_m(\boldsymbol{\theta}_t)$, computed at device m , is added to the error accumulated from previous iterations, denoted by $\Delta_m(t-1)$, where we set $\Delta_m(0) = \mathbf{0}$, $m \in [M]$. The error compensated gradient vector $\mathbf{g}_m(\boldsymbol{\theta}_t) + \Delta_m(t-1)$ is sparsified by setting $d - q(t)$ entries to zero (refer to [25, Section III] for the details of the sparsification method). Let $\hat{\mathbf{g}}_m(\boldsymbol{\theta}_t)$ denote the sparsified vector at device m , $m \in [M]$. After sparsifying the error compensated gradient estimate at device m and computing $\hat{\mathbf{g}}_m(\boldsymbol{\theta}_t)$, $m \in [M]$, the error accumulation vector is updated as follows:

$$\Delta_m(t) = \begin{cases} \mathbf{g}_m(\boldsymbol{\theta}_t) + \Delta_m(t-1) - \hat{\mathbf{g}}_m(\boldsymbol{\theta}_t), & \text{if } m = m^*(t), \\ \mathbf{g}_m(\boldsymbol{\theta}_t), & \text{otherwise.} \end{cases} \quad (12)$$

For a sparsity level $q(t)$, the D-DSGD scheme requires transmission of a total number of bits [25, Equation (10)]

$$r(t) = \log_2 \left(\frac{d}{q(t)} \right) + 33. \quad (13)$$

We assume that device $m^*(t)$ employs a capacity achieving channel code, and we set the sparsity level $q(t)$ as the highest value satisfying $r(t) \leq R(t)$.

We will evaluate the performance of the D-DSGD scheme in Section V, and study in detail the impact of various system parameters, such as the average power constraint, number of devices, etc., on the performance. We will also compare the D-DSGD scheme with the analog transmission of local gradients, which we present next.

IV. ANALOG DSGD

Analog DSGD is motivated by the fact that the PS is only interested in the average of the gradient vectors, and the underlying wireless MAC can provide the sum of the gradients if they are sent without any coding. We first present a generalization of the over-the-air computation approach introduced in [26], referred to as the *entry-wise scheduled analog DSGD (ESA-DSGD)* scheme, and then extend the ESA-DSGD scheme by introducing error accumulation, and refer to the scheme as *error compensated ESA-DSGD (ECESA-DSGD)*. Finally, we propose our scheme, built upon our previous work [25], referred to as the *compressed analog DSGD (CA-DSGD)* scheme.

A. ESA-DSGD

With the ESA-DSGD scheme studied in [26], each device sends its gradient estimate entirely after applying power allocation, which is to satisfy the average power constraint. At the t -th iteration of the DSGD, device m , $m \in [M]$, transmits its local gradient estimate $\mathbf{g}_m(\boldsymbol{\theta}_t) \in \mathbb{R}^d$ over $N = \lceil d/2s \rceil$ time slots by utilizing both the real and imaginary components of the available s subchannels. We define, for $n \in [N]$, $m \in [M]$,

$$\mathbf{g}_{m,\text{re}}^n(\boldsymbol{\theta}_t) \triangleq [g_{m,2(n-1)s+1}(\boldsymbol{\theta}_t), \dots, g_{m,(2n-1)s}(\boldsymbol{\theta}_t)]^T, \quad (14a)$$

$$\mathbf{g}_{m,\text{im}}^n(\boldsymbol{\theta}_t) \triangleq [g_{m,(2n-1)s+1}(\boldsymbol{\theta}_t), \dots, g_{m,2ns}(\boldsymbol{\theta}_t)]^T, \quad (14b)$$

$$\mathbf{g}_m^n(\boldsymbol{\theta}_t) \triangleq \mathbf{g}_{m,\text{re}}^n(\boldsymbol{\theta}_t) + \sqrt{-1}\mathbf{g}_{m,\text{im}}^n(\boldsymbol{\theta}_t), \quad (14c)$$

where $g_{m,i}(\boldsymbol{\theta}_t)$ is the i -th entry of $\mathbf{g}_m(\boldsymbol{\theta}_t)$, and we zero-pad $\mathbf{g}_m(\boldsymbol{\theta}_t)$ to have dimension $2sN$. We note that, according to (14),

$$\mathbf{g}_m(\boldsymbol{\theta}_t) = [\mathbf{g}_{m,\text{re}}^1(\boldsymbol{\theta}_t), \mathbf{g}_{m,\text{im}}^1(\boldsymbol{\theta}_t), \dots, \mathbf{g}_{m,\text{re}}^N(\boldsymbol{\theta}_t), \mathbf{g}_{m,\text{im}}^N(\boldsymbol{\theta}_t)]^T, \quad (15)$$

where $N = \lceil d/2s \rceil$. At the n -th time slot of the t -th iteration, device m , $m \in [M]$, sends $\mathbf{x}_m^n(t) = \boldsymbol{\alpha}_m^{e,n}(t) \cdot \mathbf{g}_m^n(\boldsymbol{\theta}_t)$, where $\boldsymbol{\alpha}_m^{e,n}(t) \in \mathbb{C}^s$ is the power allocation vector, which is set to

satisfy the average transmit power constraint. Thus, after N time slots, each device sends its gradient estimate of dimension d entirely. The i -th entry of the power allocation vector $\alpha_m^{e,n}(t)$ is set as follows:

$$\alpha_{m,i}^{e,n}(t) = \begin{cases} \frac{\gamma^e(t)}{h_{m,i}^n(t)}, & \text{if } |h_{m,i}^n(t)|^2 \geq \lambda_m^{e,n}(t), \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

for some $\gamma^e(t), \lambda_m^{e,n}(t) \in \mathbb{R}$, where $\lambda_m^{e,n}(t)$ is chosen to satisfy the average power constraint. According to (16), each entry of a gradient vector is transmitted if its corresponding channel gain is over a threshold. The set of devices selected to transmit the i -th entry of the channel input vector at the n -th time slot is given by, $i \in [s]$, $n \in [N]$,

$$\mathcal{M}_i^n(t) = \left\{ m \in [M] : |h_{m,i}^n(t)|^2 \geq \lambda_m^{e,n}(t) \right\}. \quad (17)$$

By substituting $\mathbf{x}_m^n(\boldsymbol{\theta}_t)$ and $\alpha_m^{e,n}(t)$ into (4), it follows that, for $i \in [s]$, $n \in [N]$,

$$y_i^n(t) = \gamma^e(t) \sum_{m \in \mathcal{M}_i^n(t)} (g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t) + \sqrt{-1}g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)) + z_i^n(t). \quad (18)$$

The PS has perfect CSI, and hence, knows set $\mathcal{M}_i^n(t)$. Its goal is to recover $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m=1}^{\mathcal{M}_i^n(t)} g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t)$ and $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m=1}^{\mathcal{M}_i^n(t)} g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)$, which provide estimates for $\frac{1}{M} \sum_{m=1}^M g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t)$ and $\frac{1}{M} \sum_{m=1}^M g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)$, respectively. The PS estimates $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m=1}^{\mathcal{M}_i^n(t)} g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t)$, for $i \in [s]$, $n \in [N]$, using its noisy observation $y_i^n(t)$, given in (18), as

$$\hat{g}_{2(n-1)s+i}^e(\boldsymbol{\theta}_t) = \begin{cases} \frac{\text{Re}\{y_i^n(t)\}}{\gamma^e(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (19)$$

and estimates $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m=1}^{\mathcal{M}_i^n(t)} g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)$ through

$$\hat{g}_{(2n-1)s+i}^e(\boldsymbol{\theta}_t) = \begin{cases} \frac{\text{Im}\{y_i^n(t)\}}{\gamma^e(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

Estimated vector $\hat{\mathbf{g}}^e(\boldsymbol{\theta}_t) \triangleq [\hat{g}_1^e(\boldsymbol{\theta}_t) \cdots \hat{g}_d^e(\boldsymbol{\theta}_t)]^T$ is then used to update the parameter vector as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \hat{\mathbf{g}}^e(\boldsymbol{\theta}_t)$.

