Over-the-Air Federated Learning with Retransmissions (Extended Version)

Henrik Hellström, *Student Member, IEEE,* Viktoria Fodor, *Member, IEEE,* and Carlo Fischione, *Senior Member, IEEE*

arXiv:2111.10267v1 [eess.SP] 19 Nov 2021

***Abstract*—Motivated by increasing computational capabilities of wireless devices, as well as unprecedented levels of user- and device-generated data, new distributed machine learning (ML) methods have emerged. In the wireless community, Federated Learning (FL) is of particular interest due to its communication efficiency and its ability to deal with the problem of non-IID data. FL training can be accelerated by a wireless communication method called Over-the-Air Computation (AirComp) which harnesses the interference of simultaneous uplink transmissions to efficiently aggregate model updates. However, since AirComp utilizes analog communication, it introduces inevitable estimation errors. In this paper, we study the impact of such estimation errors on the convergence of FL and propose retransmissions as a method to improve FL convergence over resource-constrained wireless networks. First, we derive the optimal AirComp power control scheme with retransmissions over static channels. Then, we investigate the performance of Over-the-Air FL with retransmissions and find two upper bounds on the FL loss function. Finally, we propose a heuristic for selecting the optimal number of retransmissions, which can be calculated before training the ML model. Numerical results demonstrate that the introduction of retransmissions can lead to improved ML performance, without incurring extra costs in terms of communication or computation. Additionally, we provide simulation results on our heuristic which indicate that it can correctly identify the optimal number of retransmissions for different wireless network setups and machine learning problems.**

***Index Terms*—Federated Learning, Over-the-Air Computation, Retransmissions.**

I. Introduction

THE data collection rate in wireless devices is growing at an exceptional speed due to the increasing adoption of smartphones, tablets, and Internet of Things (IoT) devices [1], [2]. These devices are expected to provide a broad range of Artificial Intelligence (AI) services in Sixth Generation (6G) networks, such as predictive healthcare [3], search-and-rescue drones [4], and environmental monitoring [5]. As a consequence, new distributed machine learning methods, such as Federated Learning (FL), have become essential to enable privacy-preserving and communication-efficient model training [6]. A recent survey on open problems of FL, argues that communication is often a primary bottleneck for FL because wireless links operate at low rates that can be both expensive and unreliable [7]. Communication-efficient FL is investigated thoroughly in [8], where various compression techniques such as quantization, random rotation, and subsampling are evaluated. In [9]–[12] it is established that new wireless methods can greatly improve the communication efficiency of edge AI.

A relatively unexplored approach to wireless communication, called Over-the-air computation (AirComp) (also known as computation over multiple-access channels), has recently been adapted to support Machine Learning (ML) services [13], [14]. AirComp is an analog communication scheme that orders its users to communicate simultaneously over the same frequency band, thereby promoting interference. This interference is leveraged to compute a function of the transmitted messages by utilizing the superposition property of the wireless channel [15]. By appropriately precoding the transmitted signals, many functions can be calculated over the air, for instance, the arithmetic mean, product, and min functions [16]. In FL, the central server is interested in collecting the arithmetic mean of model updates from the participating devices, and therefore AirComp is a suitable communication solution [10].

Compared to conventional point-to-point digital communications, AirComp is very attractive from a communication-efficiency standpoint, with throughput gains approximately proportional to the number of users [13]. The reason for this drastic improvement is that the entire wireless spectrum can be utilized concurrently by all devices, rather than dividing it and allocating a smaller resource block to each device. Additionally, AirComp obfuscates the participating users since the central server directly receives the arithmetic mean rather than the individual model updates, thereby enhancing privacy [17].

Currently, AirComp is reliant upon specialized hardware and fine synchronization that might be difficult to achieve in practice [18]. Additionally, AirComp is unable to guarantee perfect reconstruction of the transmitted messages at the receiver. Shannon's "fundamental theorem for a discrete channel with noise" establishes that for any degree of noise contamination, it is possible to communicate discrete data with an arbitrarily small frequency of errors [19]. However, to achieve a non-zero communication rate, redundant information must be transmitted in the form of a code [20]. Since the information transmitted in AirComp is not discrete, existing codes do not appear to be applicable. Instead, AirComp settles for estimating the desired function as closely as possible, while retaining some non-zero estimation error [21]. In [22], it is proved that these errors reduce the convergence rate of FL, thereby requiring additional communication rounds to reach an optimum.

All authors were with the School of Electrical Engineering and Computer Science, KTH - Royal Intitute of Technology, Stockholm, Sweden. E-mails: (hhells@kth.se, vjfodor@kth.se, carlofi@kth.se).

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

In the current AirComp literature, the main way of reducing the estimation error is to optimize the transmission powers. In [21] and [23], the authors propose a closed-form power control scheme that minimizes the mean squared error (MSE) between the received signal and the true sum of the sources' messages. They consider single-antenna devices with a peak transmission power constraint. A similar problem is solved in [24], where a sum-power constraint is used to preserve the long-term energy consumption of the participating devices. For the case of multiple antennas, no closed-form power control scheme has been found, but [14] develops a strong heuristic by using a difference-of-convex-functions representation of the problem. In [25], the multi-antenna problem is coupled with wireless power transfer to improve the battery life of participating IoT devices. To further improve the power control, [26] proposes a gradient-statistics aware scheme that learns statistical properties of the model updates to improve the AirComp estimation error. Similarly, [27] learns the temporal structure of gradient sparsity to develop a Bayesian prior that improves the estimation. Finally, several works have incorporated intelligent reflective surfaces with AirComp to reach substantially lower estimation errors [28]–[30].

