# Over-the-Air Federated Learning Exploiting Channel Perturbation

Shayan Mohajer Hamidi*, Mohammad Mehrabi†, Amir K. Khandani*, and Deniz Gündüz§
*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
†Department of Data Sciences and Operations, University of Southern California, Los Angeles, USA
§Department of Electrical and Electronic Engineering, Imperial College London, London, UK
Email: {smohajer, khandani}@uwaterloo.ca*, mehrabim@usc.edu†, d.gunduz@imperial.ac.uk§

***Abstract*—Federated learning (FL) is a promising technology which trains a machine learning model on edge devices in a distributed manner orchestrated by a parameter server (PS). To realize fast model aggregation, the uplink phase of FL could be carried out by over-the-air computation (OAC). On the one hand, engaging more devices in FL yields a model with higher prediction accuracy. On the other hand, the edge devices in OAC need to perform appropriate magnitude alignment to compensate for underlying channel coefficients. However, due to the limited power budget, this is not possible for devices experiencing deep fade. Consequently, these devices are excluded from the FL algorithm. In this paper, we propose a channel perturbation method so that no edge device is excluded due to experiencing deep fade. To this end, OAC is performed in multiple phases. In each phase, the radio frequency (RF) vicinity of PS's antenna is intentionally perturbed by means of RF mirror structure coined in [1]. This yields independent realizations of channels between PS and devices in each phase. By using proper transmit scalars, all devices concurrently transmit their local model updates in each phase subject to a total power constraint. Then, the PS estimates the arithmetic sum of the local updates by properly combining the aggregated models obtained across all phases. The devices' transmit scalars and PS's de-noising factors can be efficiently found by solving a tractable optimization problem.**

***Index Terms*—Federated learning, over-the-air computation, edge machine learning, wireless communications.**

## I. Introduction

Conventionally, a machine learning (ML) model is trained in a centralized approach where the training data is available at a data center or a cloud server [2]. However, in many emerging applications, data samples are collected by edge devices, e.g. smartphones, which brings up two concerns: (i) the devices often do not want to share their private data with a remote server, and (ii) sharing extremely large datasets is a huge burden on the communication links between the devices and the server. As a remedy to these concerns, federated learning (FL) was proposed in [3] where each device participates in training using only locally available dataset with the help of a parameter server (PS). Specifically, in FL, devices share only model parameters and their local updates with the PS, and not their raw dataset, to (i) preserve the privacy of the devices, and (ii) decrease the communication load. In FL, a global trained model is obtained after a number of communication rounds between the PS and devices. Each communication round consists of transmitting the current global model from the PS to the devices, training local models at the devices in parallel, and then aggregating the local updates by PS to update the global model.

A common way to carry out such communications is to divide the channel resources among devices, e.g., by using frequency division multiplexing (FDM), where each device transmits its own model update by encoding it against channel imperfections. The PS tries to decode as many of them as possible, and computes a global update by averaging the local updates. An alternative FL approach has recently emerged [4]–[8] from the fact that the PS is only interested in finding the average of the model updates, and not in their individual values. Therefore, if all the devices simultaneously transmit their updates with appropriate magnitude alignment, the model updates are averaged out. This method, referred to as over-the-air computation (OAC) [9], [10], can improve the communication efficiency and reduce the required bandwidth by combining communication and computation.

Participating in OAC requires each device to compensate for the underlying channel gain between itself and the PS by means of channel inversion precoding. However, in order to avoid excessive transmit power due to channel inversion, transmission is possible only if the channel gain is above a certain threshold [5]–[8]. Consequently, devices with weak channel coefficients under deep channel fading are excluded from the learning task [5]. However, in FL, it is known that engaging more devices participating in each round can improve both the convergence time of training, and the final prediction accuracy [3], [4]. Engaging more devices in the learning task is even more important when the datasets across devices are not independent and identically distributed (non-IID). In this case, excluding devices may lead to omitting some training samples from the learning process, resulting in a bias.