Remark 2. We remark here that the scheme in [26] imposes a stricter average power constraint

\bar{P} per iteration of the DSGD, i.e., at device m we should have

$$\frac{1}{N} \sum_{n=1}^N \mathbb{E} [\|\mathbf{x}_m^n(t)\|_2^2] \leq \bar{P}, \quad \forall m \in [M], \forall t. \quad (21)$$

For fairness in our comparisons we relax this power constraint, and impose the one in (5), which constrains the average power over all the iterations.

B. ECESA-DSGD

With the ESA-DSGD scheme, entries of the gradient vectors that are not sent due to the limited average transmit power are completely forgotten. The proposed ECESA-DSGD scheme modifies ESA-DSGD by incorporating error accumulation technique to retain the accuracy of local gradients.

We denote the error accumulation vector calculated by device m at the n -th time slot of the t -iteration by $\Delta_m^{v,n}(t) \in \mathbb{C}^s$, and set $\Delta_m^{v,n}(t) = \mathbf{0}$, $\forall n, t, m$. Similarly to the ESA-DSGD scheme, with ECESA-DSGD, each device sends its entire gradient estimate of dimension d through $N = \lceil d/2s \rceil$ time slots, where the gradient estimates at the devices are zero-padded to dimension $2sN$. After computing $\mathbf{g}_m(\boldsymbol{\theta}_t)$ and obtaining $\mathbf{g}_m^n(\boldsymbol{\theta}_t)$ according to (14), device m , $m \in [M]$, updates its gradient estimate with the accumulated error as $\mathbf{g}_m^{v,n}(\boldsymbol{\theta}_t) \triangleq \mathbf{g}_m^n(\boldsymbol{\theta}_t) + \Delta_m^{v,n}(t-1)$, for $n \in [N]$, and transmits vector $\mathbf{x}_m^n(t) = \boldsymbol{\alpha}_m^{v,n}(t) \cdot \mathbf{g}_m^{v,n}(\boldsymbol{\theta}_t)$, where $\boldsymbol{\alpha}_m^{v,n}(t) \in \mathbb{C}^s$ is the power allocation vector, whose i -th entry is determined as follows:

$$\alpha_{m,i}^{v,n}(t) = \begin{cases} \frac{\gamma^v(t)}{h_{m,i}^n(t)}, & \text{if } |h_{m,i}^n(t)|^2 \geq \lambda_m^{v,n}(t), \\ 0, & \text{otherwise,} \end{cases} \quad (22)$$

for some $\gamma^v(t), \lambda_m^{v,n}(t) \in \mathbb{R}$. device m , $m \in [M]$, then updates the i -th entry of vector $\Delta_m^{v,n}(t)$ as follows:

$$\begin{aligned} \Delta_{m,i}^{v,n}(t) &= (1 - \mathbb{1}(\alpha_{m,i}^{v,n}(t) \neq 0)) g_{m,i}^n(\boldsymbol{\theta}_t) \\ &= (1 - \mathbb{1}(\alpha_{m,i}^{v,n}(t) \neq 0)) (g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t) + \sqrt{-1} g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)), \end{aligned} \quad (23)$$

where $\mathbb{1}(\cdot)$ is the indicator function, and $g_{m,i}^n(\boldsymbol{\theta}_t)$ denotes the i -th entry of $\mathbf{g}_m^n(\boldsymbol{\theta}_t)$, for $i \in [s]$,

$n \in [N]$. Thus, the i -th entry of vector $\mathbf{g}_m^{v,n}(\boldsymbol{\theta}_t)$ is given by, for $i \in [s]$, $n \in [N]$,

$$g_{m,i}^{v,n}(\boldsymbol{\theta}_t) = g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t) + \sqrt{-1}g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t) \\ + (1 - \mathbb{1}(\alpha_{m,i}^{v,n}(t-1) \neq 0)) (g_{m,2(n-1)s+i}(\boldsymbol{\theta}_{t-1}) + \sqrt{-1}g_{m,(2n-1)s+i}(\boldsymbol{\theta}_{t-1})). \quad (24)$$

According to (23), each entry of the gradient vector $\mathbf{g}_m^{v,n}(\boldsymbol{\theta}_t)$ that is not transmitted due to the power allocation given in (22), is retained in the error accumulation vector $\boldsymbol{\Delta}_m^{v,n}(t)$ for possible transmission in the next iteration. From the power allocation in (22), it follows that, for $i \in [s]$, $n \in [N]$,

$$y_i^n(t) = \gamma^v(t) \sum_{m \in \mathcal{M}_i^n(t)} g_{m,i}^{v,n}(\boldsymbol{\theta}_t) + z_i^n(t), \quad (25)$$

where we have

$$\mathcal{M}_i^n(t) = \left\{ m \in [M] : |h_{m,i}^n(t)|^2 \geq \lambda_m^{v,n}(t) \right\}. \quad (26)$$

Having perfect CSI, the PS's goal is to recover $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m=1}^M g_{m,i}^{v,n}(\boldsymbol{\theta}_t)$, the real and imaginary terms of which provide estimates for $\frac{1}{M} \sum_{m=1}^M g_{m,2(n-1)s+i}^n(\boldsymbol{\theta}_t)$ and $\frac{1}{M} \sum_{m=1}^M g_{m,(2n-1)s+i}^n(\boldsymbol{\theta}_t)$, respectively, for $i \in [s]$, $n \in [N]$. The PS estimates $\frac{1}{M} \sum_{m=1}^M g_{m,2(n-1)s+i}^n(\boldsymbol{\theta}_t)$ as

$$\hat{g}_{2(n-1)s+i}^v(\boldsymbol{\theta}_t) = \begin{cases} \frac{\text{Re}\{y_i^n(t)\}}{\gamma^v(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ \hat{g}_{2(n-1)s+i}^v(\boldsymbol{\theta}_{t-1}), & \text{otherwise,} \end{cases} \quad (27)$$

and estimates $\frac{1}{M} \sum_{m=1}^M g_{m,(2n-1)s+i}^n(\boldsymbol{\theta}_t)$ through

$$\hat{g}_{(2n-1)s+i}^v(\boldsymbol{\theta}_t) = \begin{cases} \frac{\text{Im}\{y_i^n(t)\}}{\gamma^v(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ \hat{g}_{(2n-1)s+i}^v(\boldsymbol{\theta}_{t-1}), & \text{otherwise,} \end{cases} \quad (28)$$

for $i \in [s]$, $n \in [N]$. After $N = \lceil d/2s \rceil$ time slots, estimated vector $\hat{\mathbf{g}}^v(\boldsymbol{\theta}_t) \triangleq [\hat{g}_1^v, \dots, \hat{g}_d^v]^T$ is then used to update the parameter vector as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \hat{\mathbf{g}}^v(\boldsymbol{\theta}_t)$.

