Federated Learning over Wireless Networks: A Band-limited Coordinated Descent Approach

Junshan Zhang[1], Na Li[2], and Mehmet Dedeoglu[3]
[1,3]School of Electrical, Computer and Energy Engineering, Arizona State University
[2]School of Engineering and Applied Sciences, Harvard University
[1]Junshan.Zhang@asu.edu, [2]nali@seas.harvard.edu, [3]Mehmet.Dedeoglu@asu.edu

IEEE INFOCOM 2021 - IEEE Conference on Computer Communications | 978-1-6654-0325-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOM42981.2021.9488818

***Abstract*—We consider a many-to-one wireless architecture for federated learning at the network edge, where multiple edge devices collaboratively train a model using local data. The unreliable nature of wireless connectivity, together with constraints in computing resources at edge devices, dictates that the local updates at edge devices should be carefully crafted and compressed to match the wireless communication resources available and should work in concert with the receiver. Thus motivated, we propose SGD-based bandlimited coordinate descent algorithms for such settings. Specifically, for the wireless edge employing over-the-air computing, a common subset of k-coordinates of the gradient updates across edge devices are selected by the receiver in each iteration, and then transmitted simultaneously over k sub-carriers, each experiencing time-varying channel conditions. We characterize the impact of communication error and compression, in terms of the resulting gradient bias and mean squared error, on the convergence of the proposed algorithms. We then study learning-driven communication error minimization via joint optimization of power allocation and learning rates. Our findings reveal that optimal power allocation across different sub-carriers should take into account both the gradient values and channel conditions, thus generalizing the widely used water-filling policy. We also develop sub-optimal distributed solutions amenable to implementation.**

I. Introduction

In many edge networks, mobile and IoT devices collecting a huge amount of data are often connected to each other or a central node wirelessly. The unreliable nature of wireless connectivity, together with constraints in computing resources at edge devices, puts forth a significant challenge for the computation, communication and coordination required to learn an accurate model at the network edge. In this paper, we consider a many-to-one wireless architecture for distributed learning at the network edge, where the edge devices collaboratively train a machine learning model, using local data, in a distributed manner. This departs from conventional approaches which rely heavily on cloud computing to handle high complexity processing tasks, where one significant challenge is to meet the stringent low latency requirement. Further, due to privacy concerns, it is highly desirable to derive local learning model updates without sending data to the cloud. In such distributed learning scenarios, the communication between the edge devices and the server can become a bottleneck, in addition to the other challenges in achieving edge intelligence.

In this paper, we consider a wireless edge network with M devices and an edge server, where a high-dimensional machine learning model is trained using distributed learning. In such a setting with unreliable and rate-limited communications, local updates at sender devices should be carefully crafted and compressed to make full use of the wireless communication resources available and should work in concert with the receiver (edge server) so as to learn an accurate model. Notably, lossy wireless communications for edge intelligence presents unique challenges and opportunities [1], subject to bandwidth and power requirements, on top of the employed multiple access techniques. Since it often suffices to compute a function of the sum of the local updates for training the model, over-the-air computing is a favorable alternative to the standard multiple-access communications for edge learning. More specifically, over-the-air computation [2], [3] takes advantage of the superposition property of wireless multiple-access channel via simultaneous analog transmissions of the local messages, and then computes a function of the messages at the receiver, scaling signal-to-noise ratio (SNR) well with an increasing number of users. In a nutshell, when multiple edge devices collaboratively train a model, it is plausible to employ distributed learning over-the-air.

We seek to answer the following key questions: 1) What is the impact of the wireless communication bandwidth/power on the accuracy and convergence of the edge learning? 2) What coordinates in local gradient signals should be communicated by each edge device to the receiver? 3) How should the coordination be carried out so that multiple sender devices can work in concert with the receiver? 4) What is the optimal way for the receiver to process the received noisy gradient signals to be used for the stochastic gradient descent algorithm? 5) How should each sender device carry out power allocation across subcarriers to transmit its local updates? Intuitively, it is sensible to allocate more power to a coordinate with larger gradient value to speed up the convergence. Further, power allocation should also be channel-aware.

To answer the above questions, we consider an integrated learning and communication scheme where multiple edge devices send their local gradient updates over multi-carrier communications to the receiver for learning. Let K denote the number of subcarriers for communications, where K is determined by the wireless bandwidth. First, K dimensions of the gradient updates are determined (by the receiver) to be transmitted. Multiple methods can be used for selecting K coordinates, e.g., selecting the top-k (in absolute value) coordinates of the sum of the gradients or randomized uni-

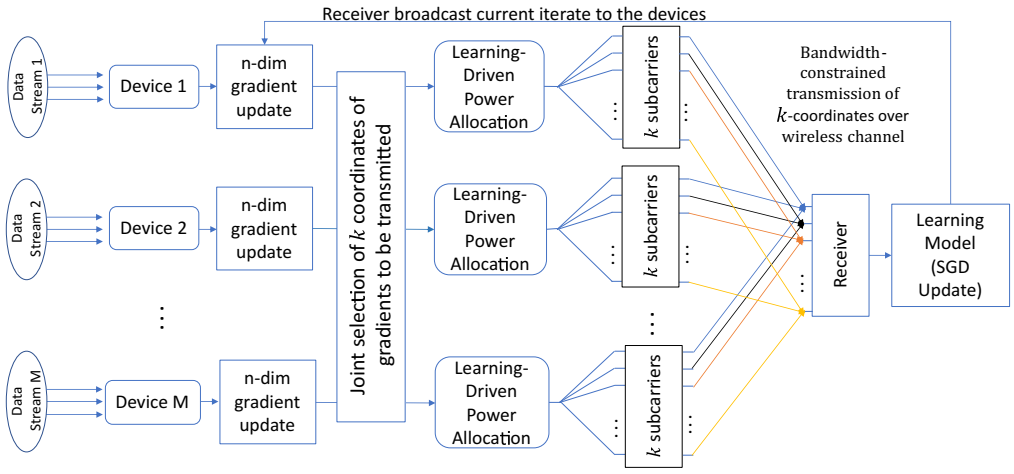

Fig. 1. A bandlimited coordinate descent algorithm for distributed learning over wireless multi-access channel

form selection. This paper will focus on randomly uniform selection (we elaborate further on this in Section V). During the subsequent communications, the gradient updates are transmitted only in the K-selected dimensions via over-the-air computing over K corresponding sub-carriers, each experiencing time-varying channel conditions and hence time-varying transmission errors. The devices are subject to power constraints, giving rise to a key question on how to allocate transmission power across dimension, at each edge device, based on the gradient update values and channel conditions. Thus, we explore joint optimization of the power allocation and the learning rate to obtain the best estimate of the gradient updates and minimize the impact of the communication error. We investigate a centralized solution to this problem as a benchmark, and then devise sub-optimal distributed solutions amenable to practical implementation. We note that we have also studied the impact of errors of synchronization across devices in this setting (we omit the details due to limited space).

The main contributions of this paper are summarized as follows:

- We take a holistic approach to study federated learning algorithms over wireless MAC channels, and the proposed bandlimited coordinated descent(BLCD) algorithm is built on innovative integration of computing in the air, multi-carrier communications, and wireless resource allocation.
- We characterize the impact of communication error and compression, in terms of its resulting gradient bias and mean squared error (MSE), on the convergence performance of the proposed algorithms. Specifically, when the communication error is unbiased, the BLCD algorithm would converge to a stationary point under very mild conditions on the loss function. In the case the bias in the communication error does exist, the iterates of the BLCD algorithm would return to a contraction region centered around a scaled version of the bias infinitely often.
- To minimize the impact of the communication error, we study joint optimization of power allocation at individual devices and learning rates at the receiver. Observe that since there exists tradeoffs between bias and variance, minimizing the MSE of the communication error does not necessarily amount to minimizing the bias therein. Our findings reveal that optimal power allocation across different sub-carriers should take into account both the gradient values and channel conditions, thus generalizing the widely used water-filling policy. We also develop sub-optimal distributed solutions amenable to implementation. In particular, due to the power constraints at individual devices, it is not always feasible to achieve unbiased estimators of the gradient signal across the coordinates. To address this complication, we develop a distributed algorithm which can drive the bias in the communication error to (close to) zero under given power constraints and then reduce the corresponding variance as much as possible.