To tackle the above issue, we propose a setup in which no active device is excluded from FL due to channel state. To this end, the uplink transmission of local model updates is carried out over $N$ transmissions, referred to as *phases* hereafter. In each *phase*, the radio frequency (RF) environment in the vicinity of the PS's antenna is intentionally perturbed by means of an RF mirror structure, as proposed in [1]. This guarantees that in each *phase*, independent realizations of channels between the devices and PS are obtained [11], [12], and all active devices simultaneously transmit their local updates using a precoding scalar subject to an individual

total power constraint (the available power at each device for transmitting the model updates is distributed over $N$ *phases*; this is in accordance with assuming an average power budget for each device [13]). The probability that channel realizations across all *phases* experience deep fade is very low, and therefore, as a rule of thumb, a device can transmit its update over the channel with a better condition at a lower power cost. Then, the PS employs $N$ de-noising factors to linearly combine the aggregated models received in all $N$ *phases* to minimize the aggregation error. More precisely, we optimize the devices' precoding scalars and the PS's de-noising factors to efficiently obtain the desired global model update with minimum noise.

We refer to the proposed method as OAC via channel perturbation, wherein all active devices participate in each FL iteration. The numerical experiments highlight that, in comparison to the channel inversion method [5], [8], the proposed approach is more accurate and efficient. More precisely, it can simultaneously improve the convergence time and the training accuracy. The improvement becomes more significant when the number of *phases* is small.

The rest of this paper is organized as follows. Section II describes the synchronous FL setup and also the RF mirror structure used by the PS. Section III formalizes the OAC via channel perturbation which boils down to solving an alternating optimization problem in Section IV. Section V presents numerical results, and Section VI concludes the paper.

## II. System model

### A. Synchronous FL System

ML algorithms often entail minimization of the empirical loss function of the form $F(\boldsymbol{\theta}) = \frac{1}{K}\sum_{i=1}^{K} f(\boldsymbol{\theta}, \boldsymbol{u}_i)$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ are model parameters to be optimized, $\boldsymbol{u}_i$ for $i \in [K]$, are the training data samples, and $f(\cdot)$ is the loss function that depends on the ML model. Iterative stochastic gradient descent (SGD) is often employed to minimize $F(\boldsymbol{\theta})$. In SGD, the model parameters at iteration $t$, denoted by $\boldsymbol{\theta}^t$, are updated as $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha^t \mathbf{g}(\boldsymbol{\theta})$, where $\mathbb{E}[\mathbf{g}(\boldsymbol{\theta})] = \nabla F(\boldsymbol{\theta})$, and $\alpha^t$ is the learning rate. SGD can easily be implemented in a distributed fashion across multiple devices, where device $D_k$ has access to only its local dataset $\mathcal{D}_k$. At each iteration of distributed SGD (DSGD), $D_k$ computes a gradient vector based on the global parameter vector with respect to $\mathcal{D}_k$, and sends back the result to the PS. Afterwards, PS updates the global parameter vector as follows

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha^t \frac{1}{K}\sum_{k=1}^{K} \mathbf{g}_k(\boldsymbol{\theta}^t), \tag{1}$$

where $K$ is the number of devices, and $\mathbf{g}_k(\boldsymbol{\theta}^t) \triangleq \frac{1}{|\mathcal{D}_k|}\sum_{\boldsymbol{u}_i \in \mathcal{D}_k} \nabla f(\boldsymbol{\theta}^t, \boldsymbol{u}_i)$ is the gradient estimate of device $k$ with respect to the global parameter vector $\boldsymbol{\theta}^t$ using its local dataset $\mathcal{D}_k$. In FL, each device can carry out a number of local updates between two global aggregations. In a synchronous FL setup, the PS would wait for all the clients to complete local training at each iteration. Iterations continue until a certain convergence criterion is met. To simplify the notation, we omit the dependency of $\mathbf{g}_k(\boldsymbol{\theta}^t)$ on $\boldsymbol{\theta}^t$, and simply use $\mathbf{g}_k$.