C. CA-DSGD

As opposed to ESA-DSGD and ECESA-DSGD, which aim to transmit all the gradient entries to the PS at each DSGD iteration, i.e., $N = \lceil d/2s \rceil$, the CA-DSGD scheme proposed here applies gradient sparsification with error accumulation followed by a linear projection of the gradients

Algorithm 1 CA-DSGD

```

1: Initialize  $\theta_1 = \mathbf{0}$  and  $\Delta_1^c(0) = \dots = \Delta_M^c(0) = \mathbf{0}$ 
2: for  $t = 1, \dots, T$  do
  • devices do:
3:   for  $m = 1, \dots, M$  in parallel do
4:     Compute  $\mathbf{g}_m(\theta_t)$  with respect to local dataset  $\mathcal{B}_m$ 
5:      $\mathbf{g}_m^{ec}(\theta_t) = \mathbf{g}_m(\theta_t) + \Delta_m^c(t-1)$ 
6:      $\mathbf{g}_m^{sp}(\theta_t) = \text{sparse}_k(\mathbf{g}_m^{ec}(\theta_t))$ 
7:      $\Delta_m^c(t) = \mathbf{g}_m^{ec}(\theta_t) - \mathbf{g}_m^{sp}(\theta_t)$ 
8:      $\tilde{\mathbf{g}}_m(\theta_t) = A\mathbf{g}_m^{sp}(\theta_t)$ 
9:     for  $n = 1, \dots, N$  do
10:       $\mathbf{x}_m^n(t) = \alpha_m^{c,n}(t) \cdot \tilde{\mathbf{g}}_m^n(\theta_t)$ 
11:    end for
12:  end for
  • PS does:
13:  if  $\hat{\mathbf{y}}(t) \neq \mathbf{0}$  then
14:     $\hat{\mathbf{g}}^c(\theta_t) = \text{AMP}_A(\hat{\mathbf{y}}(t))$ 
15:     $\theta_{t+1} = \theta_t - \eta_t \hat{\mathbf{g}}^c(\theta_t)$ 
16:  else
17:     $\theta_{t+1} = \theta_t$ 
18:  end if
19: end for

```

at each device. With the CA-DSGD scheme, each device reduces the dimension of its gradient estimate to length $\tilde{s} = 2sN$, which can then be transmitted through N time slots, for an arbitrary $N \in \lceil [d/2s] \rceil$. Thus, the CA-DSGD scheme can have any N value, where $N \in \lceil [d/2s] \rceil$, and it generalizes the ESA-DSGD and ECESA-DSGD schemes. Algorithm 1 presents the CA-DSGD scheme, which we explain below.

We describe the CA-DSGD scheme for an arbitrary number of time slots $N \in \lceil [d/2s] \rceil$ per iteration of DSGD, which is determined later. At each iteration of the DSGD the devices sparsify their gradient estimates as described below. In order to retain the accuracy of their local gradient estimates, devices employ *error accumulation* [7], where the accumulated error vector at device m until iteration t is denoted by $\Delta_m^c(t-1) \in \mathbb{R}^d$, where we set $\Delta_m^c(0) = \mathbf{0}$, $\forall m \in [M]$. After computing $\mathbf{g}_m(\theta_t)$, device m updates its estimate with the accumulated error as $\mathbf{g}_m^{ec}(\theta_t) \triangleq \mathbf{g}_m(\theta_t) + \Delta_m^c(t-1)$, $m \in [M]$, where $k \leq \tilde{s}$ is a design parameter. Next, the devices apply gradient sparsification: device m sets all but k elements with the highest magnitudes of vector $\mathbf{g}_m^{ec}(\theta_t)$ to zero, and obtains a sparse vector $\mathbf{g}_m^{sp}(\theta_t)$, $m \in [M]$. This k -level sparsification is represented by function sparse_k in Algorithm 1, i.e., $\mathbf{g}_m^{sp}(\theta_t) = \text{sparse}_k(\mathbf{g}_m^{ec}(\theta_t))$. device m ,

$m \in [M]$, then updates $\Delta_m^c(t)$ as $\Delta_m^c(t) = \mathbf{g}_m^{ec}(\boldsymbol{\theta}_t) - \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t)$. To transmit the sparse vectors over the limited-bandwidth channel, devices employ a random projection matrix, similarly to compressive sensing.

Assuming local data points with identical distribution across the devices, the local gradient estimates also follow an identical distribution (in particular, for sufficiently large $|\mathcal{B}_m|$, where the local gradient estimates are expected to be close to the true gradient); and hence, they are expected to have similar sparsity patterns. A pseudo-random matrix $\mathbf{A} \in \mathbb{R}^{\tilde{s} \times d}$, with each entry i.i.d. according to $\mathcal{N}(0, 1/\tilde{s})$, is generated and shared between the PS and the devices, where $\tilde{s} = 2sN$, for an arbitrary $N \in \lceil [d/2s] \rceil$. At each iteration t , device m computes $\tilde{\mathbf{g}}_m(\boldsymbol{\theta}_t) \triangleq \mathbf{A}\mathbf{g}_m^{sp}(\boldsymbol{\theta}_t) \in \mathbb{R}^{\tilde{s}}$, and aims to transmit it to the PS through $N = \tilde{s}/2s$ time slots. We define, for $n \in [N]$, $m \in [M]$,

$$\tilde{\mathbf{g}}_{m,\text{re}}^n(\boldsymbol{\theta}_t) \triangleq [\tilde{g}_{m,2(n-1)s+1}(\boldsymbol{\theta}_t), \dots, \tilde{g}_{m,(2n-1)s}(\boldsymbol{\theta}_t)]^T, \quad (29a)$$

$$\tilde{\mathbf{g}}_{m,\text{im}}^n(\boldsymbol{\theta}_t) \triangleq [\tilde{g}_{m,(2n-1)s+1}(\boldsymbol{\theta}_t), \dots, \tilde{g}_{m,2ns}(\boldsymbol{\theta}_t)]^T, \quad (29b)$$

$$\tilde{\mathbf{g}}_m^n(\boldsymbol{\theta}_t) \triangleq \tilde{\mathbf{g}}_{m,\text{re}}^n(\boldsymbol{\theta}_t) + \sqrt{-1}\tilde{\mathbf{g}}_{m,\text{im}}^n(\boldsymbol{\theta}_t), \quad (29c)$$

where $\tilde{g}_{m,i}(\boldsymbol{\theta}_t)$ is the i -th entry of $\tilde{\mathbf{g}}_m(\boldsymbol{\theta}_t)$, $i \in [\tilde{s}]$. At the n -th time slot of the t -th iteration of the DSGD, device m , $m \in [M]$, sends $\mathbf{x}_m^n(t) = \boldsymbol{\alpha}_m^{c,n}(t) \cdot \tilde{\mathbf{g}}_m^n(\boldsymbol{\theta}_t)$, where $\boldsymbol{\alpha}_m^{c,n}(t) \in \mathbb{C}^s$ is the power allocation vector. The i -th entry of the power allocation vector $\boldsymbol{\alpha}_m^{c,n}(t)$ is set as follows:

$$\alpha_{m,i}^{c,n}(t) = \begin{cases} \frac{\gamma^c(t)}{h_{m,i}^n(t)}, & \text{if } |h_{m,i}^n(t)|^2 \geq \lambda_m^{c,n}(t), \\ 0, & \text{otherwise,} \end{cases} \quad (30)$$

for some $\gamma^c(t), \lambda_m^{c,n}(t) \in \mathbb{R}$. The set of devices scheduled to transmit the i -th entry of the channel input vector at the n -th time slot is given by, $i \in [s]$, $n \in [N]$,

$$\mathcal{M}_i^n(t) = \left\{ m \in [M] : |h_{m,i}^n(t)|^2 \geq \lambda_m^{c,n}(t) \right\}. \quad (31)$$