As a general pattern, none of these works offer avenues to trade-off communication resources for improved estimation. In digital communications, such communication-estimation trade-offs are the main way to reduce errors. For instance, it is standard to adaptively control the modulation order and coding rate to compensate for poor channels [31]. Unfortunately, none of these approaches are directly compatible with AirComp since the communication is analog. In this paper, we take a first step towards enabling this communication-estimation tradeoff for Over-the-Air federated learning with a system we call AirReComp. The contributions of this paper are summarized as follows.

- This is the first work that allows communication resources to be traded for improved estimation in Over-the-Air FL;
- A power control scheme for AirReComp is proposed. The proposed scheme is proven to be globally optimal in terms of MSE between the estimated and desired function.
- Upper bounds on the FL loss function are derived for single-epoch Lipschitz-smooth functions, both for the strongly-convex and convex case.
- Using the bound for convex functions, we develop a heuristic for selecting the number of retransmissions that minimize the FL loss given a limited energy budget;
- To further support the feasibility of AirReComp under non-convex functions, we provide numerical results with Deep Neural Networks (DNNs). These results suggest that AirReComp can outperform state-of-the-art Over-the-Air FL in terms of classification accuracy, without increasing the total cost of training.

The remainder of the paper is organized as follows. Section II introduces the system model of Federated Learning, Over-the-Air Computation, and the retransmissions. Section III presents and solves the power control problem to minimize the MSE between the desired and received sum. Section IV provides worst-case analysis on the performance of AirReComp in terms of two upper bounds on the FL loss function. These bounds are then used in Section V to develop a heuristic for selecting the number of retransmissions that minimizes the FL loss without exceeding communication and computation cost constraints. In Section VI, the performances of AirReComp and the retransmission selection heuristic are numerically evaluated for non-convex loss functions. Finally, section VII concludes the work and discusses future work.

Notation: z is a scalar, $\mathbf{z}$ is a vector, and $\mathbf{Z}$ is a matrix. Element i of vector $\mathbf{z}$ is expressed as $z^{(i)}$. To denote element-wise operations of vectors, we overload the scalar equivalent, e.g. $\mathbf{x}/\mathbf{y}$ is the element-wise division of $\mathbf{x}$ and $\mathbf{y}$. $\bar{z}$ denotes the complex conjugate. $\hat{z}$ denotes an estimate of z and $\tilde{\mathbf{z}}$ denotes a normalized version of $\mathbf{z}$.

II. System Model

In this section, we describe the system model and the AirReComp algorithm. For the reader's convenience, we have included a table with all key variables, see Table I.

We consider a distributed ML system consisting of K single-antenna user devices each carrying a distinct dataset $\mathcal{D}_k$ and a single-antenna parameter server (PS) which can be reached by all devices in a single hop, as illustrated in Fig. 1. The objective of the system is to solve the following optimization problem

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}}\, F(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}}\, \frac{1}{K}\sum_{k=1}^{K} F_k(\mathbf{w}), \tag{1}$$

using the datasets at the user devices. The vector $\mathbf{w} \in \mathbb{R}^d$ is the $d \times 1$ parameter vector that defines the ML model, $F(\mathbf{w})$ is denoted the global loss function, and $F_k(\mathbf{w})$ is a local loss function. The choice of $F_k(\mathbf{w})$ describes the ML task, where for instance binary cross entropy loss corresponds to a classification task and mean-squared error to regression. The results of this paper are applicable to any problem that can be expressed as (1), and as such $F_k(\mathbf{w})$ can be seen as an arbitrary loss function

The uplink wireless channel is modeled as a block-fading multiple access channel (MAC) with additive noise [22], [32]. If the K users simultaneously transmit a vector $\mathbf{x}_k \in \mathbb{R}^d$ over the MAC, the PS receives

$$\mathbf{y} = \sum_{k=1}^{K} h_k \mathbf{x}_k + \mathbf{z}, \tag{2}$$

where $h_k \in \mathbb{C}$ denotes the channel coefficient from device k to the PS and $\mathbf{z} \in \mathbb{C}^d$ denotes additive white Gaussian noise (AWGN) with variance σ_z^2. Such a model is a good approximation of reality if all users transmit simultaneously and synchronously over a wireless channel [16], [21], [32]. For simplicity, we assume error-free broadcast transmission in the downlink, which is an acceptable approximation for most practical scenarios since the PS generally has much greater communication capability than the user devices [33].

TABLE I
REFERENCE LIST OF VARIABLES USED IN THIS PAPER. ORDERED ALPHABETICALLY AND BY CASE.