II. Related Work

Communication-efficient SGD algorithms are of great interest to reduce latency caused by the transmission of the high dimensional gradient updates with minimal performance loss. Such algorithms in the ML literature are based on compression via quantization [4]–[7], sparsification [8]–[10] and federated learning [11] (or local updates [12]), where lossless communication is assumed to be provided. At the wireless edge, physical-layer design and communication loss should be taken into consideration for the adoption of the communication-efficient algorithms.

Power allocation for over-the-air computation is investigated for different scenarios in many other works [13]–[17] including MIMO, reduced dimensional MIMO, standard many to one channel and different channel models. In related works on ML over wireless channels, [18]–[25] consider over-the-air transmissions for training of the ML model. The authors in [21] propose sparsification of the updates with compressive sensing for further bandwidth reduction, and recovered sum of the compressed sparse gradients is used for the update. They also apply a similar framework for federated learning and fading

channels in [22]. [18] considers a broadband aggregation for federated learning with opportunistic scheduling based on the channel coefficients for a set of devices uniformly distributed over a ring. Lastly, [25] optimize the gradient descent based learning over multiple access fading channels. It is worth noting that the existing approaches for distributed learning in wireless networks do not fully account for the characteristics of lossy wireless channels. It is our hope that the proposed BLCD algorithms can lead to an innovative architecture of distributed edge learning over wireless networks that accounts for computation, power, spectrum constraints and packet losses.

III. Federated Learning over Wireless Multi-access Networks

A. Distributed Edge Learning Model

Consider an edge computing environment with M devices $\mathcal{M} = \{1, \ldots, M\}$ and an edge server. As illustrated in Figure 1, a high-dimensional ML model is trained at the server by using an SGD based algorithm, where stochastic gradients are calculated at the devices with the data points obtained by the devices and a (common) subset of the gradient updates are transmitted through different subcarriers via over-the-air.

The general edge learning problem is as follows:

$$\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{\xi_m}[l(w, \xi_m)], \tag{1}$$

in which $l(\cdot)$ is the loss function, and edge device m has access to inputs ξ_m. Such optimization is typically performed through empirical risk minimization iteratively. In the sequel, we let w_t denote the parameter value of the ML model at communication round t, and at round t edge device m uses its local data $\xi_{m,t}$ to compute a stochastic gradient $g_t^m(w_t) := \nabla l(w_t, \xi_{m,t})$. Define $g_t(w_t) = \frac{1}{M} \sum_{m=1}^{M} g_t^m(w_t)$. The standard vanilla SGD algorithms is given as

$$w_{t+1} = w_t - \gamma g_t(w_t) \tag{2}$$

with γ being the learning rate. Nevertheless, different updates can be employed for different SGD algorithms, and this study will focus on communication-error-aware SGD algorithms.

B. Bandlimited Coordinate Descent Algorithm

Due to the significant discrepancy between the wireless bandwidth constraint and the high-dimensional nature of the gradient signals, we propose a sparse variant of the SGD algorithm over wireless multiple-access channel, named as bandlimited coordinate descent (BLCD), in which at each iteration only a common set of K coordinates, $I(t) \subset \{1, \ldots, d\}$ (with $K \ll d$), of the gradients are selected to be transmitted through over-the-air computing for the gradient updates. The details of coordinate selection for the BLCD algorithm are relegated to Section VI. Worth noting is that due to the unreliable nature of wireless connectivity, the communication is assumed to be lossy, resulting in erroneous estimation of the updates at the receiver. Moreover, gradient correction is performed by keeping the difference between the update made at the receiver and the gradient value at the transmitter for the subsequent rounds, as gradient correction dramatically improves the convergence rate with sparse gradient updates [9].

Algorithm 1 Bandlimited Coordinate Descent Algorithm

Input: Sample batches $\xi_{m,t}$, model parameters w_1, initial learning rate γ, sparsification operator $C_t(.)$, $\forall m = 1, \ldots, M; \forall t = 1, \ldots, T$.
Initialize: $r_t^m := 0$.
for $t = 1 : T$ **do**
  for $m = 1 : M$ **do**
    $g_t^m(w_t) :=$ stochasticGradient$(f(w_t, \xi_{m,t}))$
    $u_t^m := \gamma g_t^m(w_t) + r_t^m$
    $r_{t+1}^m := u_t^m - C_t(u_t^m)$
    Compute power allocation coefficients $b_{km}^*, \forall k = 1, \ldots, K$.
    Transmit $\mathbf{b}^* \odot C_t(u_t^m)$
  end for
  Compute gradient estimator $\hat{G}_t(w_t)$
  $w_{t+1} := w_t - \hat{G}_t(w_t)$.
  Broadcast w_{t+1} back to all transmitters.
end for

For convenience, we first define the gradient sparsification operator as follows.

Definition 1. *$C_I : \mathbb{R}^d \to \mathbb{R}^d$ for a set $I \subseteq \{1, \ldots, d\}$ as follows: for every input $x \in \mathbb{R}^d$, $\big(C_I(x)\big)_j$ is $(x)_j$ for $j \in I$ and 0 otherwise.*

Since this operator C_I compress a d-dimensional vector to a k-dimension one, we will also refer this operator as compression operator in the rest of the paper.

With a bit abuse of notation, we let C_t denote $C_{I(t)}$ for convenience in the following. Following [26], we incorporate the sparsification error made in each iteration (by the compression operator C_t) into the next step to alleviate the possible gradient bias therein and improve the convergence possible. Specifically, as in [26], one plausible way for compression error correction is to update the gradient correction term as follows:

$$r_{t+1}^m = u_t^m - C_t(u_t^m), \tag{3}$$

$$u_t^m \triangleq \gamma g_t^m(w_t) + r_t^m \tag{4}$$

which r_{t+1}^m keeps the error in the sparsification operator that is in the memory of user m at around t, and u_t^m is the scaled gradient with correction at device m where the scaling factor γ is the learning rate in equation (2). (We refer readers to [26] for more insights of this error-feedback based compression SGD.) Due to the lossy nature of wireless communications, there would be communication errors and the gradient estimators at the receiver would be erroneous. In particular, the gradient estimator at the receiver in the BLCD can be written as

$$\hat{G}_t(w_t) = \frac{1}{M} \sum_{m=1}^{M} C_t\left(u_t^m\right) + \epsilon_t, \tag{5}$$

where ϵ_t denotes the random communication error in round t. In a nutshell, the bandlimited coordinate descent algorithm is outlined in Algorithm 1.

Recall that $g_t(w_t) = \frac{1}{M}\sum_{m=1}^{M} g_t^m(w_t)$ and define $r_t \triangleq \frac{1}{M}\sum_{m=1}^{M} r_t^m$. Thanks to the common sparsification operator across devices, the update in the SGD algorithm at communicatioon round t is given by

$$w_{t+1} = w_t - \big[C_t(\gamma g_t(w_t) + r_t) + \epsilon_t\big]. \tag{6}$$

To quantify the impact of the communication error, we use the corresponding communication-error free counterpart as the benchmark, defined as follows:

$$\hat{w}_{t+1} = w_t - C_t(\gamma g_t(w_t) + r_t). \tag{7}$$

It is clear that $w_{t+1} = \hat{w}_{t+1} - \epsilon_t$.