### B. Channel Perturbation by RF Mirrors

In the proposed method, we assume that the PS's antenna is equipped with the RF mirror structure as proposed in [1]. This structure consists of a number of switchable parasitic elements, referred to as RF mirrors. This is an essential part of the proposed system by means of which the RF environment of the PS's antenna is intentionally perturbed. Each surrounding mirror can be selectively turned on (to reflect light back to the interior of the structure), or off (to let the light rays leave the structure). Therefore, if the mirror structure has $M$ mirrors, then $2^M$ mirror states could be realized. As discussed in [11], [12], the outgoing RF signal corresponding to each mirror state will take various independent paths in reaching the distant receiver, resulting in a different complex gain for multi-path fading in a rich scattering environment. Therefore, switching the mirrors to a new on-off state yields an independent realization of the underlying channel between each device and PS. In addition, we assume that the channel realizations are quasi-static; that is, they are random, yet remain the same throughout the FL algorithm. Our simulation results show that a small number of mirror states is sufficient to obtain good results in our scenario.

## III. OAC via Channel Perturbation

Prior to transmitting the local gradient vectors, device $D_k$, $k \in [K]$, transforms its gradient vector $\mathbf{g}_k \in \mathbb{R}^d$ into a normalized symbol vector $\mathbf{x}_k \in \mathbb{R}^d$ with zero mean and unit variance such that $\mathbb{E}[\mathbf{x}_k \mathbf{x}_{k'}^H] = \mathbf{0}$ and $\mathbb{E}[\mathbf{x}_k \mathbf{x}_k^H] = \mathbf{I}_d$, where $\mathbf{0}$ is the all-zero matrix. Denote by $x_k[i]$ the i-th entry of $\mathbf{x}_k$. The vectors $\mathbf{x}_k$, $k \in [K]$, are transmitted in an uncoded fashion; therefore, $d$ symbols are required to send $\mathbf{x}_k$. In the proposed method, however, each entry $x_k[i]$ is transmitted over $N$ symbols spread over $N$ *phases* (see Fig. 1). In each *phase*, PS switches to a different mirror state. Therefore, if using $N$ *phases*, $N$ different mirror states are utilized. We label the selected mirror states by $1, 2, \ldots, N$. The PS uses the same $N$ mirror states for the entire course of training. We assume that channel state information (CSI) for the specified $N$ mirror states is known at the PS. To this end, for each of these $N$ states, devices are required to transmit pilot sequences to the PS. In addition, we assume that the channels are static, and therefore, pilot transmission is required only once before the FL algorithm starts. In the following, the proposed method is elaborated for *Phase* $n$, $n \in [N]$.

• ***Phase*** $n$, $n \in [N]$: RF mirrors at the PS are switched to state $n$, and all the devices concurrently transmit the i-th entry of their gradient vector. Device $D_k$ scales its input by scalar $a_{k,1} \in \mathbb{C}$ before transmission. Then, the $d$-dimensional signal received at the PS in *phase* $n$ is given by

$$\mathbf{y}_n = \sum_{k=1}^{K} h_{k,n} a_{k,n} \mathbf{x}_k + \mathbf{z}_n, \tag{2}$$

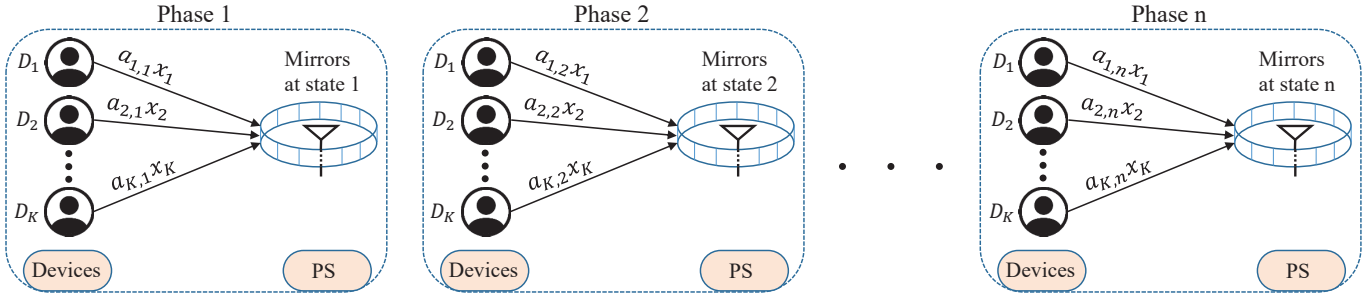


Fig. 1. The uplink transmission in the proposed method takes place in $N$ *phases*. In each *phase*, the RF mirror structure at the PS is switched to a new state to realize independent channels between the PS and devices. The devices use transmit scalars in each *phase* subject to an individual power constraint for the total power utilized in all the transmissions.