Variable	Interpretation
E	Number of local epochs in each communication round
$F(\mathbf{w}_n)$	Federated Learning loss function
K	Number of devices
M	Number of uplink transmissions per communication round
N	Total number of communication rounds
β	Static learning rate
d	Number of model parameters
η	Post-transmission scalar
h_k	Channel fading coefficient from device k to PS
L	Lipschitz smoothness parameter of $F(\mathbf{w}_n)$
μ	Strong convexity parameter of $F(\mathbf{w}_n)$
$\mu_{n,k}$	Mean of local update at iteration n and device k
p_k	Transmission power at k
$\sigma^2_{n,k}$	Variance of local update at iteration n and device k
σ^2_z	Variance of additive white Gaussian noise
$\boldsymbol{\sigma}$	Variance bound on local and global model difference
$\mathbf{w}_n$	Global model parameters at iteration n
$\Delta\mathbf{w}_n$	Global model update at iteration n
$\mathbf{w}_{n,k}$	Local model parameters at iteration n and device k
$\Delta\mathbf{w}_{n,k}$	Local model update
$\Delta\tilde{\mathbf{w}}_{n,k}$	Normalized local model update
$\Delta\hat{\mathbf{w}}_{n,k}$	Estimate of local model update
$\mathbf{x}_k$	Amplitude of transmitted signal from device k
$\mathbf{z}$	Additive white Gaussian noise

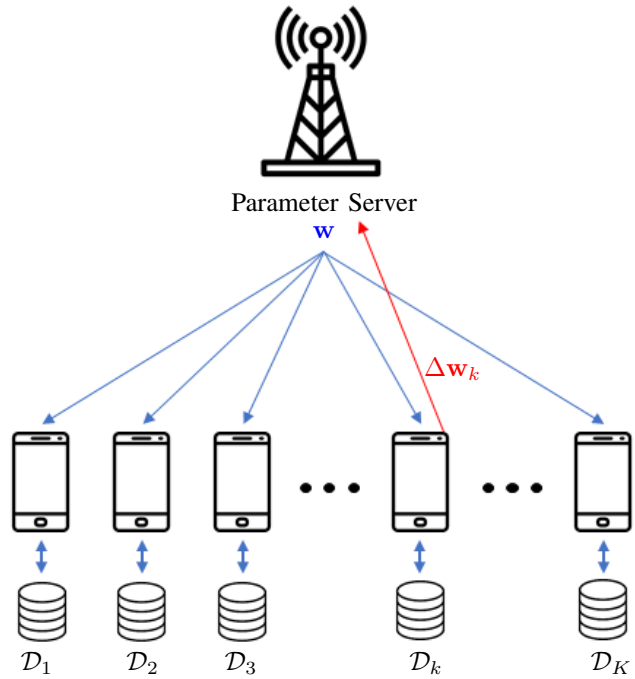

Fig. 1. Illustration of the Parameter Server and wireless network setup used throughout this paper. The training dataset is distributed among the User Devices, and all training takes place at the devices. The role of the Parameter Server is to orchestrate the communication of model updates from and to the User Devices.

A. Federated Learning Algorithm

FL is an iterative algorithm to solve (1), where each iteration is denoted a communication round and consists of downlink broadcast, model training at the user devices, and uplink aggregation. Each of these communication rounds is associated with a computational cost C_t (model training) and a communication cost C_u (uplink aggregation). Considering negligible downlink communication cost, the total cost of FL is $(C_t+C_u)N$, where N is the total number of communication rounds.

Communication round n starts when the PS broadcasts the global model $\mathbf{w}_n$ to all user devices in the downlink [12]. Upon receiving the model, user device k solves the local problem

$$\mathbf{w}^*_{n,k} = \underset{\mathbf{w}}{\operatorname{argmin}}\, F_k(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{\mathbf{u}_i \in \mathcal{D}_k} l(\mathbf{w}, \mathbf{u}_i) \tag{3}$$

where $\mathbf{u}_i$ denotes one training sample and $l(\mathbf{w}, \mathbf{u}_i)$ is the sample-wise loss function. Generally, (3) can not be solved exactly. Instead, each device runs E epochs of gradient descent to approximately solve (3) as follows

$$\mathbf{w}_{n,k}(i) \leftarrow \mathbf{w}_{n,k}(i-1) - \beta \nabla F_k(\mathbf{w}_{n,k}(i-1)), \forall i = 1, .., E, \tag{4}$$

where the first iteration is based on the global model, i.e. $\mathbf{w}_{n,k}(0) = \mathbf{w}_n$, and β is the step size. After executing E epochs, device k calculates a local model update as $\Delta\mathbf{w}_{n,k} = (\mathbf{w}_n - \mathbf{w}_{n,k}(E))/\beta$. After all local model updates have been computed, they are transmitted in the uplink to the PS.

At the PS, the local model updates are aggregated to form a global model update, written as

$$\Delta\mathbf{w}_n = \frac{1}{K}\sum_{k=1}^{K} \Delta\mathbf{w}_{n,k}. \tag{5}$$

Finally, the PS concludes the communication round by generating the next iteration of the model parameters

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \beta\Delta\mathbf{w}_n. \tag{6}$$

The algorithm repeats until $\mathbf{w}_N$ is generated as the final model. After all N communication rounds have completed, the PS broadcasts $\mathbf{w}_N$ to the user devices, so that every device has the ability to perform inference with the model.

B. Over-the-Air Computation Protocol

In the uplink aggregation step of FL, see (5), the PS reconstructs the sum of K model updates. In this section, we describe how this is achieved by AirComp.