By substituting $\mathbf{x}_m^n(\boldsymbol{\theta}_t)$ and $\boldsymbol{\alpha}_m^{c,n}(t)$ into (4), it follows that, for $i \in [s]$, $n \in [N]$

$$\begin{aligned} y_i^n(t) &= \gamma^c(t) \sum_{m \in \mathcal{M}_i^n(t)} (\tilde{g}_{m,2(n-1)s+i}(\boldsymbol{\theta}_t) + \sqrt{-1}\tilde{g}_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)) + z_i^n(t) \\ &= \gamma^c(t) \left(\mathbf{a}_{2(n-1)s+i}^T \sum_{m \in \mathcal{M}_i^n(t)} \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t) + \sqrt{-1}\mathbf{a}_{(2n-1)s+i}^T \sum_{m \in \mathcal{M}_i^n(t)} \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t) \right) + z_i^n(t), \end{aligned} \quad (32)$$

where \mathbf{a}_j^T denotes the j -th row of measurement matrix \mathbf{A} , and we note that $\tilde{\mathbf{g}}_{m,j}(\boldsymbol{\theta}_t) = \mathbf{a}_j^T \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t)$, $j \in [\tilde{s}]$. The PS wants to recover $\frac{1}{M} \sum_{m=1}^M \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t)$ from its noisy observations in (32). For this, using its knowledge of matrix \mathbf{A} and the CSI, PS employs the approximate message passing (AMP) algorithm [29]. The AMP algorithm is represented by $\text{AMP}_{\mathbf{A}}$ in Algorithm 1. The PS first obtains, for $i \in [s]$, $n \in [N]$,

$$\hat{y}_{2(n-1)s+i}(t) = \begin{cases} \frac{\text{Re}\{y_i^n(t)\}}{\gamma^c(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (33a)$$

$$\hat{y}_{(2n-1)s+i}(t) = \begin{cases} \frac{\text{Im}\{y_i^n(t)\}}{\gamma^c(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (33b)$$

and then estimates

$$\hat{\mathbf{g}}^c(\boldsymbol{\theta}_t) = \text{AMP}_{\mathbf{A}}(\hat{\mathbf{y}}(t)), \quad (34)$$

where we define $\hat{\mathbf{y}}(t) \triangleq [\hat{y}_1(t), \dots, \hat{y}_{\tilde{s}}(t)]^T$. If $\hat{\mathbf{y}}(t) \neq \mathbf{0}$, $\hat{\mathbf{g}}^c(\boldsymbol{\theta}_t)$ is used to update the parameter vector as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \hat{\mathbf{g}}^c(\boldsymbol{\theta}_t)$. On the other hand, if $\hat{\mathbf{y}}(t) = \mathbf{0}$, the previous parameter vector is simply used as the new one, i.e., $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$.

Remark 3. For $N = \lceil d/2s \rceil$, in which the entire gradient vectors are transmitted to the PS at each iteration, the CA-DSGD scheme reduces to the ECESA-DSGD scheme; that is, sparsification and projection to a vector of smaller dimension, given in lines 6 and 8 of Algorithm 1, respectively, are not performed at the devices.

Remark 4. We remark that k is a design parameter which can take different values limited to $k < \tilde{s}$. For relatively small k values, $\frac{1}{M} \sum_{m=1}^M \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t)$ can be recovered from $\frac{1}{M} \sum_{m=1}^M \tilde{\mathbf{g}}_m(\boldsymbol{\theta}_t)$ in a more reliable manner as we have more noisy measurements of the sparsified gradient estimate than the number of its non-zero elements; however, $\frac{1}{M} \sum_{m=1}^M \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t)$ provides a less accurate estimate of the actual average gradient $\frac{1}{M} \sum_{m=1}^M \mathbf{g}_m(\boldsymbol{\theta}_t)$ as we have removed more components from the gradient estimates. On the other hand, for a relatively high k value, $\frac{1}{M} \sum_{m=1}^M \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t)$ provides a better estimate for $\frac{1}{M} \sum_{m=1}^M \mathbf{g}_m(\boldsymbol{\theta}_t)$; however, it becomes harder to recover $\frac{1}{M} \sum_{m=1}^M \mathbf{g}_m^{sp}(\boldsymbol{\theta}_t)$ from $\frac{1}{M} \sum_{m=1}^M \tilde{\mathbf{g}}_m(\boldsymbol{\theta}_t)$ in a reliable manner.

Remark 5. The proposed CA-DSGD scheme does not require data to be independent across

the devices. We assume that the gradient estimates computed by different devices have similar sparsity patterns. The rationale behind this assumption is that the gradient estimates across the devices follow an identical distribution, which is the result of assuming identical distribution for the local datasets across the devices. It is worth noting that this assumption helps to speed up the computation, but this is neither a requirement nor a limitation for the CA-DSGD scheme.

Remark 6. With ESA-DSGD, each device transmits only the entries of its estimated gradient whose corresponding channel conditions are sufficiently good. Thus, the gradient vector is inherently sparsified, but only based on the channel gains, regardless of the importance of the gradient entries. Then the entire gradient vector is sent over the bandwidth-limited wireless MAC over orthogonal time periods. On the other hand, with CA-DSGD, each device sends only $k \leq \tilde{s}$ important gradient entries, where the magnitude of each entry is regarded as the importance metric, by projecting the sparse gradient vector to a low-dimensional vector of length $\tilde{s} \leq d$. We further highlight the error accumulation technique incorporated into ECESA-DSGD and CA-DSGD, whereas with ESA-DSGD, entries of the gradient vectors that are not sent are forgotten.

D. Average Transmit Power Analysis

Here we analyze the performance of the analog DSGD schemes under consideration in terms of the average transmit power. The average transmit power at device m , $m \in [M]$, of the ESA-DSGD scheme in time slot n , $n \in [\lceil d/2s \rceil]$, of iteration t based on (16), is given by

$$\bar{P}_m^{e,n}(t) \triangleq \mathbb{E} [\|\mathbf{x}_m^n(t)\|_2^2] = \sum_{i=1}^s \mathbb{E} \left[|\alpha_{m,i}^{e,n}(t)|^2 |g_{m,(n-1)s+i}(\boldsymbol{\theta}_t)|^2 \right]. \quad (35)$$

We highlight that the entries of the gradient vector $\mathbf{g}_m^n(\boldsymbol{\theta}_t)$ are independent of the channel gains $h_{m,i}^n(t)$, $\forall i, n, m$. Since the power allocation vector $\boldsymbol{\alpha}_m^{e,n}(t)$ is a function of $\mathbf{g}_m^n(\boldsymbol{\theta}_t)$, it follows that, for $n \in [\lceil d/2s \rceil]$, $m \in [M]$,

$$\bar{P}_m^{e,n}(t) = \sum_{i=1}^s |g_{m,(n-1)s+i}(\boldsymbol{\theta}_t)|^2 \mathbb{E} \left[|\alpha_{m,i}^{e,n}(t)|^2 \right]. \quad (36)$$

Note that $|h_{m,i}^n(t)|^2$ follows an exponential distribution with mean σ_m^2 , $\forall i, n, m$. Thus, we have

$$\mathbb{E} \left[|\alpha_{m,i}^{e,n}(t)|^2 \right] = \left(\frac{\gamma^e(t)}{\sigma_m} \right)^2 \mathbb{E}_1(\lambda_m^{e,n}(t)), \quad (37)$$

where $E_1(x) \triangleq \int_x^\infty \frac{e^{-\tau}}{\tau} d\tau$. It follows that, $m \in [M], n \in [\lceil d/2s \rceil]$,

$$\bar{P}_m^{e,n}(t) = \left(\frac{\gamma^e(t)}{\sigma_m} \right)^2 E_1(\lambda_m^{e,n}(t)) P_m^{e,n}(t), \quad (38)$$

where we define $P_m^{e,n}(t) \triangleq \|\mathbf{g}_m^n(\boldsymbol{\theta}_t)\|_2^2$. To satisfy the average power constraint, we set the values $\{\bar{P}_m^{e,n}(t), \forall m, n, t\}$, such that

$$\max_{m \in [M]} \left\{ \frac{1}{NT} \sum_{t=1}^T \sum_{n=1}^N \bar{P}_m^{e,n}(t) \right\} \leq \bar{P}, \quad (39)$$

where $N = \lceil d/2s \rceil$.