where $h_{k,n}$ denotes the channel gain from $D_k$ to the PS when the mirrors are at state $n$, and $\mathbf{z}_n \in \mathbb{C}^d$ is a complex Gaussian noise vector with its i-th entry $z_n[i]$ IID according to $\mathcal{N}(0, \sigma^2)$.

We consider an individual transmit power constraint for each communication round, which is calculated as follows in our setup:

$$\mathbb{E}\left[\sum_{n=1}^{N} |a_{k,n} x_k[i]|^2\right] = \sum_{n=1}^{N} |a_{k,n}|^2 \leq P_0, \tag{3}$$

for $k \in [K]$, where the expected value is taken over the $d$ entries of the gradient vector, and $P_0 > 0$ is the maximum transmit power. Equation (3) implies that the average power used by $D_k$ for sending $\mathbf{x}_k$ over $N$ *phases* is bounded. In other words, $P_0$ must be distributed over $N$ *phases*.

After receiving the signals $\mathbf{y}_n$ for $n \in [N]$ over the $N$ *phases*, the PS uses $N$ de-noising factors $\eta_n \in \mathbb{C}$, $n \in [N]$, to linearly combine them to calculate its estimate of $\mathbf{x} = \sum_{k=1}^{K} \mathbf{x}_k$ as follows

$$\hat{\mathbf{x}} \triangleq \sum_{n=1}^{N} \eta_n \mathbf{y}_n = \sum_{n=1}^{N} \sum_{k=1}^{K} \eta_n h_{k,n} a_{k,n} \mathbf{x}_k + \sum_{n=1}^{N} \eta_n \mathbf{z}_n. \tag{4}$$

Based on this estimation, the PS broadcasts the updated global gradient vector to the devices. In the next iteration, devices will carry out the same $N$ *phases* using the same $N$ mirror states to upload their local updates. This process will continue until a certain pre-defined termination criterion is met. We refer to the above design as over-the-air model aggregation using channel perturbation. To quantify the performance of the proposed method of data aggregation, similarly to [4], we use mean square error (MSE) that measures the distortion between $x[i]$ and $\hat{x}[i]$ calculated as

$$\begin{aligned} \mathrm{MSE}\left(\hat{x}[i], x[i]\right) &= \mathbb{E}\left[|\hat{x}[i] - x[i]|^2\right] \\ &= \sum_{k=1}^{K} \left| \sum_{n=1}^{N} \eta_n h_{k,n} a_{k,n} - 1 \right|^2 + \left(\sum_{n=1}^{N} |\eta_n|^2\right) \sigma^2. \end{aligned} \tag{5}$$

As discussed in [14], a smaller MSE yields a larger learning convergence rate. Thus, in Section IV, we aim to minimize the MSE calculated in (5). However, for brevity and ease of notation, the minimization problem is solved when $N = 2$, i.e., only two *phases* are exploited for uploading the local updates. The derivations could be easily generalized for $N$ *phases*.

*Remark 1:* In this paper, we assumed that the number of training samples is the same across devices, and thus the PS aims to find the arithmetic sum of the local updates (and not their weighted sum) for an unbiased estimate of the gradient at each iteration. The generalization to the imbalanced training datasets of different sizes is straight-forward.

*Remark 2:* As discussed earlier, here we assumed quasi-static channels over the course of training. Note that our method is orthogonal to exploiting frequency diversity when the channels experience non-flat fading; in this case, as another source of diversity, the PS can appropriately combine the received signals over different frequency bands.