To start, the devices normalize the model updates according to

$$\Delta\tilde{\mathbf{w}}_{n,k} = \frac{\Delta\mathbf{w}_{n,k} - \mu_{n,k}}{\sigma_{n,k}}, \tag{7}$$

where $\mu_{n,k}$ and $\sigma_{n,k}$ are calculated as

$$\begin{aligned} \mu_{n,k} &= \frac{1}{d}\sum_{i=1}^{d} \Delta w^{(i)}_{n,k} \\ \sigma^2_{n,k} &= \frac{1}{d-1}\sum_{i=1}^{d} \left|\Delta w^{(i)}_{n,k} - \mu_{n,k}\right|^2, \end{aligned} \tag{8}$$

where $\Delta w_{n,k}^{(i)}$ is element i of vector $\Delta \mathbf{w}_{n,k}$. For denormalization purposes, the values $\mu_{n,k}$ and $\sigma_{n,k}^2$ are transmitted to the PS over a separate control channel. Note that the overhead of transmitting 2 scalars per device is negligible for most FL problems. After normalizing, the model updates are communicated over the MAC defined in (2) as

$$\mathbf{x}_k = \Delta \tilde{\mathbf{w}}_{n,k} \frac{\bar{h}_k}{|h_k|} \sqrt{p_k}. \tag{9}$$

By combining (2) and (9) we get the received value at the PS

$$\mathbf{y} = \sum_{k=1}^{K} |h_k| \sqrt{p_k} \Delta \tilde{\mathbf{w}}_{n,k} + \mathbf{z}. \tag{10}$$

Ideally, the transmission powers would be chosen as $p_k = 1/|h_k|^2$, which would completely compensate for the fading effect. However, with a natural constraint on the maximum transmission power, $p_k = 1/|h_k|^2$ might be impossible to achieve. Because of this limitation and due to the additive noise, the PS can never perfectly reconstruct $\Delta \tilde{\mathbf{w}}_n$. Instead, it estimates $\Delta \tilde{\mathbf{w}}_n$ by dividing the received signal by a post-transmission scalar $\sqrt{\eta}$ and the number of devices K

$$\overline{\mathbf{y}} = \frac{\mathbf{y}}{\sqrt{\eta} K} = \sum_{k=1}^{K} \frac{|h_k| \Delta \tilde{\mathbf{w}}_{n,k} \sqrt{p_k}}{\sqrt{\eta} K} + \frac{\mathbf{z}}{\sqrt{\eta} K}. \tag{11}$$

Coupled with the transmission powers, $\sqrt{\eta}$ has an important role. We see that the ideal choice of the transmission powers is now $\sqrt{p_k} = \sqrt{\eta}/|h_k|$. As such, the selection of a small η will reduce the amount of energy required to invert a channel and thereby reduce the fading error. However, lowering η will also increase the relative power of the noise. Therefore, the post-transmission scalar $\sqrt{\eta}$ will play the role of a tradeoff parameter between the fading error and the noise-induced error [21].

In this work we propose AirReComp, which considers retransmissions in the uplink aggregation step. Specifically, the devices transmit the same values in the uplink M times. This way, the signal part of (11) will combine constructively, while the additive noise is different for each transmission. After receiving M values, the PS forms its estimate by calculating their arithmetic mean

$$\overline{\mathbf{y}} = \frac{\sum_{m=1}^{M} \mathbf{y}_m}{M \sqrt{\eta} K} = \sum_{k=1}^{K} \frac{|h_k| \Delta \tilde{\mathbf{w}}_{n,k} \sqrt{p_k}}{\sqrt{\eta} K} + \sum_{m=1}^{M} \frac{\mathbf{z}_m}{M \sqrt{\eta} K}. \tag{12}$$

Here, we have assumed static fading coefficients h_k for the duration of the M transmissions. To finalize the global update, the PS takes the real part of $\overline{\mathbf{y}}$ and denormalizes the result according to

$$\Delta \hat{\mathbf{w}}_n = \mathrm{Re}(\overline{\mathbf{y}}) \frac{\sum_{k=1}^{K} \sigma_{n,k}}{K} + \frac{\sum_{k=1}^{K} \mu_{n,k}}{K}. \tag{13}$$

This final estimate is then used in the global update step to generate the next iteration of the model as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \beta \Delta \hat{\mathbf{w}}_n. \tag{14}$$

The whole AirReComp process is summarized in Algorithm 1.

Algorithm 1 AirReComp

Parameter Server:
  initialize $\mathbf{w}_0$
for each round $n = 0, 1, .., N-1$ **do**
  Parameter Server:
    broadcast $\mathbf{w}_n$ to devices
  Device k**:**
    $\mathbf{w}_{n,k}(E) \leftarrow$ Equation (4)
    $\Delta \mathbf{w}_{n,k} \leftarrow (\mathbf{w}_n - \mathbf{w}_{n,k}(E))/\beta$
    $\Delta \tilde{\mathbf{w}}_{n,k} \leftarrow$ Equation (7)
    $\mathbf{x}_k \leftarrow$ Equation (9)
    transmit $\mu_{n,k}$ and $\sigma_{n,k}$ to server
  for each $m = 0, 1, ..., M-1$ **do**
    all devices simultaneously:
      transmit $\mathbf{x}_k$ to server
  end for
  Parameter Server:
    $\overline{\mathbf{y}} \leftarrow$ Equation (12)
    $\Delta \hat{\mathbf{w}}_n \leftarrow$ Equation (13)
    $\mathbf{w}_{n+1} \leftarrow \mathbf{w}_n - \beta \Delta \hat{\mathbf{w}}_n$
end for