For convenience, we define $\tilde{w}_t \triangleq w_t - r_t$. It can be shown that $\tilde{w}_{t+1} = \tilde{w}_t - \gamma g_t(w_t) - \epsilon_t$. Intuitively, w_{t+1} in (6) is a noisy version of the iterate $\hat{w}_{t+1}$ in (7), which implies that $\tilde{w}_{t+1}$ is a noisy version of the compression-error correction of $\hat{w}_{t+1}$ in (7), where the "noisy perturbation" is incurred by the communication error.

C. BLCD Coordinate Transmissions over Multi-access Channel

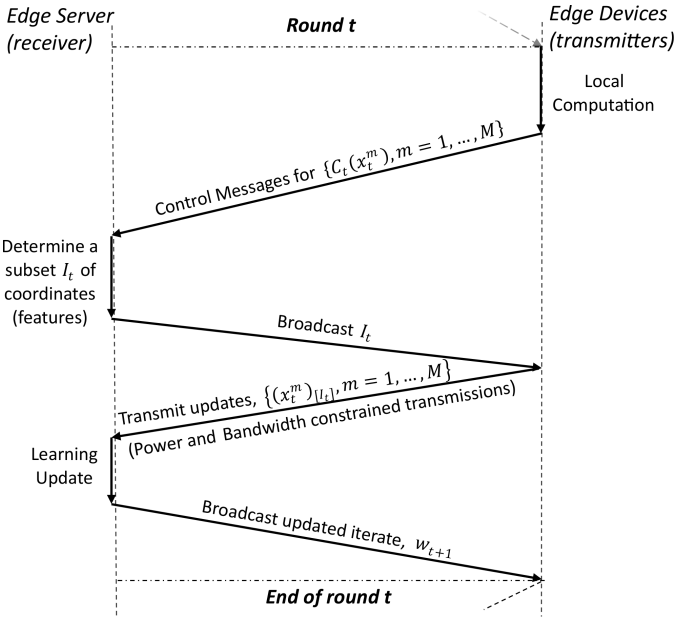

Fig. 2. A multi-access communication protocol for bandlimited coordinate selection and transmission.

A key step in the BLCD algorithm is to achieve coordinate synchronization of the transmissions among many edge devices. To this end, we introduce a receiver-driven low-complexity multi-access communication protocol, as illustrated in Fig. 2, with the function $C_t(x)$ denoting the compression of x at round t. Let $I(t)$ (of size K) denote the subset of coordinates chosen for transmission by the receiver at round t. Observe that the updates at the receiver are carried out only in the dimensions $I(t)$. Further, the edge receiver can broadcast its updated iterate to participant devices, over the reverse link. This task is quite simple, given the broadcast nature of wireless channels. In the transmissions, each coordinate of the gradient updates is mapped to a specific subcarrier and then transmitted through the wireless MAC channel, and the coordinates transmitted by different devices over the same subcarrier are received by the edge server in the form of an aggregate sum. It is worth noting that the above protocol is also applicable to the case when the SGD updates are carried out for multiple rounds at the devices.

When there are many edge devices, over-the-air computation can be used to take advantage of superposition property of wireless multiple-access channel via simultaneous analog transmissions of the local updates. More specifically, at round t, the received signal in subcarrier k is given by:

$$y_k(t) = \sum_{m=1}^{M} b_{km}(t) h_{km}(t) x_{km}(t) + n_k(t) \tag{8}$$

where $b_{km}(t)$ is a power scaling factor, $h_{km}(t)$ is the channel gain, and $x_{km}(t)$ is the message of user m through the subcarrier k, respectively, and $n_k(t) \sim \mathcal{N}(0, \sigma^2)$ is the channel noise.

To simplify notation, we omit (t) when it is clear from the context in the following. Specifically, the message $x_{km} = (C_t(u_t^m))_{l(k)}$, with a one-to-one mapping $l(k) = (I(t))_k$, which indicates the k-th element of $I(t)$, transmitted through the k-th subcarrier. The total power that a device can use in the transmission is limited in practical systems. Without loss of generality, we assume that there is a power constraint at each device, given by $\sum_{k=1}^{K} |b_{km} x_{km}|^2 \leq E_m, \ \forall m \in \{1, \ldots, M\}$. Note that b_{km} hinges heavily upon both $\boldsymbol{h}_m = [h_{1m}, \ldots, h_{Km}]^\top$ and $\boldsymbol{x}_m = [x_{1m}, \ldots, x_{Km}]^\top$, and a key next step is to optimize $b_{km}(\boldsymbol{h}_m, \boldsymbol{x}_m)$. In each round, each device optimizes its power allocation for transmitting the selected coordinates of its update signal over the K subcarriers, aiming to minimize the communication error so as to achieve a good estimation of $G_t(w_t)$ (or its scaled version) for the gradient update, where

$$G_t(w_t) \triangleq \frac{1}{M}\sum_{m=1}^{M} C_t(u_t^m).$$

From the learning perspective, based on $\{y_k\}_{k=1}^K$, it is of paramount importance for the receiver to get a good estimate of $G_t(w_t)$. Since $n_k(t)$ is Gaussian noise, the optimal estimator is in the form of

$$\big(\widehat{G}_t(w_t)\big)_k = \begin{cases} \alpha_{l(k)} y_{l(k)}, & k \in I(t) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\{\alpha_k\}_{k=1}^K$ are gradient estimator coefficients for subcarriers. It follows that the communication error (i.e., the gradient estimation error incurred by lossy communications) is given by

$$\epsilon_t = \widehat{G}_t(w_t) - G_t(w_t). \tag{10}$$

We note that $\{\alpha_k\}_{k=1}^K$ are intimately related to the learning rates for the K coordinates, scaling the learning rate to be $\{\gamma\alpha_k\}_{k=1}^K$. It is interesting to observe that the learning rates in the proposed BLCD algorithm are essentially different across the dimensions, due to the unreliable and dynamically changing channel conditions across different subcarriers.

IV. Impact of Communication Error and Compression on BLCD Algorithm

Recall that due to the common sparsification operator across devices, the update in the SGD algorithm at communication round t is given by

$$w_{t+1} = w_t - \big[C_t(\gamma g_t(w_t) + r_t) + \epsilon_t\big].$$

Needless to say, the compression operator C_t plays a critical role in sparse transmissions. In this study, we impose the following standard assumption on the compression rate of the operator.

Assumption 1. *For a set of the random compression operators $\{C_t\}_{t=1}^T$ and any $x \in \mathbb{R}^d$, it holds*

$$\mathbb{E}\left\|x - C_t(x)\right\|^2 \leq (1-\delta)\left\|x\right\|^2 \tag{11}$$

for some $\delta \in (0, 1]$.

We impose the following standard assumptions on the non-convex objective function $f(\cdot)$ and the corresponding stochastic gradients $g_t^m(w_t)$ computed with the data samples of device m in round t. (We assume that the data samples $\{\xi_{m,t}\}$ are i.i.d. across the devices and time.)

Assumption 2. *(Smoothness) A function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if for all $x, y \in \mathbb{R}^d$, it holds*

$$|f(y) - f(x) - \langle \nabla f(x), y - x\rangle| \leq \frac{L}{2}\left\|y - x\right\|^2. \tag{12}$$

Assumption 3. *For any $x \in \mathbb{R}^d$ and for any $m = 1, \ldots, M$, a stochastic gradient $g_t^m(x), \forall t$, satisfies*

$$\mathbb{E}[g_t^m(x)] = \nabla f(x),\ \ \mathbb{E}\left\|g_t^m(x)\right\|^2 \leq G^2 \tag{13}$$

where $G > 0$ is a constant.