*Remark 3:* By assuming that all the devices are in the communication range of the PS, the proposed scheme tackles the problem of deep fade incurred by small-scale fading.

## IV. Optimization for MSE Minimization

In this section, our goal is to minimize the MSE in (5) for $N = 2$. The minimization is over the devices' transmit scalars $\{a_{k,1}, a_{k,2}, \forall k\}$ and PS's de-noising factors $\{\eta_1, \eta_2\}$ subject to a power constraint at each device, which can be written as

$$\begin{aligned} \min_{\{a_{k,1}, a_{k,2}, \eta_1, \eta_2\}} & \sum_{k=1}^{K} |\eta_1 h_{k,1} a_{k,1} + \eta_2 h_{k,2} a_{k,2} - 1|^2 \\ & + (|\eta_1|^2 + |\eta_2|^2)\sigma^2 \end{aligned} \tag{6a}$$

$$\text{s.t.} \quad |a_{k,1}|^2 + |a_{k,2}|^2 \leq P_0, \quad \forall k \in [K]. \tag{6b}$$

The optimization problem in (6) is non-convex because of the coupling variables in its objective function. We use a low-complexity alternating minimization method to solve (6). To this end, we split our parameters into two groups, namely $\mathcal{G}_1 = \{a_{k,1}, a_{k,2}\}_{k=1:K}$, and $\mathcal{G}_2 = \{\eta_1, \eta_2\}$. In each iteration, we start by optimizing over $\mathcal{G}_1$ parameters, for given $\mathcal{G}_2$ values, and then switch the roles of $\mathcal{G}_1$ and $\mathcal{G}_2$. We next characterize the closed-form solutions for each optimization problem.

**Optimizing over $\mathcal{G}_1$ given $\mathcal{G}_2$.** For $k \in [K]$ introduce $\mathbf{a}_k, \mathbf{c}_k \in \mathbb{C}^2$ as $\mathbf{a}_k = [a_{k,1}, a_{k,2}], \mathbf{c}_k = [\eta_1 h_{k,1}, \eta_2 h_{k,2}]$. In particular, for given values of $\eta_1, \eta_2$ the optimization problem (6) can be written as the following:

$$\min_{\mathbf{a}_1, \ldots, \mathbf{a}_K \in \mathbb{C}^2} \sum_{k=1}^{K} |\mathbf{a}_k^{\mathsf{T}} \mathbf{c}_k - 1|^2 \tag{7a}$$

$$\text{s.t } \|\mathbf{a}_k\|_{\ell_2}^2 \leq P_0, \forall k \in [K]\,. \tag{7b}$$

The Lagrangian is given by

$$\mathcal{L}(\mathbf{a}_{1:K}, \lambda_{1:K}) = \sum_{k=1}^{K} |\mathbf{a}_k^{\mathsf{T}}\mathbf{c}_k - 1|^2 + \lambda_k(\|\mathbf{a}_k\|_{\ell_2}^2 - P_0)$$
$$\lambda_k \geq 0.$$

The optimization problem in (7) involves the summation of $K$ independent objective functions; therefore, it is separable into $K$ sub-problems that can be solved independently. It is a quadratic program with $\ell_2$-ball constraints, whose feasible region has a non-empty interior. Therefore, strong duality holds, and the solutions obtained by Karush–Kuhn–Tucker (KKT) conditions are given by

$$\mathbf{a}_k^* = \frac{\mathbf{c}_k}{\lambda_k^* + \|\mathbf{c}_k\|_{\ell_2}^2}, \quad \lambda_k^* = \left(\frac{\|\mathbf{c}_k\|_{\ell_2}}{\sqrt{P_0}} - \|\mathbf{c}_k\|_{\ell_2}^2\right)^+, \quad (8)$$

where $(\cdot)^+$ represents the ramp function with $(x)^+ = \max(0, x)$, and $(\cdot)^*$ denotes the optimal value for a parameter. In this case, the optimal parameters of optimization (6) for given values of $\eta_1, \eta_2$ are the following