Since the power allocation vectors of ECESA-DSGD and CA-DSGD schemes, given in (22) and (30), respectively, are similar to that of the ESA-DSGD scheme, we follow a similar procedure to obtain the average transmit power of each device for both the ECESA-DSGD and CA-DSGD schemes. For ECESA-DSGD, we have, $m \in [M], n \in [\lceil d/2s \rceil]$,

$$\bar{P}_m^{v,n}(t) = \left(\frac{\gamma^v(t)}{\sigma_m} \right)^2 E_1(\lambda_m^{v,n}(t)) P_m^{v,n}(t), \quad (40)$$

where we define $P_m^{v,n}(t) \triangleq \|\mathbf{g}_m^{v,n}(\boldsymbol{\theta}_t)\|_2^2$. Then, the values of $\{\bar{P}_m^{v,n}(t), \forall m, n, t\}$ are set to satisfy

$$\max_{m \in [M]} \left\{ \frac{1}{NT} \sum_{t=1}^T \sum_{n=1}^N \bar{P}_m^{v,n}(t) \right\} \leq \bar{P}, \quad (41)$$

where $N = \lceil d/2s \rceil$. Similarly, for CA-DSGD, we have, $m \in [M], n \in [N]$,

$$\bar{P}_m^{c,n}(t) = \left(\frac{\gamma^c(t)}{\sigma_m} \right)^2 E_1(\lambda_m^{c,n}(t)) P_m^{c,n}(t), \quad (42)$$

where we define $P_m^{c,n}(t) \triangleq \|\tilde{\mathbf{g}}_m^n(\boldsymbol{\theta}_t)\|_2^2$. Then, the values of $\{\bar{P}_m^{c,n}(t), \forall m, n, t\}$ are set to satisfy

$$\max_{m \in [M]} \left\{ \frac{1}{NT} \sum_{t=1}^T \sum_{n=1}^N \bar{P}_m^{c,n}(t) \right\} \leq \bar{P}, \quad (43)$$

where $N \in \lceil d/2s \rceil$.

V. NUMERICAL EXPERIMENTS

Here we compare the performances of the presented wireless edge learning schemes for the task of image classification. We run experiments on the MNIST dataset [30] with 60000 training and 10000 test samples, and train a single layer neural network with $d = 7850$ parameters utilizing ADAM optimizer [31]. To model the data distribution across the devices, we assume

that B training data samples, selected at random, are assigned to each device at the beginning of training, in which case, with high probability, the datasets of any two devices will have non-empty intersection; that is, the local datasets are not independent across the devices. We highlight the fact that, rather than distributing the training dataset to the devices by assigning each of them a mutually exclusive distinct subset, we assign a random subset to each device allowing intersections. We believe the latter represents our scenario of edge learning across wireless nodes, e.g., mobile devices, vehicles, drones, or IoT sensors, which may share some of their data for various sensors, such as physical proximity, whereas the former would be more appropriate for a distributed learning scenario, where a master node with all the dataset employs a number of devices to speed up the learning process. We consider $\sigma_m^2 = 1, \forall m \in [M]$, which results in $\pi_m = 1/M$, each device is selected for transmission with equal probability in the D-DSGD scheme. Thus, according to (7), for D-DSGD the following average power is imposed at device m :

$$\frac{1}{T} \sum_{t=1}^T P_m(t) \leq M\bar{P}, \quad \forall m \in [M]. \quad (44)$$

Throughout the experiments, we consider $s = \lceil d/20 \rceil$ parallel subchannels, which results in $N = 10$ for ESA-DSGD and ECESA-DSGD; and for any \tilde{s} of the CA-DSGD scheme, we set the sparsity level to $k = \lfloor \tilde{s}/2.5 \rfloor$.

For a fair comparison between the analog DSGD schemes, we consider $\gamma^x(t) = \gamma, \forall x \in \{e, v, c\}$. We also set the threshold values $\lambda_m^{x,n}(t), \forall x \in \{e, v, c\}$, such that device m consumes the same amount of power with any of the analog DSGD schemes at each time slot. To clarify, for two different DSGD schemes with N_1 and N_2 number of time slots per iteration, respectively, the n_1 -th time slot of the t_1 -th iteration of the first scheme is equivalent to $((t_1 - 1)N_1 + n_1)$ -th total time slot, and therefore, n_2 -th time slot of the t_2 -th DSGD iteration of the second scheme, where

$$t_2 = \left\lfloor \frac{(t_1 - 1)N_1 + n_1}{N_2} \right\rfloor, \quad (45a)$$

$$n_2 = (t_1 - 1)N_1 + n_1 - N_2 t_2. \quad (45b)$$

Accordingly, for a design parameter \tilde{s} with the CA-DSGD scheme resulting in $N = \tilde{s}/2s \in$

$\lceil d/2s \rceil$, we consider, for $m \in [M]$, $n \in [N]$,

$$\bar{P}_m^{c,n}(t) = \bar{P}_m^{e,n'}(t') = \bar{P}_m^{v,n'}(t'), \quad \forall t, \quad (46)$$

where

$$t' = \left\lfloor \frac{(t-1)N + n}{\lceil d/2s \rceil} \right\rfloor, \quad (47a)$$

$$n' = (t-1)N + n - t' \lceil d/2s \rceil. \quad (47b)$$

We remind here that $\bar{P}_m^{e,n}(t)$, $\bar{P}_m^{v,n}(t)$ and $\bar{P}_m^{c,n}(t)$ are given in (38), (40) and (42), respectively, and from (46), it follows that

$$\mathbb{E}_1(\lambda_m^{c,n}(t))P_m^{c,n}(t) = \mathbb{E}_1(\lambda_m^{e,n'}(t'))P_m^{e,n'}(t') = \mathbb{E}_1(\lambda_m^{v,n'}(t'))P_m^{v,n'}(t'), \quad \forall m, n, t. \quad (48)$$

We remark that (48) guarantees the same average transmit power for all the analog DSGD schemes under consideration. Furthermore, to have a fair comparison between the analog and digital DSGD schemes, we set, for $n \in [N]$,

$$P_m((t-1)N + n) = \sum_{m=1}^M \bar{P}_m^{c,n}(t), \quad \forall t, \quad (49)$$

i.e., the power allocated to the selected device for digital transmission at each time slot is equal to the total power consumed by all the devices for analog transmission at the same time slot.

We note that, since

$$\frac{1}{NT} \sum_{t=1}^T \sum_{n=1}^N \bar{P}_m^{c,n}(t) \leq \bar{P}, \quad \forall m \in [M], \quad (50)$$

(49) satisfies the average power constraint for the D-DSGD scheme for a total of $T' = NT$ iterations, given as follows:

$$\frac{1}{T'} \sum_{t=1}^{T'} P_m(t) \leq M\bar{P}, \quad \forall m \in [M]. \quad (51)$$

The performance is measured as the accuracy with respect to the training dataset versus the normalized time Nt , and the final accuracy for the test samples, i.e., test accuracy, based on the parameter vector obtained after a total of $NT = 100$ time slots, which corresponds to 10 iterations for the ESA-DSGD and ECESA-DSGD schemes, and 100 iterations for the D-DSGD scheme, while the number of iterations for the CA-DSGD scheme depends on \tilde{s} .

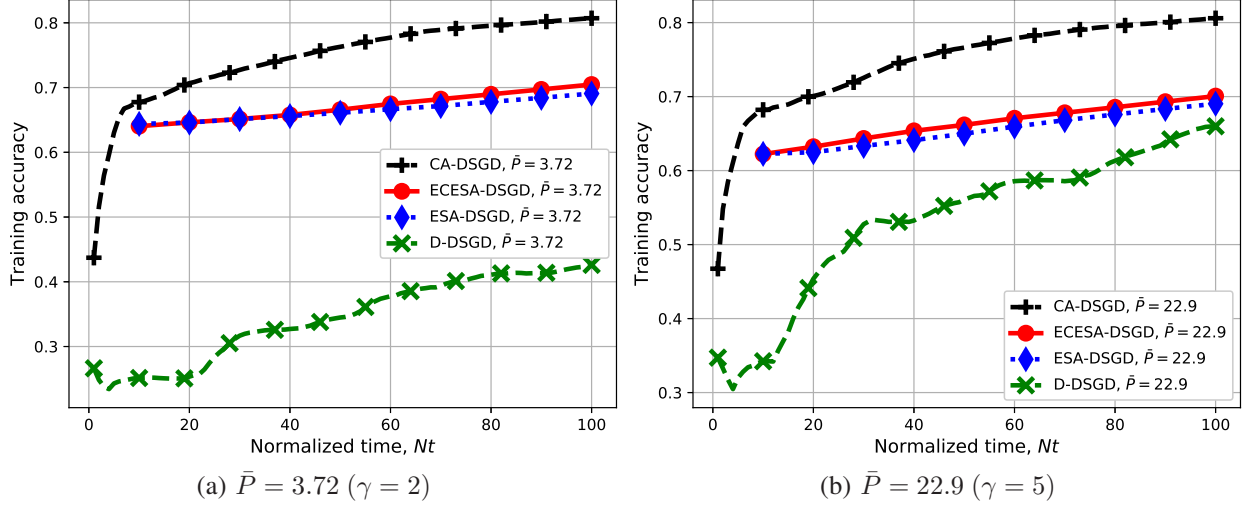


Fig. 2: Training accuracy for two different \bar{P} values, $\bar{P} \in \{3.72, 22.9\}$.