It follows directly from [26] that $\mathbb{E}[\|r_t\|_2^2] \leq \frac{4(1-\delta)}{\delta^2}\gamma^2 G^2$. Recall that $\tilde{w}_{t+1} = \tilde{w}_t - \gamma g_t(w_t) - \epsilon_t$ and that $\tilde{w}_{t+1}$ can be viewed as a noisy version of the compression-error correction of $\hat{w}_{t+1}$ in (7), where the "noisy perturbation" is incurred by the communication error. For convenience, let $\mathbb{E}_t[\epsilon_t]$ denote the gradient bias incurred by the communication error and $\mathbb{E}_t[\|\epsilon_t\|_2^2]$ be the corresponding mean square error, where $\mathbb{E}_t$ is taken with respect to channel noise.

Let $\eta = \frac{L-1+2\gamma}{\gamma(2-\rho\gamma)}$ with $0 < \rho < 2$. Let f^* denote the globally minimum value of f. We have the following main result on the iterates in the BLCD algorithm.

Theorem 1. *Under Assumptions 1, 2 and 3, the iterates $\{w_t\}$ in the BLCD algorithm satisfies that*

$$\begin{aligned}
&\frac{1}{T+1}\sum_{t=0}^{T}\left(\|\nabla f(w_t)\|_2 - \eta\|\underbrace{\mathbb{E}_t[\epsilon_t]}_{bias}\|_2\right)^2 \\
&\leq \frac{1}{T+1}\sum_{t=0}^{T}\left[\frac{L\eta}{L-1+2\gamma}\underbrace{\mathbb{E}_t[\|\epsilon_t\|_2^2]}_{MSE} + \big(1+\eta^2\big)\|\underbrace{\mathbb{E}_t[\epsilon_t]}_{bias}\|_2^2\right] \\
&\quad + \frac{2}{T+1}\frac{f(w_0)-f^*}{\gamma(2-\rho\gamma)} + \left(\frac{L}{\rho}\frac{2(1-\delta)}{\delta^2} + \frac{1}{2}\right)\frac{2L\gamma G^2}{2-\rho\gamma}.
\end{aligned} \tag{15}$$

Proof. Due to the limited space, we outline only a few main steps for the proof. Recall that $\tilde{w}_t = w_t - r_t$. It can be shown that $\tilde{w}_{t+1} = \tilde{w}_t - \gamma g_t(w_t) - \epsilon_t$. As shown in (14), using the properties of the iterates in the BLCD algorithm and the smoonthness of the objective function f, we can establish an upper bound on $\mathbb{E}_t[f(\tilde{w}_{t+1})]$ in terms of $f(\tilde{w}_t)$ the corresponding gradient $\nabla f(w_t)$, and the gradient bias and MSE due to the communication error. Then, (15) can be obtained after some further algebraic manipulation. □

Remarks. Based on Theorem 1, we have a few observations in order.

- We first examine the four terms on the right hand side of (15): The first two terms capture the impact on the gradient by the time average of the bias in the communication error ϵ_t and that of the corresponding the mean square, denoted as MSE; the two items would go to zero if the bias and the MSE diminish; the third term is a scaled version of $f(w_0) - f^*$ and would go to zero as long as $\gamma = O(T^{-\beta})$ with $\beta < 1$; and the fourth term is proportional to γ and would go to zero when $\gamma \to 0$.
- If the right hand side of (15) diminishes as $T \to \infty$, the iterates in the BLCD algorithm would "converge" to a neighborhood around $\eta\|\mathbb{E}_t[\epsilon_t]\|_2$, which is a scaled version of the bias in the communication error. For convenience, let $\bar{\epsilon} = \limsup_t \|\mathbb{E}_t[\epsilon_t]\|_2$, and define a contraction region as follows:
$$A_\gamma = \{w_t : \|\nabla f(w_t)\|_2 \leq (\eta + \Delta)\bar{\epsilon}\}.$$
where $\Delta > 0$ is an arbitrarily small positive number. It then follows that the iterates in the BLCD algorithm would "converge" to a contraction region given by A_γ, in the sense that the iterates return to A_γ infinitely often. Note that f is assumed to be any nonconvex smooth function, and there can be many contraction regions, each corresponding to a stationary point.
- When the communication error is unbiased, the gradients would diminish to 0 and hence the BLCD algorithm would converge to a stationary point. In the case the bias in the communication error does exist, there exists intrinsic tradeoff between the size of the contraction region and $\eta\|\mathbb{E}_t[\epsilon_t]\|_2$. When the learning rate γ is small, the right hand side of (15) would small, but η can be large, and vice verse. It makes sense to choose a fixed learning rate that would make η small. In this way, the gradients in the BLCD algorithm would "concentrate" around a (small) scaled version of the bias.
- Finally, the impact of gradient sparsification is captured by δ. For instance, when (randomly) uniform selection is used, $\delta = \frac{k}{d}$. We will elaborate on this in Section VI.

Further, we have the following corollary.

Corollary 1. *Under Assumptions 1, 2, and 3, we have that if $\mathbb{E}_t[\epsilon_t] = 0$ and $\gamma = \frac{1}{\sqrt{T+1}}$, the BLCD algorithm converges to a stationary point and satisfies that*

$$\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla f(w_t)\|_2^2$$

$$
\begin{aligned}
\mathbb{E}_t[f(\tilde{w}_{t+1})] &\le f(\tilde{w}_t)+\langle \nabla f(\tilde{w}_t), \mathbb{E}_t[\tilde{w}_{t+1}-\tilde{w}_t]\rangle+\frac{L}{2}\mathbb{E}_t[\|\tilde{w}_{t+1}-\tilde{w}_t\|^2] \\
&= f(\tilde{w}_t)-\langle \nabla f(\tilde{w}_t), \gamma \mathbb{E}_t[g_t(w_t)]+\mathbb{E}_t[\epsilon_t]\rangle+\frac{L}{2}\mathbb{E}_t[\|\gamma g_t(w_t)\|^2]+\frac{L}{2}\mathbb{E}_t[\|\epsilon_t\|^2]+L\mathbb{E}_t[\langle \gamma g_t(w_t), \epsilon_t\rangle] \\
&= f(\tilde{w}_t)-\langle \nabla f(w_t), \gamma \mathbb{E}_t[g_t(w_t)]+\mathbb{E}_t[\epsilon_t]\rangle-\langle \nabla f(\tilde{w}_t)-\nabla f(w_t), \gamma \mathbb{E}_t[g_t(w_t)]+\mathbb{E}_t[\epsilon_t]\rangle+\frac{L}{2}\mathbb{E}_t[\|\epsilon_t\|_2^2]+L\mathbb{E}_t[\langle \gamma g_t(w_t), \epsilon_t\rangle]+\frac{L}{2}\mathbb{E}_t[\|\gamma g_t(w_t)\|^2 \\
&\le f(\tilde{w}_t)-\gamma\|\nabla f(w_t)\|_2^2-\langle \nabla f(w_t), \mathbb{E}_t[\epsilon_t]\rangle+\frac{\rho}{2}\|\gamma \nabla f(w_t)+\mathbb{E}_t[\epsilon_t]\|_2^2+\frac{L^2}{2\rho}\mathbb{E}_t[\|r_t\|_2^2]+\frac{L}{2}\mathbb{E}_t[\|\epsilon_t\|_2^2]+L\langle \nabla f(w_t), \mathbb{E}_t[\epsilon_t]\rangle+\frac{L\gamma^2}{2}\mathbb{E}_t\|g_t(w_t)\|_2^2 \\
&\le f(\tilde{w}_t)-\gamma\|\nabla f(w_t)\|_2^2+(L-1)\|\nabla f(w_t)\|\|\mathbb{E}_t[\epsilon_t]\|+\frac{\rho}{2}\left(\gamma^2\|\nabla f(w_t)\|_2^2+\|\mathbb{E}_t[\epsilon_t]\|_2^2+2\gamma\langle \nabla f(w_t), \mathbb{E}_t[\epsilon_t]\rangle\right)+\frac{L^2}{2\rho}\mathbb{E}_t[\|r_t\|_2^2]+\frac{L}{2}\mathbb{E}_t[\|\epsilon_t\|_2^2]+\frac{L\gamma^2}{2}G^2 \\
&\le f(\tilde{w}_t)-\gamma\|\nabla f(w_t)\|_2^2+(L-1+2\gamma)\|\nabla f(w_t)\|\|\mathbb{E}_t[\epsilon_t]\|+\frac{\gamma^2\rho}{2}\|\nabla f(w_t)\|_2^2+\frac{L^2}{2\rho}\mathbb{E}_t[\|r_t\|_2^2]+\|\mathbb{E}_t[\epsilon_t]\|_2^2+\frac{L}{2}\mathbb{E}_t[\|\epsilon_t\|_2^2]+\frac{L\gamma^2}{2}G^2 \\
&= f(\tilde{w}_t)-\gamma\left[1-\frac{\rho}{2}\gamma\right]\|\nabla f(w_t)\|_2^2+(L-1+2\gamma)\|\nabla f(w_t)\|\|\mathbb{E}_t[\epsilon_t]\|+\frac{L^2}{2\rho}\mathbb{E}_t[\|r_t\|_2^2]+\|\mathbb{E}_t[\epsilon_t]\|_2^2+\frac{L}{2}\mathbb{E}_t[\|\epsilon_t\|_2^2]+\frac{L\gamma^2}{2}G^2 \qquad (14)
\end{aligned}
$$