$$a_{k,1}^* = \frac{\eta_1 h_{k,1}}{v_k}, \quad a_{k,2}^* = \frac{\eta_2 h_{k,2}}{v_k}, \quad \forall k \in [K], \quad (9)$$

where for $u_k = \eta_1^2 h_{k,1}^2 + \eta_2^2 h_{k,2}^2$, the value of $v_k$ is given by

$$v_k = u_k + \left(\sqrt{u_k/P_0} - u_k\right)^+.$$

**Optimizing over $\mathcal{G}_2$ given $\mathcal{G}_1$.** For given values of $\mathbf{a}_{1:K}$, the optimization problem (6) is a quadratic programming with no constraints. We first introduce $d_j \in \mathbb{C}^K$, $j = 1, 2$,

$$d_j = [a_{1,j}h_{1,j}, ..., a_{K,j}h_{K,j}]^{\mathsf{T}}.$$

The first order optimality conditions imply

$$\eta_1^*(\sigma^2 + \|d_1\|_{\ell_2}^2) + \eta_2^* d_1^{\mathsf{T}} d_2 = d_1^{\mathsf{T}}\mathbf{1}, \quad (10)$$
$$\eta_1^* d_1^{\mathsf{T}} d_2 + \eta_2^*(\sigma^2 + \|d_2\|_{\ell_2}^2) = d_2^{\mathsf{T}}\mathbf{1}, \quad (11)$$

where $\mathbf{1} \triangleq [1, 1, ..., 1]^{\mathsf{T}}$. Solving the system of linear equations (10)-(11) gives us

$$\eta_1^* = \frac{d_1^{\mathsf{T}}1(\sigma^2 + \|d_2\|_{\ell_2}^2) - d_2^{\mathsf{T}}1 d_1^{\mathsf{T}} d_2}{(\sigma^2 + \|d_1\|_{\ell_2}^2)(\sigma^2 + \|d_2\|_{\ell_2}^2) - (d_1^{\mathsf{T}} d_2)^2}, \quad (12)$$

$$\eta_2^* = \frac{-d_1^{\mathsf{T}}1 d_1^{\mathsf{T}} d_2 + d_2^{\mathsf{T}}1(\sigma^2 + \|d_1\|_{\ell_2}^2)}{(\sigma^2 + \|d_1\|_{\ell_2}^2)(\sigma^2 + \|d_2\|_{\ell_2}^2) - (d_1^{\mathsf{T}} d_2)^2}. \quad (13)$$

In order to deploy the above iterative relations, we need to initialize the optimization parameters with proper values. To this end, we initialize $\{a_{k,1}, a_{k,2}, \forall k, \eta_1, \eta_2\}$ as follows: the power budget is equally allocated to each transmission *phase* (i.e., each *phase* has $P_0/2$ power budget), and then the channel inversion power control is exploited for each transmission *phase* separately. The algorithm is terminated when either (i) relative increase in the objective function (6a) is less than a predefined threshold $\epsilon$, or (ii) the maximum number of iterations $J_{max}$ is reached. The optimization problem (6) can be solved efficiently by Algorithm 1.

*Remark 4:* We assume that the PS performs Algorithm 1, and then broadcasts the solution to the devices (the PS has sufficient power to solve such iterative problems). In addition, by assuming that the channels are static, the PS performs Algorithm 1 only once.