Remark 1. *Since $\Delta \tilde{\mathbf{w}}_{n,k}$ has unit variance, p_k denotes the expected transmission power for each element of the model update, as follows*

$$\mathbb{E}[||\mathbf{x}_k||^2] = \mathbb{E}[\Delta \tilde{\mathbf{w}}_{n,k}^T \Delta \tilde{\mathbf{w}}_{n,k}] p_k = d p_k. \tag{15}$$

Remark 2. *If the PS only has access to the sum of the normalized model updates $\sum_{k=1}^{K} \Delta \tilde{\mathbf{w}}_{n,k}$, it is not possible to perfectly reconstruct $\Delta \mathbf{w}_n$ in general. However, the distortion caused by transmitting raw model updates with AirComp is generally much greater than the distortion caused by the denormalization scheme we opt for here. In the specific case of $\sigma_1 = \sigma_2 = ... = \sigma_K$, this scheme offers perfect reconstruction.*

III. Power Control

In this section, we consider a power control problem to minimize the mean-squared estimation error defined as

$$\mathbb{E}[(\Delta \mathbf{w}_n - \Delta \hat{\mathbf{w}}_n)^2], \tag{16}$$

where the expectation is taken over the AWGN, and $\Delta \mathbf{w}_n$ and $\Delta \hat{\mathbf{w}}_n$ are defined in (5) and (13) respectively. For mathematical tractability, we assume that these gradient elements are independent and identically distributed (IID) [21], [23], [24]. Additionally, we ignore the effect of normalization by considering $\mu_{n,k} = 0$ and $\sigma_{n,k} = 1$. To perform the minimization, we seek the optimal choice of the transmission powers $\sqrt{p_k}$ and the post-transmission scalar $\sqrt{\eta}$. Since we consider static fading coefficients, the power control problem only has to be solved once per communication round. To model the limited transmission power of the devices, we consider a peak power constraint

$$\mathbb{E}[|x_k^{(i)}|^2] = \mathbb{E}\left[|\Delta \tilde{w}_{n,k}^{(i)} \frac{\bar{h}_k}{|h_k|} \sqrt{p_k}|^2 \right] = p_k \leq P_{\max} \ \forall k, \tag{17}$$

where the expectation is taken over the model update $\Delta\tilde{w}_{n,k}^{(i)}$ and P_{max} is the maximum transmission power. The minimization of (16) is formulated as

$$\begin{aligned}&\min_{\mathbf{p},\eta}\\&\mathbb{E}\left[\left(\sum_{k=1}^{K}\frac{|h_k|\Delta\mathbf{w}_{n,k}\sqrt{p_k}}{\sqrt{\eta}K}+\sum_{m=1}^{M}\frac{\text{Re}(\mathbf{z}_m)}{M\sqrt{\eta}K}-\sum_{k=1}^{K}\frac{\Delta\mathbf{w}_n}{K}\right)^2\right]\\&\text{s.t.}\quad p_k\leq P_{\text{max}},\ \forall k\in\{1<k<K\}.\end{aligned}\tag{18}$$

Note that the number of transmissions M is given as an input parameter and is selected before the power control problem is solved.

Proposition 1. *Problem* (18) *has a unique solution. The optimal post-transmission scalar is given by the solution to the K subproblems*

$$\eta^*=\min_k\tilde{\eta}_k,\tag{19}$$

where

$$\tilde{\eta}_k=\left(\frac{\sum_{j=1}^{k}|h_j|^2\overline{P}+\sigma_z^2/M}{\sum_{j=1}^{k}|h_j|\sqrt{\overline{P}}}\right)^2.\tag{20}$$

The optimal transmission powers are

$$p_k^*=\min\left(\overline{P},\frac{\eta^*}{|h_k|^2}\right).\tag{21}$$

The proof of Proposition 1 follows the proof in [21] and is omitted from this paper.

Remark 3. *From* (20)*, we see that the post-transmission scalar $\sqrt{\eta}$ assumes a lower value when more retransmissions are used. As we increase the number of retransmissions, the signal-to-noise ratio (SNR) increases and consequently, the noise-induced error reduces. Therefore, the fading error becomes dominant and the optimal post-transmission scalar η^* is lowered to improve it.*

IV. Convergence Analysis

In this section, we analyze the learning performance of Algorithm 1. As before, we assume that the updates are zero-mean and unit variance, i.e. $\mu_{n,k}=0$, $\sigma_{n,k}=1$. Additionally, we assume that there is only one epoch of local training in each communication round ($E=1$). The performance is measured as the gap between the FL loss gap at iteration n, defined as

$$\mathbb{E}[F(\mathbf{w}_n)]-F(\mathbf{w}^*).\tag{22}$$

We derive two upper bounds on this loss gap, one for strongly-convex functions and one for convex functions. For both bounds, we use the following well-known lemma [22], [34].

Lemma 1. *Let $F(\mathbf{x}):\mathbb{R}^d\to\mathbb{R}$ be a convex function with L-Lipschitz gradient. Then, the following inequality holds:*

$$F(\mathbf{y})-F(\mathbf{x})-\nabla F(\mathbf{x})^T(\mathbf{y}-\mathbf{x})\leq\frac{L}{2}||\mathbf{x}-\mathbf{y}||^2.\tag{23}$$

Additionally, we make an assumption on the similarity of the local model updates $\Delta\mathbf{w}_{n,k}$ and the global model update $\Delta\mathbf{w}_n$ [26], [35].