$$
\le \frac{1}{2-\frac{\rho}{\sqrt{T+1}}}\left\{\frac{2(f(w_0)-f^*)}{\sqrt{T+1}}+\frac{2LG^2}{\sqrt{T+1}}\left(\frac{L}{\rho}\frac{2(1-\delta)}{\delta^2}+\frac{1}{2}\right)+\frac{L}{T+1}\sum_{t=0}^{T}\underbrace{\mathbb{E}_t[\|\epsilon_t\|_2^2]}_{MSE}\right\} \qquad (16)
$$

V. Communication Error Minimization via Joint Optimization of Power Allocation and Learning Rates

Theorem 1 reveals that the communication error has a significant impact on the convergence behavior of the BLCD algorithm. In this section, we turn our attention to minimizing the communication error (in term of MSE and bias) via joint optimization of power allocation and learning rates.

Without loss of generality, we focus on iteration t (with abuse of notation, we omit t in the notation for simplicity). Recall that the coordinate updates in the BLCD algorithm, sent by different devices over the same subcarrier, are received by the edge server as an aggregate sum, which is used to estimate the gradient value in that specific dimension. We denote the power coefficients and estimators as $\boldsymbol{b} \triangleq [b_{11}, b_{12}, \ldots, b_{1M}, b_{21}, \ldots, b_{KM}]$ and $\boldsymbol{\alpha} \triangleq [\vec{\alpha}_1, \ldots, \vec{\alpha}_K]$. In each round, each sender device optimizes its power allocation for transmitting the selected coordinates of their updates over the K subcarriers, aiming to achieve the best convergence rate. We assume that the perfect channel state information is available at the corresponding transmitter, i.e., $\boldsymbol{h}_m = [h_{1m}, \ldots, h_{Km}]^\top$ is available at the sender m only.

Based on (10), the mean squared error of the communication error in iteration t is given by

$$
\mathbb{E}_t[\|\epsilon_t\|_2^2] = \mathbb{E}\left[\left\|\widehat{G}_t(w_t) - G_t(w_t)\right\|^2\right] \qquad (17)
$$

where the expectation is taken over the channel noise. For convenience, we denote $\mathbb{E}_t[\|\epsilon_t\|_2^2]$ as MSE_1, and after some algebra, it can be rewritten as the sum of the variance and the square of the bias:

$$
\text{MSE}_1(\boldsymbol{\alpha}, \boldsymbol{b}) = \sum_{k=1}^{K}\underbrace{\left[\sum_{m=1}^{M}\left(\alpha_k b_{km} h_{km} - \frac{1}{M}\right)x_{km}\right]}_{\text{bias in } k\text{th coordinate}}{}^2 + \underbrace{\sum_{k=1}^{K}\sigma^2\alpha_k^2}_{\text{variance}} \qquad (18)
$$

Recall that $\{\alpha_k\}_{k=1}^K$ are intimately related to the learning rates for the K coordinates, making the learning rate effectively $\{\gamma\alpha_k\}_{k=1}^K$.

A. Centralized Solutions to Minimizing MSE (Scheme 1)

In light of the above, we can cast the MSE minimization problem as a learning-driven joint power allocation and learning rate problem, given by

$$
\textbf{P1:} \min_{\boldsymbol{\alpha}, \boldsymbol{b}} \quad \text{MSE}_1(\boldsymbol{\alpha}, \boldsymbol{b}) \qquad (19)
$$

$$
\text{s.t.} \quad \sum_{k=1}^{K}|b_{km}x_{km}|^2 \le E_m, \quad \forall m \qquad (20)
$$

$$
b_{km} \ge 0, \ \alpha_k \ge 0 \quad \forall k, m \qquad (21)
$$

which minimizes the MSE for every round.

The above formulated problem is non-convex because the objective function involves the product of variables. Nevertheless, it is biconvex, i.e., for one of the variables being fixed, the problem is convex for the other one. In general, we can solve the above bi-convex optimization problem in the same spirit as in the EM algorithm, by taking the following two steps, each optimizing over a single variable, iteratively:

$$
\textbf{P1-a:} \min_{\boldsymbol{\alpha}} \quad \text{MSE}_1(\boldsymbol{\alpha}, \boldsymbol{b}) \quad \text{s.t.} \quad \alpha_k \ge 0, \quad \forall k
$$

$$
\textbf{P1-b:} \min_{\boldsymbol{b}} \quad \text{MSE}_{11}(\boldsymbol{\alpha}, \boldsymbol{b})
$$

$$
\text{s.t.} \quad \sum_{k=1}^{K}|b_{km}x_{km}|^2 \le E_m \quad \forall m, \quad b_{km} \ge 0 \quad \forall k, m.
$$

Since (**P1-a**) is unconstrained,for given $\{b_{km}\}$, the optimal solution to (**P1-a**) is given by

$$
\alpha_k^* = \max\left\{\frac{\left(\sum_{m=1}^{M}x_{km}\right)\left(\sum_{m=1}^{M}b_{km}h_{km}x_{km}\right)}{M\left[\sigma^2+\left(\sum_{m=1}^{M}b_{km}h_{km}x_{km}\right)^2\right]}, 0\right\}. \qquad (22)
$$

Then, we can solve (**P1-b**) by optimizing $\boldsymbol{b}$ only. Solving the sub-problems (**P1-a**) and (**P1-b**) iteratively leads to a local minimum, however, not necessarily to the global solution.

Observe that the above solution requires the global knowledge of x_{km}'s and h_{km}'s of all devices, which is difficult to implement in practice. We will treat it as a benchmark only. Next, we turn our attention to developing distributed suboptimal solutions.