TABLE I: Final test accuracy comparison following a training period of $NT = 100$ time slots.

| | CA-DSGD | ECESA-DSGD | ESA-DSGD | D-DSGD |
|-----------------------------------|---------|------------|----------|--------|
| $\bar{P} = 3.72$ ($\gamma = 2$) | 0.806 | 0.704 | 0.689 | 0.42 |
| $\bar{P} = 22.9$ ($\gamma = 5$) | 0.806 | 0.704 | 0.689 | 0.65 |

In Figures 2 and 3, we compare the performance of the analog DSGD schemes with that of the D-DSGD scheme. We consider a relatively high average transmit power in order to guarantee that, at each iteration, the selected device with the D-DSGD scheme can transmit at least one information bit, i.e., $q(t) \geq 1, \forall t$.

In Fig. 2, we compare the performances of the CA-DSGD, ESA-DSGD and ECESA-DSGD schemes with D-DSGD for two different average transmit power values $\bar{P} = 3.72$ and $\bar{P} = 22.9$, averaged over $NT = 100$ time slots. These values are obtained by setting $\gamma = 2$ and $\gamma = 5$, respectively. We consider $\tilde{s} = 2s = d/10$, i.e., $N = 1$ for CA-DSGD, $M = 50$ and $B = 1200$. We set $\lambda_m^{e,n}(t) = 5 \times 10^{-5}, \forall m, n, t$, and we find the corresponding thresholds for CA-DSGD and ECESA-DSGD according to (48), and the transmit power at the selected device for the D-DSGD scheme from (49). The final test accuracies of different schemes under consideration are presented in Table I. As it can be seen, CA-DSGD performs significantly better than both ESA-DSGD and ECESA-DSGD. The main reasons for the degradation of the ESA-DSGD over CA-DSGD are i) scheduling gradient entries for transmission only based on the channel gains; ii) transmitting the entire gradient vectors of relatively huge dimensions (compared to the channel

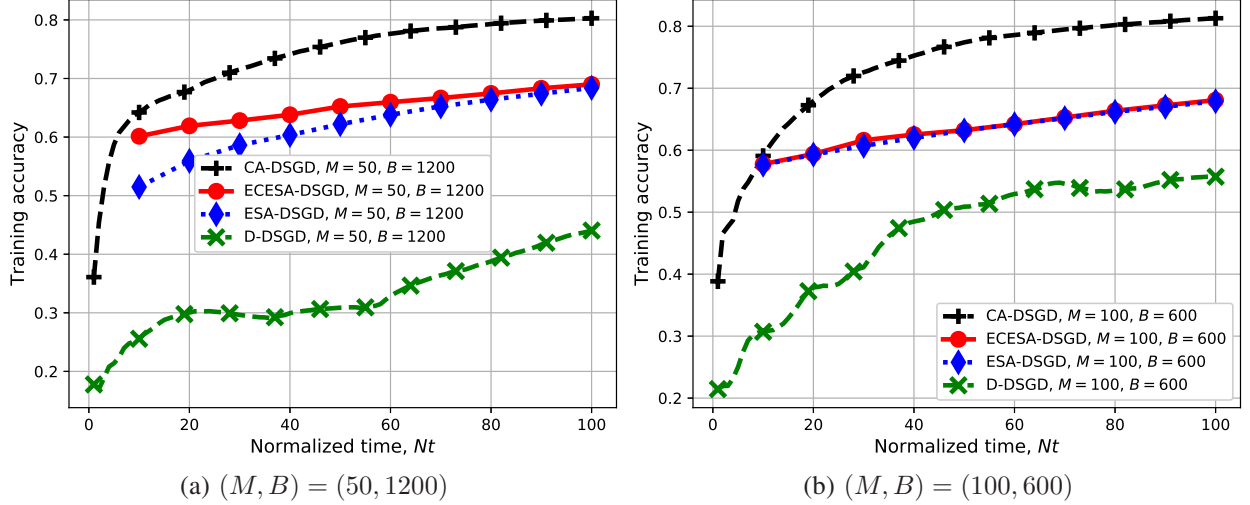


Fig. 3: Training accuracy for different M and B values with the same total training dataset size.

TABLE II: Final test accuracy comparison following a training period of $NT = 100$ time slots.

| | CA-DSGD | ECESA-DSGD | ESA-DSGD | D-DSGD |
|-----------------------|---------|------------|----------|--------|
| $(M, B) = (50, 1200)$ | 0.802 | 0.68 | 0.66 | 0.43 |
| $(M, B) = (100, 600)$ | 0.812 | 0.685 | 0.67 | 0.556 |

bandwidth); iii) ignoring the gradient entries which have not been transmitted due to the bad conditions of their corresponding channels. We note that ECESA-DSGD resolves the last issue by utilizing error accumulation technique, which provides some gains with respect to ESA-DSGD, but we observe that better scheduling of the gradient transmissions and more efficient utilization of the bandwidth through linear projection provide significant gains, increasing the accuracy from 0.7 to 0.8. We also observe that the analog schemes are superior to the digital D-DSGD scheme, particularly in the low-power regime. By comparing Figures 2a and 2b, it can be seen that the analog schemes are much less sensitive to a reduction in the average transmit power than the D-DSGD scheme. This is because the signal-superposition property of the wireless MAC aligns the transmission power of all the transmitting devices in the case of analog transmission, which provides a much stronger protection against noise collectively, as the system continues to operate in a relatively high effective signal-to-noise ratio (SNR) regime despite some reduction in the transmission power of individual devices.

In Fig. 3, we compare the performances of analog and digital schemes for different (M, B) pairs, $(M, B) \in \{(50, 1200), (100, 600)\}$, both having the same size of training data in total.

We set $\tilde{s} = 2s = d/10$, which results in $N = 1$ for CA-DSGD. We consider $\gamma = 2$, and $\lambda_m^{c,n}(t) = 0.1, \forall m, n, t$, and find the thresholds for other analog schemes from (48), and the transmit power of the scheduled device at the corresponding time slot for D-DSGD from (49). For a total number of time slots $NT = 100$, the average transmit power is $\bar{P} = 3.72$. We present the final test accuracies for the settings under consideration in Table II. It is evident that the analog schemes again perform significantly better than the D-DSGD scheme, and CA-DSGD improves upon ESA-DSGD and ECESA-DSGD noticeably in terms of the convergence speed and accuracy. As it can be seen, the performances of the analog schemes improve with M , since increasing M provides additional power introduced by each device and increases the robustness of the estimation against noise. D-DSGD also gains from increasing M , which is due to the additional power allocated to the selected device, as the devices are less frequently scheduled for transmission. We note that the superiority of the ECESA-DSGD over ESA-DSGD reduces with M , which shows that error accumulation is less effective for higher M values when MB is fixed. The reason is that, for higher M , the chance of receiving more estimates for each entry of the actual gradient vector is higher (each gradient entry is estimated more accurately at the PS), and it is less likely that no estimate of any gradient entry is received by the PS. Accordingly, the benefit of error accumulation is less significant for higher number of devices.