Assumption 1. *The local model updates $\Delta\mathbf{w}_{n,k}$ are assumed to be independent and unbiased estimates of the global model update $\Delta\mathbf{w}_n$.*

$$\mathbb{E}[\Delta\mathbf{w}_{n,k}]=\Delta\mathbf{w}_n,\ \forall k\in\{1,2,\ldots,K\}.\tag{24}$$

The local model updates and the global gradient update are in general different. The difference has coordinate bounded variance:

$$\mathbb{E}[(\Delta w_{n,k}^{(i)}-\Delta w_n^{(i)})^2]\leq(\sigma^{(i)})^2,\tag{25}$$

where $\Delta w_{n,k}^{(i)}$ is the i-th element of $\Delta\mathbf{w}_{n,k}$, and $(\sigma^{(i)})^2$ are the element-wise upper bounds. We will also use $\boldsymbol{\sigma}\in\mathbb{R}^d$ to denote the vector of variance bounds.

A. Strongly-convex loss

In this subsection, we assume that the FL loss is μ-strongly convex. For such a loss, we use the following lemma [22], [34]:

Lemma 2. *Let $F(\mathbf{x}):\mathbb{R}^d\to\mathbb{R}$ be a μ-strongly convex function with L-Lipschitz gradient. Then, the following inequality holds:*

$$\begin{aligned}&(\nabla F(\mathbf{x})-\nabla F(\mathbf{y}))^T(\mathbf{x}-\mathbf{y})\geq\\&\frac{\mu L}{\mu+L}||\mathbf{x}-\mathbf{y}||^2+\frac{1}{\mu+L}||\nabla F(\mathbf{x})-\nabla F(\mathbf{y})||^2.\end{aligned}\tag{26}$$

Before we are ready to state the upper bound, we must also assert that the fixed step size β has been selected to be sufficiently small for convergence. For AirReComp, the norm of the gradient is dependent on the power with which the signal is received, as such, the step size must be upper bounded as a function of the power control.

Assumption 2. *Let the fixed step size β be:*

$$\beta<\min\left(\frac{K\sqrt{\eta}(\mu+L)}{2\mu L\sum_{k=1}^{K}\sqrt{p_k}|h_k|},\frac{2\sqrt{\eta}}{\mu+L}\frac{\sum_{k=1}^{K}\sqrt{p_k}|h_k|}{\sum_{k=1}^{K}p_k|h_k|^2}\right).\tag{27}$$

We are now ready to give the first upper bound on the FL loss function (22) given the update described in (13) and (14).

Proposition 2. *Let*

$$c_1\coloneqq\frac{\sum_{k=1}^{K}\sqrt{p_k}|h_k|}{\sqrt{\eta}},\tag{28}$$

$$c_2\coloneqq1-\frac{2\beta}{K}\frac{\mu L}{\mu+L}c_1,\tag{29}$$

and

$$c_3\coloneqq||\boldsymbol{\sigma}||^2\frac{\sum_{k=1}^{K}p_k|h_k|^2}{K\eta}+\frac{d\sigma_z^2}{MK^2\eta}.\tag{30}$$

Then the FL loss is upper bounded by

$$\mathbb{E}[F(\mathbf{w}_n)]-F(\mathbf{w}^*)\leq\frac{L}{2}c_2^n\mathbb{E}[r_0^2]+\frac{\beta^2L}{2(1-c_2)}c_3,\tag{31}$$

where $r_0=||\mathbf{w}_0-\mathbf{w}^*||$ is the distance between the initial weight vector and the optimal one, $\boldsymbol{\sigma}$ is a vector of the coordinate bounded variances from (25)*, and d is the number of model parameters.*

Proof. The proof is provided in Appendix A. □

We refer to the first term of the RHS of (31) as the *diminishing term*, because it approaches zero if $n \to \infty$. Along the same line, we refer to the other terms as the *post-convergence terms* because they remain even if $n \to \infty$. The implications of Proposition 2 are given in Section IV-C.

B. Convex loss

In this subsection, we relax the assumption on strong convexity and develop a bound for Lipschitz smooth and convex loss functions. For this bound, we need a different guarantee on the fixed step size than for the strongly convex case.

Assumption 3. *The fixed step size β is selected to satisfy:*

$$0 < \beta < \frac{\sqrt{\eta}}{L} \frac{\sum_{k=1}^{K} \sqrt{p_k}|h_k|}{\sum_{k=1}^{K} p_k |h_k|^2}. \tag{32}$$

Proposition 3. *Consider Assumption 1 and 3. Then the FL loss is upper bounded by*

$$\mathbb{E}[F(\mathbf{w}_n)] - F(\mathbf{w}^*) \leq \frac{K}{2n\beta c_1}\mathbb{E}[r_0^2] + \frac{\beta}{2}(\frac{K}{c_1} + L\beta)c_3, \tag{33}$$

where c_1 and c_3 are defined in (28) *and* (30) *respectively.*

Proof. The proof is provided in Appendix B. □

Just as for Proposition 2, we refer to the first term of the RHS of (31) as the *diminishing term*, and the other terms as the *post-convergence terms*. The implications of Proposition 3 are given in Section IV-C.