B. Distributed Solutions towards Zero Bias and Variance Reduction (Scheme 2)

As noted above, the centralized solution to (**P1**) requires the global knowledge of x_{km}'s and h_{km}'s and hence is not amenable to implementation. Further, minimizing the MSE of the communication error does not necessarily amount

to minimizing the bias therein since there exists tradeoffs between bias and variance. Thus motivated, we next focus on devising distributed sub-optimal solutions which can drive the bias in the communication error to (close to) zero, and then reduce the corresponding variance as much as possible.

Specifically, observe from (18) that the minimization of MSE cost does not necessarily ensure $\hat{G}$ to be an unbiased estimator, due to the intrinsic tradeoff between bias and variance. To this end, we take a sub-optimal approach where the optimization problem is decomposed into two subproblems. In the subproblem at the transmitters, each device m utilizes its available power and local gradient/channel information to compute a power allocation policy in terms of $\{b_{1m}, b_{2m}, \ldots, b_{Km}\}$. In the subproblem at the receiver, the receiver finds the best possible α_k for all $k = 1, \ldots, K$. Another complication is that due to the power constraints at individual devices, it is not always feasible to achieve unbiased estimators of the gradient signal across the coordinates. Nevertheless, for given power constraints, one can achieved unbiased estimators of a scaled down version of the coordinates of the gradient signal. In light of this, we formulate the optimization problem at each device (transmitter) m to ensure an unbiased estimator of a scaled version ζ_m of the transmitted coordinates, as follows:

$$\textbf{Device m:} \quad \max_{\{b_{km}\}_{k=1:K}} \ \zeta_m \tag{23}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} b_{km}^2 x_{km}^2 \le E_m, \quad b_{km} \ge 0, \tag{24}$$

$$\zeta_m x_{km} - b_{km} h_{km} x_{km} = 0, \quad \forall k = 1, \ldots, K, \tag{25}$$

where maximizing ζ_m amounts to maximizing the corresponding SNR (and hence improving the gradient estimation accuracy). The first constraint in the above is the power constraint, and the second constraint is imposed to ensure that there is no bias of the same scaled version of the transmitted signals across the dimensions for user m. The power allocation solution can be found using Karush-Kuhn-Tucker (KKT) conditions as follows:

$$\boxed{\zeta_m^* = \sqrt{\frac{E_m}{\sum_{k=1}^{K} \frac{x_{km}^2}{h_{km}^2}}}, \ b_{km}^* = \frac{\zeta_m^*}{h_{km}}, \ \forall k.} \tag{26}$$

Observe that using the obtained power allocation policy in (26), all K transmitted coordinates for device m have the same scaling factor ζ_m. Next, we will ensure zero bias by choosing the right $\boldsymbol{\alpha}$ for gradient estimation at the receiver, which can be obtained by solving the following optimization problem since all transmitted gradient signals are superimposed via the over-the-air transmission:

$$\textbf{Receiver side:} \ \min_{\{\alpha_k\},} \ \sum_{k=1}^{K} \nu_k^2(\alpha_k, \{b_{km}^*\}) \tag{27}$$

$$\text{s.t. } e_k(\alpha_k, \{b_{km}^*\}) = 0, \quad \alpha_k \ge 0, \forall k = 1, \ldots, K, \tag{28}$$

where e_k and ν_k^2 denote the bias and variance components, given as follows:

$$e_k(\alpha_k, \{b_{km}^*\}) = \alpha_k \left(\sum_{m=1}^{M} \zeta_m^* x_{km} \right) - \frac{1}{M} \sum_{m=1}^{M} x_{km},$$

$$\nu_k^2(\alpha_k, \{b_{km}^*\}) = \alpha_k^2 \sigma^2, \tag{29}$$

for all $k = 1, \ldots, K$. For given $\{\zeta_m^*\}$, it is easy to see that

$$\boxed{\alpha_k^* = \frac{\frac{1}{M}\sum_{m=1}^{M} x_{km}}{\sum_{m=1}^{M} \zeta_m^* x_{km}} \simeq \frac{1}{\sum_{m=1}^{M} \zeta_m^*}, \quad \forall k.} \tag{30}$$

We note that in the above, from an implementation point of view, since $\{x_{km}\}$ is not available at the receiver, it is sensible to set $\alpha_k^\dagger \simeq \frac{1}{\sum_{m=1}^{M} \zeta_m^*}$. Further, $\{\zeta_m^*\}$ is not readily available at the receiver either. Nevertheless, since there is only one parameter ζ_m^* from each sender m, the sum $\sum_{m=1}^{M} \zeta_m^*$ can be sent over a control channel to the receiver to compute $\alpha_k^\dagger$. It is worth noting that in general the bias exists even if E_m is the same for all senders.

Next, we take a closer look at the case when the number of subchannels K is large (which is often the case in practice). Suppose that $\{x_{km}\}$ are i.i.d. across subchannels and users, and so are $\{h_{km}\}$. We can then simplify ζ_m^* further. For ease of exposition, we denote $\mathbb{E}[x_{km}^2] = \varphi^2 + \bar{x}^2$ and $\mathbb{E}\left[\frac{1}{h_{km}^2}\right] = \varpi^2$. When K is large, for every user m we have that:

$$\zeta_m^* = \frac{\sqrt{E_m}}{\sqrt{\sum_{k=1}^{K} \frac{x_{km}^2}{h_{km}^2}}} \underset{\text{when } K \text{ is large}}{\Longrightarrow} \zeta_m^* \approx \frac{\sqrt{E_m}}{\sqrt{K(\varphi^2 + \bar{x}^2)\varpi^2}} \tag{31}$$

As a result, the bias and variance for each dimension k could be written as,

$$e_k(\alpha_k^*, \{b_{km}^*\}) = \sum_{m=1}^{M} \left[\frac{\sqrt{E_m}}{\sum_{m=1}^{M} \sqrt{E_m}} - \frac{1}{M} \right] x_{km}, \forall k. \tag{32}$$

$$\nu_k^2 = \frac{K\varpi^2(\varphi^2 + \bar{x}^2)}{\left(\sum_{m=1}^{M} \sqrt{E_m}\right)^2} \sigma^2, \forall k. \tag{33}$$

Observe that when E_m is the same across the senders, the bias term $\mathbb{E}_t[\epsilon_t] = \mathbf{0}$ in the above setting according to (32).

C. A User-centric Approach Using Single-User Solution (Scheme 3)

In this section, we consider a suboptimal user-centric approach, which provides insight on the power allocation across the subcarriers from a single device perspective. We formulate the single device (say user m) problem as

$$\textbf{P2:} \ \min_{\{b_{km}\},\{\alpha_k\}} \ \sum_{k=1}^{K} \left[\left(\alpha_k b_{km} h_{km} - 1\right) x_{km} \right]^2 + \sigma^2 \sum_{k=1}^{K} \alpha_k^2$$

$$\text{s.t.} \ \sum_{k=1}^{K} |b_{km} x_{km}|^2 \le E_m; \ b_k \ge 0, \ \alpha_k \ge 0, \forall k.$$

Theorem 2. *The optimal solution $\{b_{km}^*, \alpha_k^*\}$ to (**P2**) is given by*

$$(b_{km}^*)^2 = \left[\sqrt{\frac{\sigma^2}{\lambda x_{km}^2 h_{km}^2}} - \frac{\sigma^2}{h_{km}^2 x_{km}^2} \right]^+, \forall k, \tag{34}$$

$$\alpha_k^* = \frac{b_{km}^* h_{km} x_k^2}{\sigma^2 + (b_{km}^*)^2 h_{km}^2 x_{km}^2}, \quad \forall k, \tag{35}$$

where λ_m is a key parameter determining the waterfilling level:

$$\sum_{k=1}^{K} \left[\sqrt{\frac{1}{\lambda_m}} \sqrt{\frac{x_{km}^2 \sigma^2}{h_{km}^2}} - \frac{\sigma^2}{h_{km}^2} \right]^+ = E_m. \tag{36}$$

The proof of Theorem 2 is omitted due to space limitation. Observe that Theorem 2 reveals that the larger the gradient value (and the smaller channel gain) in one subcarrier, the higher power the it should be allocated to in general, and that $\{x_{km}/h_{km)}\}$ can be used to compute the water level for applying the water filling policy.