In the following, we only consider analog DSGD schemes, where we evaluate the performance in the low-power regime, which is the regime of interest for many applications requiring edge learning across low-power wireless devices, and we set $\gamma = 1$. Specifically, we consider the impact of changing \bar{P} , M , and \tilde{s} on the performance of the analog DSGD schemes for low-power regime. We compare the performances of the three analog schemes for two different average power values of 0.33 and 0.11 in Fig. 4. We consider $M = 100$ and $B = 600$. For the CA-DSGD scheme, we set $\tilde{s} = 2s = d/10$, in which case $N = 1$. We set $\lambda_m^{c,n}(t) = 0.5$ and $\lambda_m^{n,n}(t) = 1, \forall m, n, t$, which, respectively, result in average transmit powers 0.33 and 0.11 over $NT = 100$ time slots. The corresponding thresholds for CA-DSGD and ECESA-DSGD at each time slot are obtained according to (48). The final test accuracies are listed in Table III. It is obvious that the gain of gradient compression proposed by the CA-DSGD scheme is huge compared to ESA-DSGD and ECESA-DSGD. Similarly to Fig. 3, we observe that the performance of are not very sensitive to the variation in the average transmit power. Observe that the superiority of ECESA-DSGD over ESA-DSGD decreases with the average power, since, for higher power (equivalently, smaller threshold value) a more accurate gradient vector is transmitted by each

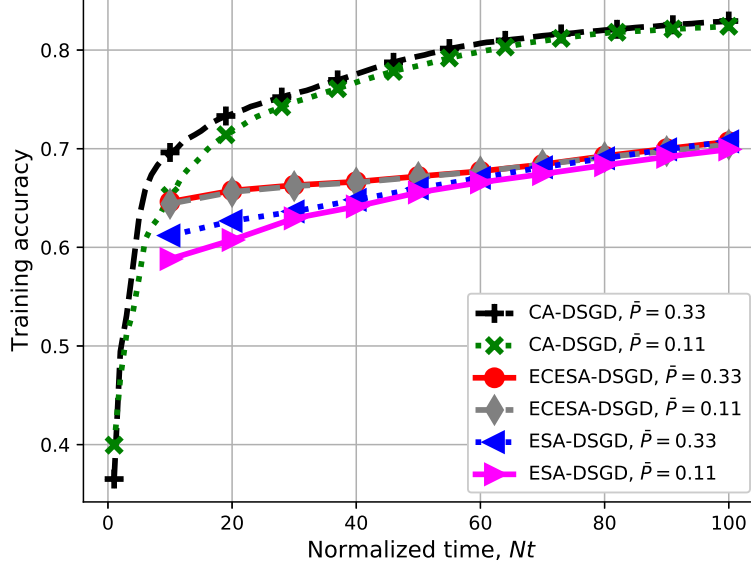


Fig. 4: Training accuracies of different analog schemes for two different \bar{P} values.

TABLE III: Final test accuracy comparison following a training period of $NT = 100$ time slots.

| | CA-DSGD | ECESA-DSGD | ESA-DSGD |
|---|---------|------------|----------|
| $\bar{P} = 0.33$ ($\lambda_m^{c,n}(t) = 0.5$) | 0.828 | 0.707 | 0.706 |
| $\bar{P} = 0.11$ ($\lambda_m^{c,n}(t) = 1$) | 0.824 | 0.703 | 0.698 |

device (less number of gradient entries are set to zero). Thus, error compensation provides less performance gain.

In Fig. 5, we compare the CA-DSGD, ESA-DSGD and ECESA-DSGD schemes for different (M, B) pairs, $(M, B) \in \{(50, 1200), (100, 600)\}$, where MB is fixed. For CA-DSGD, we set $\tilde{s} = 2s = d/10$ resulting in $N = 1$ time slot per iteration. We consider $\lambda_m^{c,n}(t) = 0.025, \forall m, n, t$, and find the thresholds for other analog schemes from (48). The resultant average power for a total of $NT = 100$ time slots is $\bar{P} = 0.33$. Table IV provides the final accuracies of different schemes based on the parameter vector obtained after $NT = 100$ time slots. In addition to the significant improvement of the CA-DSGD scheme over the other two analog schemes under consideration, it also gains more than other schemes from increasing M , which is due to the more efficient utilization of the gradient estimates computed by the devices. On the other hand, the gap between ECESA-DSGD and ESA-DSGD reduces with M , since by introducing more devices while MB is fixed, it is more likely to have a more accurate gradient entry at each gradient dimension even some gradient entries are ignored.

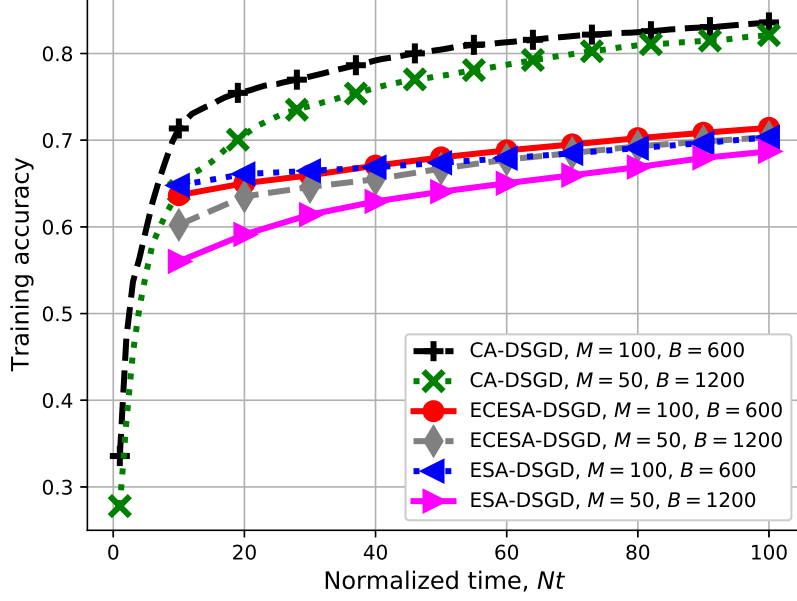


Fig. 5: Training accuracies of different analog schemes for different M and B values with the same total training dataset size.

TABLE IV: Final test accuracy comparison following a training period of $NT = 100$ time slots.

| | CA-DSGD | ECESA-DSGD | ESA-DSGD |
|-----------------------|---------|------------|----------|
| $(M, B) = (50, 1200)$ | 0.82 | 0.698 | 0.686 |
| $(M, B) = (100, 600)$ | 0.835 | 0.708 | 0.698 |

In Fig. 6 we investigate the impact of \tilde{s} on the performance of CA-DSGD. We consider $\tilde{s} \in \{2s, 4s\} = \{d/10, d/5\}$ for CA-DSGD, in which $\tilde{s} = 2s$ and $\tilde{s} = 4s$ are equivalent to $N = 1$ and $N = 2$, respectively. We have $M = 100$ and $B = 600$, and set $\lambda_m^{e,n}(t) = 0.05$, $\forall m, n, t$, and find the thresholds for other analog schemes from (48). The resultant average transmit power over $NT = 100$ time slots is $\bar{P} = 0.26$. In Table V, we summarize the final test accuracy of different schemes. We highlight the superiority of the CA-DSGD scheme for both \tilde{s} values under consideration over ESA-DSGD. The ECESA-DSGD scheme, which is equivalent to CA-DSGD for $\tilde{s} = d$, also outperforms ESA-DSGD slightly. However, as it can be seen, the performance of CA-DSGD degrades as \tilde{s} increases, which indicates that transmitting more sparse versions of the gradient estimates while using the available channel bandwidth for further iterations results in a higher accuracy. The flexibility in choosing the dimension of the transmitted gradient estimates makes the proposed CA-DSGD scheme particularly compelling for learning at the wireless edge under strict bandwidth limit.