**Algorithm 1** Alternating Method for Solving (6)

**Input:** $J_{max}$ and $\epsilon$
**Output:** $\{a_{k,1}^*, a_{k,2}^*, \eta_1^*, \eta_2^*\}$

**Initialization:**
$a_{k,1}^{(0)} = \frac{P_0/2}{h_{k,1}\max_{k\in[K]}\frac{1}{|h_{k,1}|}}$, $a_{k,2}^{(0)} = \frac{P_0/2}{h_{k,2}\max_{k\in[K]}\frac{1}{|h_{k,2}|}}$ $\forall k$;
$\eta_1^{(0)} = \frac{1}{\sqrt{2P_0}}\max_{k\in[K]}\frac{1}{|h_{k,1}|}$, $\eta_2^{(0)} = \frac{1}{\sqrt{2P_0}}\max_{k\in[K]}\frac{1}{|h_{k,2}|}$.
Calculate MSE by plugging $\{a_{k,1}^{(0)}, a_{k,2}^{(0)}, \forall k, \eta_1^{(0)}, \eta_2^{(0)}\}$ into (6a);
**for** $i = 1$ to $J_{max}$ **do**
  Calculate $\{a_{k,1}^{(i)}, a_{k,2}^{(i)}, \forall k\}$ from (9);
  Calculate $\eta_1^{(0)}$ from (12);
  Calculate $\eta_2^{(0)}$ from (13);
  Calculate $\text{MSE}^{(i)}$ by plugging $\{a_{k,1}^{(i)}, a_{k,2}^{(i)}, \forall k, \eta_1^{(i)}, \eta_2^{(i)}\}$ into (6a);
  **if** $\frac{\text{MSE}^{(i)} - \text{MSE}^{(i-1)}}{\text{MSE}^{(i)}} \leq \epsilon$ **then**
    **break**;
  **end if**
**end for**
**return** $\{a_{k,1}^{(i)}, a_{k,2}^{(i)}, \eta_1^{(i)}, \eta_2^{(i)}\}$.

## V. Simulation Results

In this section, we conduct numerical experiments to evaluate the performance of the proposed method. To this end, we consider classification of handwritten digits using MNIST dataset [15]. This dataset contains 60,000 training and 10,000 testing hand-written images of 10 digits. We use a neural network (NN) with one hidden layer of 128 hidden nodes, and ReLU activation function. We consider K = 20 active edge devices for our FL problem. Each device uses a mini-batch size of 10 and the learning rate is set to $\alpha^t = 0.01$, $\forall t$. We set the number of local iterations at the devices to 3.

We assume that the training dataset is evenly distributed, each device having the same number of training samples. On the other hand, we consider non-IID data distribution by introducing a new variable $L$ in the following manner: each device first selects $L$ digits/labels at random, and then uniformly samples its local dataset from the whole dataset with labels being among the $L$ selected digits. To illustrate this better, for L=2, the first device may uniformly sample from digits '0' and '5', and device two does the same for digits '4' and '7'. Based on this definition, $L = 1$ implies a high level non-IID distribution.

The channel coefficient between the PS and device $k$ at transmission *phase* $n$ follows the IID complex normal distribution, i.e., $h_{k,n} \sim \mathcal{CN}(0, 1)$. The noise power is set to $\sigma^2 = 1$ and the average power constraint is $P_0 = 20$. Additionally, in Algorithm 1, we set $J_{max} = 100$ and $\epsilon = 10^{-4}$.

We use two benchmarks to evaluate the performance of the proposed method: (i) benchmark 1, devices with a channel gain below a certain threshold are excluded, and the others use channel-inversion power control [13]; (ii) benchmark 2, error-free transmission, which is equivalent to the centralized SGD

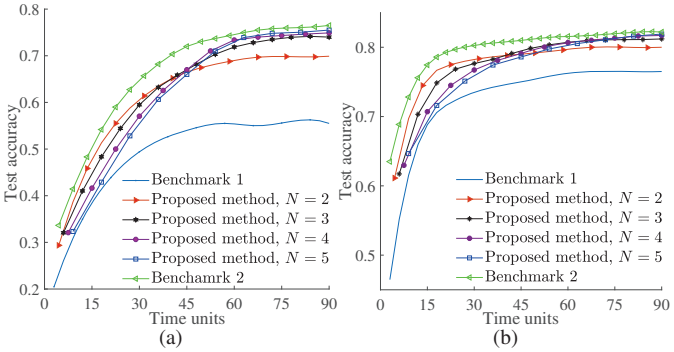


Fig. 2. The test accuracy for the benchmarks and proposed method Vs. time units for (a) $L = 1$ and (b) $L = 5$. The figures are obtained using the Monte-Carlo method by averaging over 1000 random realizations.

algorithm. For a fair comparison between the benchmarks and the proposed method, we need to consider the effect of the delay caused by our method that results from multiple-phase transmission. To this aim, we assume that the transmission of a single gradient vector takes one time unit. Based on this assumption, a communication round in the traditional FL using OAC takes two time units, where a one-shot transmission by the users is followed by a transmission by the PS. However, in the proposed setup, since OAC is performed over $N$ *phases*, each communication round would take $N + 1$ time units; that is, $N$ time units for uploading the local updates and one time unit for the downlink transmission from the PS. Henceforth, the performance of the proposed method is measured as the accuracy with respect to the test data samples, called test accuracy, versus time units.