Based on the above result, in the multi-user setting, each device can adopt the above single-user power allocation solution as given in Theorem 2. This solution can be applied individually without requiring any coordination between devices.

Next, we take a closer look at the case when the number of subchannels K is large. Let $\bar{E}_m$ denote the average power constraint per subcarrier. When K is large, after some algebra, the optimization problem **P2** can be further approximated as follows:

$$\textbf{P3:} \min_{b_{km}} \mathbb{E}\left[\frac{x_{km}^2\sigma^2}{b_{km}^2 h_{km}^2 x_{km}^2+\sigma^2}\right] \quad \text{s.t. } \mathbb{E}\left[b_{km}^2 x_{km}^2\right] \le \bar{E}_m,\ b_{km}\ge 0, \tag{37}$$

where the expectation is taken with respect to $\{h_{km}\}$ and $\{x_{km}\}$.

The solution for $k = 1, \ldots, K$ is obtained as follows:

$$b_{km}^*=\sqrt{\left[\frac{\sigma|x_{km}|^{-1}}{h_{km}\sqrt{\lambda_m}}-\frac{\sigma^2}{x_{km}^2 h_{km}^2}\right]^+} \tag{38}$$

$$\lambda_m < \frac{h_{km}^2 x_{km}^2}{\sigma_k^2} \Rightarrow b_{km}^*>0 \tag{39}$$

We can compute the bias and the variance accordingly.

VI. Coordinate Selection for Bandlimited Coordinate Descent Algorithms

The selection of which coordinates to operate on is crucial to the performance of sparsified SGD algorithms. It is not hard to see that selecting the top-k (in absolute value) coordinates of the sum of the gradients provides the best performance. However, in practice it may not always be feasible to obtain top-k of the sum of the gradients, and in fact there are different solutions for selecting k dimensions with large absolute values; see e.g., [22], [27]. Note that each device individually transmitting top-k coordinates of their local gradients is not applicable to the scenario of over-the-air communications considered here. Sequential device-to-device transmissions provides an alternative approach [28], but these techniques are likely to require more bandwidth with wireless connection.

Another approach that is considered is the use of compression and/or sketching for the gradients to be transmitted. For instance, in [22], a system that updates SGD via decompressing the compressed gradients transmitted through over-the-air communication is examined. To the best of our knowledge, such techniques do not come with rigorous convergence guarantees. A similar approach is taken in [27], where the sketched gradients are transmitted through an error-free medium and these are then used to obtain top-k coordinates; the devices next simply transmit the selected coordinates. Although such an approach can be taken with over-the-air computing since only the summation of the sketched gradients is necessary; this requires the transmission of $\mathcal{O}(k \log d)$ dimensions. To provide guarantees with such an approach $\mathcal{O}(k \log d + k)$ up-link transmissions are needed. Alternatively, uniformly selected $\mathcal{O}(k \log d + k)$ coordinates can be transmitted with similar bandwidth and energy requirements. For the practical learning models with non-sparse updates, uniform coordinate selection tend to perform better. Moreover, the common K dimensions can be selected uniformly via synchronized pseudo-random number generators without any information transfer. To summarize, uniform selection of the coordinates is more attractive based on the energy, bandwidth and implementation considerations compared to the methods aiming to recover top-k coordinates; indeed, this is the approach we adopt.

VII. Experimental Results

In this section, we evaluate the accuracy and convergence performance of the BLCD algorithm, when using one of the following three schemes for power allocation and learning rate selection (aiming to minimize the impact of communication error): 1) the bi-convex program based solution (Scheme 1), 2) the distributed solution towards zero bias in Section V. (Scheme 2); 3) the single-user solution (Scheme 3). We use the communication error free scheme as the baseline to evaluate the performance degradation. We also consider the naive scheme (Scheme 4) using equal power allocation for all dimensions, i.e., $b_{km} = \sqrt{E/\sum_{k=1}^{K} x_{km}^2}$.

In our first experiment, we consider a simple single layer neural network trained on the MNIST dataset. The network consists of two 2-D convolutional layers with filter size 5×5 followed by a single fully connected layer and it has 7840 parameters. $K = 64$ dimensions are uniformly selected as the support of the sparse gradient transmissions. For convenience, we define E_{avg} as the average sum of the energy (of all devices) per dimension normalized by the channel noise variance, i.e., $E_{avg} = EM\,\mathbb{E}[h_{km}^2]/K\sigma^2$. Without loss of generality, we take the variance of the channel noise as $\sigma^2 = 1$ and $\{h_{km}\}$ are independent and identically distributed Rayleigh random variables with mean 1. The changes on E_{avg} simply amount to different SNR values. In Fig. 3, we take $K = 64$, $M = 8$, batch size 4 to calculate each gradient, and the learning rate $\gamma = 0.01$. In the second experiment, we expand the model to a more sophisticated 5-layer neural network and an 18-layer ResNet [29] with 61706 and 11175370 parameters, respectively. The 5-layer network consists of two 2-D convolutional layers with filter size 5×5 followed by three fully connected layers. In all experiments, we have used a learning rate of 0.01. the local dataset of each worker is randomly selected from the entire MNIST dataset. We use 10 workers with varying batch sizes and we utilize $K = 1024$ sub-channels for sparse gradient signal transmission.

It can be seen from Fig. 3 that in the presence of the communication error, the centralized solution (Scheme 1) based on bi-convex programming converges quickly and performs the best, and it can achieve accuracy close to the ideal error-free scenario. Further, the distributed solution (Scheme

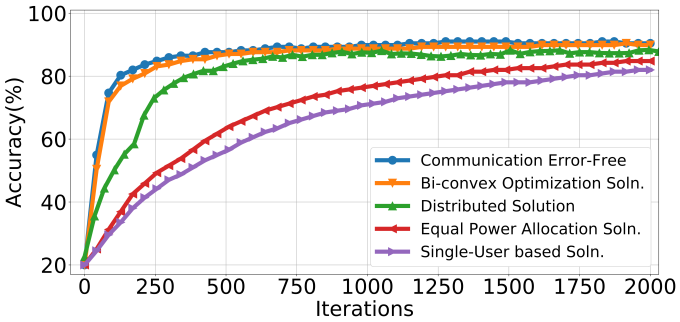

Fig. 3. Testing accuracy over training iterations for $\alpha_k = 1/8$, $E_{avg} = 0.1$ and a batch size of 4. Training model consists of a single layer neural network with 7840 differentiable parameters.

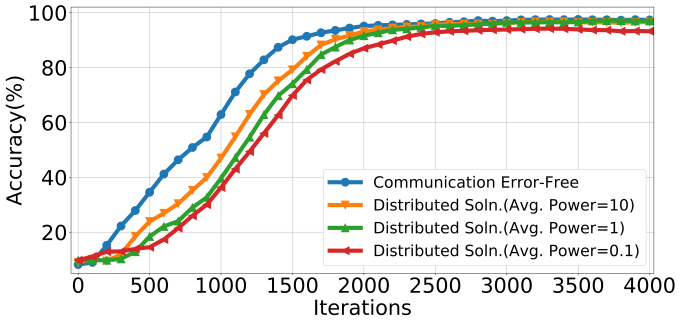

Fig. 4. Testing accuracy over training iterations for 10 workers and a batch size of 256. Training model consists of a 5-layer deep neural network with 61706 differentiable parameters.