C. Discussion on Proposition 2 and 3

In this section, we discuss Proposition 2 and 3. Since the propositions are upper bounds, we are discussing the worst-case properties of the FL loss using AirReComp. We are specifically interested in the impact of the number of retransmissions.

1) Convergence Rate: As a shorthand, let x_n equal the diminishing term of (31) or (33) depending on which bound we are considering. Then, suppose the following holds

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|} = \alpha. \tag{34}$$

If $\alpha = 1$, we say that the worst-case convergence rate of AirReComp is sublinear, and if $0 < \alpha < 1$, we say that the worst-case convergence rate is linear.

For the strongly convex bound, we substitute x_n with the right-hand-side of (31) and L with the post-convergence terms. Basic algebra then tells us that $\alpha = c_2$. Since $c_2 < 1$, AirReComp has linear convergence rate for strongly-convex loss functions. Additionally, the rate of convergence is increasing with decreasing c_2. Since c_2 is decreasing in c_1, we can also say that the rate of convergence is increasing with c_1.

For the convex bound, the same procedure yields $\alpha = 1$. Therefore, AirReComp has sublinear convergence rate for convex loss functions. Since α is constant with respect to M, this indicates that the asymptotic rate of convergence is independent of the number of retransmissions. However, if we consider a finite n, the diminishing term of (33) is decreasing in c_1. Therefore, we say that the non-asymptotic convergence rate of AirReComp with convex loss functions is increasing in c_1. With these clarifications, we can state the following corollary.

Corollary 1. *The worst-case convergence rate of AirReComp is increasing in M.*

Proof. As clarified previously, the convergence rate is increasing in c_1 for both bounds. Therefore, we need to establish that c_1 is increasing in M. From (28), we see that c_1 depends on the power control policy via the post-transmission scalar η and the transmission powers p_k. We already know that η is decreasing in M from Remark 3. For the transmission powers, we see in (21) that the transmission powers take one of two values:

1) The device inverts its channel with $p_k = \eta/|h_k|^2$. These devices contribute 1 to the sum in c_1 regardless of M;
2) The device transmits with maximum power $p_k = \overline{P}$. These devices contribute $\sqrt{\overline{P}}|h_k|/\sqrt{\eta}$ to the sum in c_1. Since η is decreasing in M, these devices contribute more to c_1 for higher M.

With these two statements we can readily conclude that c_1 is strictly increasing in M, given the condition that at least one device transmits with maximum power. In [21], they prove that this condition always holds with optimal power control. Thus, we can conclude that c_1 is strictly increasing in M. □

2) Final error: Since both bounds have post-convergence terms, the algorithm does not converge to a local optimum. Instead, the algorithm converges to a region around the optimum, where the expected remaining loss gap is given by the post-convergence terms. There are two reasons why AirReComp does not converge exactly. Firstly, the channel noise (characterized by σ_z) causes unavoidable errors which prevents exact convergence. Secondly, the difference between local and global model updates (characterized by $\boldsymbol{\sigma}$) causes a global model update that differs from what is achieved in centralized gradient descent. This result aligns with what was found in [35].

Unfortunately, the post-convergence terms are not strictly decreasing in M[1]. Depending on the parameters, the diminishing terms can both increase and decrease with M. As a rule of thumb, if $||\boldsymbol{\sigma}||^2 >> \sigma_z^2$, the final error increases with M. However, due to resource constraints, training must generally be stopped long before convergence is reached. As such, the diminishing term tends to be dominant [37] and thus the convergence rate tends to be more relevant than the final error. It is worth to note that the gap could possibly be made to approach zero as $n \to \infty$ by considering learning rate scheduling, but in this paper we restrict ourselves to static β.

V. Heuristic for selecting M

As we have demonstrated in Section IV, the inclusion of retransmissions in Over-the-Air FL speeds up the worst-case

[1]We made a mistake in [36] claiming the opposite.

convergence speed of FL training. Simultaneously, the uplink communication cost is proportional to M. Therefore, M acts as a tradeoff parameter between communication cost and FL convergence speed. As such, the selection of M is an important factor in model training on resource-constrained devices.

To characterize this tradeoff more precisely, we consider a cost budget $\overline{C}$ for the whole training process from communication round $n = 0$ to the final round $n = N$. For the purposes of this paper, the cost budget $\overline{C}$ corresponds to an abstract cost, but could be replaced with, e.g., energy or time depending on the application. As introduced in Section II, each communication round is associated with a computational cost C_t and an uplink communication cost of C_u, which leads to the following constraint

$$(C_t + MC_u)N \leq \overline{C}. \tag{35}$$

Given that the system satisfies constraint (35), we want to find the M that minimizes the FL loss after the final communication round N. We can formulate this as the following integer programming problem

$$\begin{aligned} &\min_{M,N} \mathbb{E}[F(\mathbf{w}_N)] \\ &\text{s.t.} \quad (C_t + MC_u)N \leq \overline{C} \\ &M, N \in \mathbb{Z}^+. \end{aligned} \tag{36}$$

Since $\mathbb{E}[F(\mathbf{w}_N)]$ is decreasing in N, the highest possible integer N that does not break constraint (35) can be selected without loss of optimality. Thus we can transform (36) to

$$\begin{aligned} &\min_{M} \mathbb{E}[F(\mathbf{w}_N)] \\ &\text{s.t.} \quad N = \left\lfloor \frac{\overline{C}}{C_t + MC_u} \right\rfloor \\ &M \in \mathbb{Z}^+, \end{aligned} \tag{37}$$

which is easier than (36). However, in order to solve this problem, we require an exact expression of $\mathbb{E}[F(\mathbf{w}_N)]$. In the current ML literature, no such expression is known. Instead, we propose to replace the objective function of (37) with a proxy function that resembles $\mathbb{E}[F(\mathbf{w}_N)]$.