Fig. 2 shows the test accuracy for six FL methods, namely the two benchmarks, and the proposed method using $N = \{2, 3, 4, 5\}$ *phases*. In Fig. 2a, we have $L = 1$, i.e., the non-IID level of the local datasets is high. For the same time units, the proposed method achieves a higher test accuracy than the benchmark 1. As the number of *phases* $N$ increases, (i) the number of independent channel realizations available to the users increases, and (ii) the probability that all the $N$ channel realizations experience deep fade becomes smaller. Thus, with more *phases* available, the PS can more accurately estimate the sum of the local updates at each iteration. Specifically, since the accuracy of PS's estimations becomes more important after some iterations, the test accuracy curves in Fig. 2a will be ordered by their respective phase numbers after enough time units (after time $\approx$ 70). On the other hand, it is seen that the test accuracy has an inverse relation with the number of *phases* at the beginning of the FL communication rounds (prior to time $\approx 45$). This is because a noisy estimate of the sum of the local updates can be sufficient in the first few iterations, and using more *phases* only incurs excessive delay.

Additionally, it is observed that the ultimate test accuracy achieved by using $N = 4$ and $N = 5$ *phases* is almost the same, suggesting that using more than $N = 4$ *phases* does not noticeably enhance the test accuracy. We also observe that the proposed method yields a better accuracy than the benchmark 1 for both $L = 1$ and $L = 5$ cases, but the improvement of the proposed method is more significant when the non-IID level of the local datasets is high.

## VI. Conclusion

In this work, we considered federated edge learning with OAC, and aimed at increasing the number of devices participating in the learning process by mitigating channel fading. To this end, by means of an RF mirror structure at the PS's antenna, independent realizations of the underlying channels between the PS and edge devices are obtained. Then, the devices concurrently transmit their local updates in each phase exploiting a proper precoding scalar subject to a power constraint over the phases. The PS linearly combines the aggregated updates obtained in each phase aiming to minimize the MSE of the estimated sum of the local updates. In essence, the proposed scheme exploits bandwidth expansion to improve the computation accuracy over fading channels. The simulation results justify that the proposed algorithm increases the test accuracy of the trained model, and decreases the convergence time of training. For future work, we plan to extend our framework to support the digital OAC as well.

## References


[1] A. K. Khandani, "Media-based modulation: A new approach to wireless transmission," in *2013 IEEE International Symposium on Information Theory*. IEEE, 2013, pp. 3050–3054.

[2] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[4] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, 2020.

[5] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, 2020.

[6] T. Sery and K. Cohen, "On analog gradient descent learning over multiple access fading channels," *IEEE Trans. Signal Process.*, vol. 68, pp. 2897–2911, 2020.

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

[8] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

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

[10] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing interference for analog function computation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 61, no. 20, pp. 4893–4906, 2013.

[11] Y. Naresh and A. Chockalingam, "On media-based modulation using rf mirrors," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4967–4983, 2017.

[12] S. M. Hamidi, A. K. Khandani, and E. Bateni, "A secure key sharing algorithm exploiting phase reciprocity in wireless channels," *arXiv preprint arXiv:2111.15046*, 2021.

[13] X. Cao, G. Zhu, J. Xu, and K. Huang, "Optimized power control for over-the-air computation in fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7498–7513, 2020.

[14] H. Liu, X. Yuan, and Y.-J. A. Zhang, "Reconfigurable intelligent surface enabled federated learning: A unified communication-learning design approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7595–7609, 2021.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.