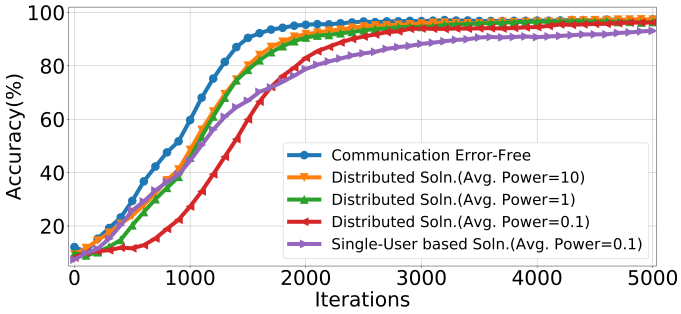

Fig. 5. Testing accuracy over training iterations for 10 workers and a batch size of 4. Training model consists of a 5-layer deep neural network with 61706 differentiable parameters.

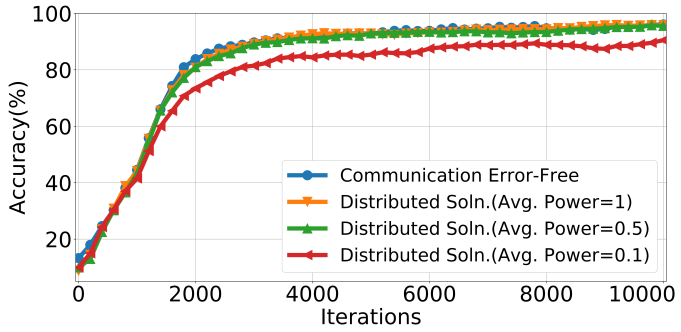

Fig. 6. Testing accuracy over training iterations for 10 devices and a batch size of 4. Training model consists of an 18-layer ResNet network with more than 11 million differentiable parameters.

2) can eventually approach the performance of Scheme 1, but the single-user solution (Scheme 3) performs poorly, so does the naive scheme using equal power allocation (Scheme 4). Clearly, there exists significant gap between its resulting accuracy and that in the error-free case, and this is because the bias in Scheme 3 is more significantly.

Next, Figures 4, 5 and 6 depict the results in the second experiment using much larger-scale deep neural networks. It can be observed from Figs. 4, 5 and 6 that the SNR can have significant impact on the final accuracy. As expected, the convergence on the ResNet network is slower in comparison to other DNNs due to the huge number of parameters and small batch size. Nevertheless, it is clear that the learning accuracy improves significantly at high SNR. (The solution of the distributed algorithm for $E_{avg} = 10$ is omitted in Fig. 6, since it is indistinguishably close to error-free solution.) It is interesting to observe that when the SNR increases, the distributed solution (Scheme 2) can achieve accuracy close to the ideal error-free case, but the single-user solution (Scheme 3) would not. It is worth noting that due to the computational complexity of bi-convex programming in this large-scale case, Scheme 4 could be solved effectively (we did not present it here). Further, the batch size at each worker can impact the convergence rate, but does not impact the final accuracy.

VIII. Conclusions

In this paper, we consider a many-to-one wireless architecture for distributed learning at the network edge, where multiple edge devices collaboratively train a machine learning model, using local data, through a wireless channel. Observing the unreliable nature of wireless connectivity, we design an integrated communication and learning scheme, where the local updates at edge devices are carefully crafted and compressed to match the wireless communication resources available. Specifically, we propose SGD-based bandlimited coordinate descent algorithms employing over-the-air computing, in which a subset of k-coordinates of the gradient updates across edge devices are selected by the receiver in each iteration and then transmitted simultaneously over k subcarriers. We analyze the convergence of the algorithms proposed, and characterize the effect of the communication error. Further, we study joint optimization of power allocation and learning rates therein to maximize the convergence rate. Our findings reveal that optimal power allocation across different sub-carriers should take into account both the gradient values and channel conditions. We then develop sub-optimal solutions amenable to implementation and verify our findings through numerical experiments.

Acknowledgements

The authors thank Gautam Dasarathy for insightful discussion in the early stage of this work. This work is supported in part by NSF Grants CNS-2003081, CNS-CNS-2003111, CPS-1739344 and ONR YIP N00014-19-1-2217.

REFERENCES

[1] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Towards an intelligent edge: Wireless communication meets machine learning," *arXiv preprint arXiv:1809.00343*, 2018.

[2] M. Goldenbaum and S. Stanczak, "Robust analog function computation via wireless multiple-access channels," *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3863–3877, 2013.

[3] O. Abari, H. Rahul, and D. Katabi, "Over-the-air function computation in sensor networks," *arXiv preprint arXiv:1612.02307*, 2016.

[4] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1709–1720.

[5] 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," in *Advances in Neural Information Processing Systems*, 2017, pp. 1509–1519.

[6] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "Signsgd: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*, 2018, pp. 559–568.

[7] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized sgd and its applications to large-scale distributed optimization," in *International Conference on Machine Learning*, 2018, pp. 5321–5329.

[8] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 440–445.

[9] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," in *Advances in Neural Information Processing Systems*, 2018, pp. 4447–4458.

[10] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Advances in Neural Information Processing Systems*, 2018, pp. 5973–5983.

[11] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[12] S. U. Stich, "Local sgd converges fast and communicates little," in *ICLR 2019 International Conference on Learning Representations*, 2019.

[13] J. Dong, Y. Shi, and Z. Ding, "Blind over-the-air computation and data fusion via provable wirtinger flow," *arXiv preprint arXiv:1811.04644*, 2018.

[14] W. Liu and X. Zang, "Over-the-air computation systems: Optimization, analysis and scaling laws," *arXiv preprint arXiv:1909.00329*, 2019.

[15] D. Wen, G. Zhu, and K. Huang, "Reduced-dimension design of MIMO over-the-air computing for data aggregation in clustered iot networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5255–5268, 2019.

[16] G. Zhu and K. Huang, "MIMO over-the-air computation for high-mobility multi-modal sensing," *IEEE Internet of Things Journal*, 2018.

[17] X. Cao, G. Zhu, J. Xu, and K. Huang, "Optimal power control for over-the-air computation in fading channels," *arXiv preprint arXiv:1906.06858*, 2019.

[18] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, 2019.

[19] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *arXiv preprint arXiv:1812.11750*, 2018.

[20] Q. Zeng, Y. Du, K. K. Leung, and K. Huang, "Energy-efficient radio resource allocation for federated edge learning," *arXiv preprint arXiv:1907.06040*, 2019.

[21] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *arXiv preprint arXiv:1901.00844*, 2019.

[22] ——, "Federated learning over wireless fading channels," *arXiv preprint arXiv:1907.09769*, 2019.

[23] ——, "Over-the-air machine learning at the wireless edge," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[24] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2019, pp. 1–6.

[25] T. Sery and K. Cohen, "On analog gradient descent learning over multiple access fading channels," *arXiv preprint arXiv:1908.07463*, 2019.

[26] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error feedback fixes signsgd and other gradient compression schemes," *arXiv preprint arXiv:1901.09847*, 2019.

[27] N. Ivkin, D. Rothchild, E. Ullah, V. Braverman, I. Stoica, and R. Arora, "Communication-efficient distributed sgd with sketching," *arXiv preprint arXiv:1903.04488*, 2019.

[28] S. Shi, Q. Wang, K. Zhao, Z. Tang, Y. Wang, X. Huang, and X. Chu, "A distributed synchronous sgd algorithm with global top-k sparsification for low bandwidth networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 2238–2247.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.