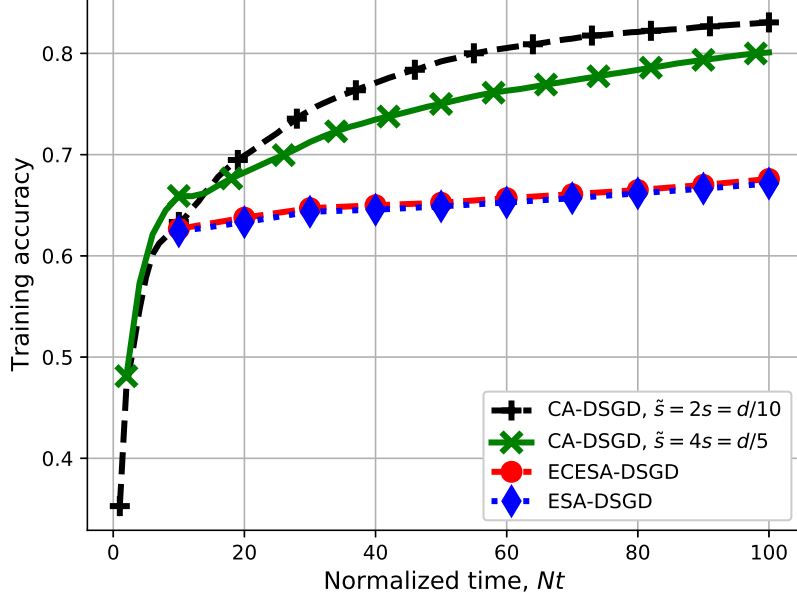


Fig. 6: Training accuracies of different analog schemes, and different \tilde{s} values for the CA-DSGD scheme.

TABLE V: Final test accuracy comparison following a training period of $NT = 100$ time slots.

| CA-DSGD $\tilde{s} = 2s$ | CA-DSGD $\tilde{s} = 4s$ | ECESA-DSGD | ESA-DSGD |
|-----------------------------|-----------------------------|------------|----------|
| 0.83 | 0.8 | 0.675 | 0.67 |

VI. CONCLUSIONS

We have studied distributed ML at the wireless edge, where M devices with limited transmit power and datasets communicate with the PS over a bandwidth-limited fading MAC to minimize a loss function by performing DSGD. The PS updates the parameter vector, and shares it with the devices over an error-free shared link. We first presented a digital approach that treats computation and communication separately. At each iteration of the proposed D-DSGD scheme, one device is selected depending on the channel states, and the selected device first quantizes its gradient estimate, and transmits the quantized bits to the PS using a capacity-achieving channel code. Then we studied the alternative analog transmission approach, considered recently in [25]–[27], which does not employ quantization or channel coding, and exploits the superposition property of the wireless MAC, rather than orthogonalizing the transmissions from different devices. First, we have extended the scheme in [26] by employing error accumulation. We have then proposed the CA-DSGD scheme, where each device employs gradient sparsification with error accumulation

followed by linear projection to reduce the typically very large parameter vector dimension to the limited channel bandwidth. We have also designed a power allocation scheme to align the received vectors at the PS while satisfying the average power constraints at the devices. The CA-DSGD scheme allows a much more efficient use of the limited channel bandwidth, and benefits from the “beamforming effect” thanks to the similarity in the patterns of the gradient estimates across the devices. Numerical results have shown significant improvements in the performance of CA-DSGD compared to D-DSGD and the state-of-the-art analog schemes.

REFERENCES

- [1] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” *arXiv:1610.05492v2 [cs.LG]*, Oct. 2017.
- [2] B. McMahan and D. Ramage, “Federated learning: Collaborative machine learning without centralized training data,” [online]. Available: <https://ai.googleblog.com/2017/04/federated-learning-collaborative.html>, Apr. 2017.
- [3] J. Konecny and P. Richtarik, “Randomized distributed mean estimation: Accuracy vs communication,” *arXiv:1611.07555 [cs.DC]*, Nov. 2016.
- [4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2017.
- [5] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning,” in *Proc. Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017.
- [6] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, “Deep learning with limited numerical precision,” in *ICML*, Jul. 2015.
- [7] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, “1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs,” in *INTERSPEECH*, Singapore, Sep. 2014, pp. 1058–1062.
- [8] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-efficient SGD via randomized quantization and encoding,” in *NIPS*, Long Beach, CA, Dec. 2017, pp. 1709–1720.
- [9] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, “TernGrad: Ternary gradients to reduce communication in distributed deep learning,” *arXiv:1705.07878v6 [cs.LG]*, Dec. 2017.
- [10] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, “DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients,” *arXiv:1606.06160v3 [cs.NE]*, Feb. 2018.
- [11] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, “ATOMO: Communication-efficient learning via atomic sparsification,” *arXiv:1806.04090v2 [stat.ML]*, Jun. 2018.
- [12] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, “signSGD: Compressed optimisation for non-convex problems,” *arXiv:1802.04434v3 [cs.LG]*, Aug. 2018.
- [13] B. Li, W. Wen, J. Mao, S. Li, Y. Chen, and H. Li, “Running sparse and low-precision neural network: When algorithm meets hardware,” in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jeju, South Korea, Jan. 2018.
- [14] N. Strom, “Scalable distributed DNN training using commodity gpu cloud computing,” in *INTERSPEECH*, 2015.
- [15] A. F. Aji and K. Heafield, “Sparse communication for distributed gradient descent,” *arXiv:1704.05021v2 [cs.CL]*, Jul. 2017.

- [16] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, “Deep gradient compression: Reducing the communication bandwidth for distributed training,” *arXiv:1712.01887v2 [cs.CV]*, Feb. 2018.
- [17] X. Sun, X. Ren, S. Ma, and H. Wang, “meProp: Sparsified back propagation for accelerated deep learning with reduced overfitting,” *arXiv:1706.06197v4 [cs.LG]*, Oct. 2017.
- [18] F. Sattler, S. Wiedemann, K. Muller, and W. Samek, “Sparse binary compression: Towards distributed deep learning with minimal communication,” *arXiv:1805.08768v1 [cs.LG]*, May 2018.
- [19] C. Renggli, D. Alistarh, T. Hoefer, and M. Aghagolzadeh, “SparCML: High-performance sparse communication for machine learning,” *arXiv:1802.08021v2 [cs.DC]*, Oct. 2018.
- [20] D. Alistarh, T. Hoefer, M. Johansson, S. Khirirat, N. Konstantinov, and C. Renggli, “The convergence of sparsified gradient methods,” *arXiv:1809.10505v1 [cs.LG]*, Sep. 2018.
- [21] Y. Tsuzuku, H. Imachi, and T. Akiba, “Variance-based gradient compression for efficient distributed deep learning,” *arXiv:1802.06058v2 [cs.LG]*, Feb. 2018.
- [22] S. U. Stich, “Local SGD converges fast and communicates little,” *arXiv:1805.09767v2 [math.OC]*, Jun. 2018.
- [23] T. Lin, S. U. Stich, and M. Jaggi, “Don’t use large mini-batches, use local SGD,” *arXiv:1808.07217v3 [cs.LG]*, Oct. 2018.
- [24] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, “LAG: Lazily aggregated gradient for communication-efficient distributed learning,” *arXiv:1805.09965 [stat.ML]*, May 2018.
- [25] M. M. Amiri and D. Gündüz, “Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air,” *arXiv:1901.00844 [cs.DC]*, Jan. 2019.
- [26] G. Zhu, Y. Wang, and K. Huang, “Low-latency broadband analog aggregation for federated edge learning,” *arXiv:1812.11494 [cs.IT]*, Jan. 2019.
- [27] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *arXiv:1812.11750 [cs.LG]*, Jan. 2019.
- [28] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge, UK: Cambridge University Press, 2005.
- [29] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.
- [30] Y. LeCun, C. Cortes, and C. Burges, “The MNIST database of handwritten digits,” <http://yann.lecun.com/exdb/mnist/>, 1998.
- [31] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980v9 [cs.LG]*, Jan. 2017.