A. Full bound proxy

Two possible proxy functions for $\mathbb{E}[F(\mathbf{w}_N)]$ are the upper bounds on $\mathbb{E}[F(\mathbf{w}_N)] - F(\mathbf{w}^*)$ in (31) (Proposition 2) and (33) (Proposition 3). There are two things to note about this choice.

- Initially, it might seem wrong to use $\mathbb{E}[F(\mathbf{w}_N)] - F(\mathbf{w}^*)$ as a proxy function for $\mathbb{E}[F(\mathbf{w}_N)]$, because the desired objective function is the expected loss, while our bounds are on the difference between the expected loss and the optimal loss. However, the optimal loss $F(\mathbf{w}^*)$ is constant with respect to M, so including it in the objective function has no impact on the optimal decision variable M^*.
- The upper bounds represent a worst-case analysis of the actual loss and we have no proof of tightness for these bounds. Therefore, this proxy function can at best be considered a heuristic for finding the optimal M and the efficacy of the choice should be evaluated numerically, which we do in Section VI.

B. Diminishing term proxy

The full bound proxy described above is the best approximation of $F(\mathbf{w}_N)$ we have available. However, to use it in optimization problem (37), rich knowledge about the dataset and loss function is required. Specifically, the Lipschitz smoothness constant L and the strong convexity parameter μ must be known to calculate the full bound. The calculation of these constants requires access to the global dataset, whereas the devices are only carrying smaller local datasets. Additionally, for more complicated models such as DNNs, the calculation of L is difficult and one can at best hope to estimate it [38]. As such, it is desirable to have a proxy function that can be calculated without the knowledge of μ and L.

In the strongly convex bound from Proposition 2 (31), the μ and L constants are ubiquitous, which makes the bound difficult to calculate in practice. However, in (33) from Proposition 3 there is of course no μ and the Lipschitz smoothness constant L only appears in the post-convergence term. As such, the diminishing term of (33) can be calculated using quite limited information. Additionally, we expect the diminishing term to be dominant since training over resource-constrained networks is generally stopped long before the algorithm has fully converged [37]. With this choice of proxy function, the optimization problem for finding M becomes

$$\begin{aligned} &\min_{M} \frac{K\sqrt{\eta}}{2N\beta \sum_{k=1}^{K} \sqrt{p_k}|h_k|} \mathbb{E}[r_0^2] \\ &\text{s.t.} \quad N = \left\lfloor \frac{\overline{C}}{C_t + MC_u} \right\rfloor \\ &M \in \mathbb{Z}^+. \end{aligned} \tag{38}$$

Here, every variable except $\mathbb{E}[r_0^2]$ is reasonable to know in practice. However, since $\mathbb{E}[r_0^2]$ is constant with respect to M it can be dropped without changing the optimal decision variable M^*. We then have the problem

$$\begin{aligned} &\min_{M} \frac{K\sqrt{\eta}}{2N\beta \sum_{k=1}^{K} \sqrt{p_k}|h_k|} \\ &\text{s.t.} \quad N = \left\lfloor \frac{\overline{C}}{C_t + MC_u} \right\rfloor \\ &M \in \mathbb{Z}^+. \end{aligned} \tag{39}$$

This final optimization problem is practical and computationally cheap to solve, however it is based on rough approximations of the desired problem (36). Therefore, the efficacy of the heuristic is not clear without experimental results, which we provide in Section VI.

VI. Numerical Results

The performance of the proposed AirReComp system is now evaluated in terms of the model update estimation error, federated learning loss, and classification accuracy. Specifically, we have three goals with this section:

- To demonstrate the need for retransmission-aware power control, by comparing our proposed solution with the state-of-the-art single transmission schemes proposed in [21], [23];

- To demonstrate that introducing retransmissions is also beneficial for non-convex loss functions. Note that our analytical results assumed convex loss functions.
- To demonstrate the viability of selecting M using our proposed heuristic.

A. Power Control

In this subsection, we wish to evaluate the impact of our proposed power control scheme on the estimation error of the global update $\Delta\tilde{w}_k$. Specifically, we compare the MSE of the $\Delta\tilde{w}_k$ for different choices of M, and compare the AirReComp power control scheme to the baseline solutions of [21], [23]. For this, we consider the transmission of randomly generated scalars instead of running a complete FL simulation setup. For this simulation, we consider $K = 20$ users and varying noise powers σ_z^2. To simulate the network, we generate channel coefficients according to unit Rayleigh fading $h_k \sim \mathcal{N}(0, 1/2) + j\mathcal{N}(0, 1/2)$ and additive noise components as $z \sim \mathcal{N}(0, \sigma_z^2)$. The transmitted scalars $\Delta\tilde{w}_k$ are generated according to the unit normal distribution, which matches the assumption in Section III. The PS estimate of the arithmetic mean $\Delta\hat{\tilde{w}}$ is generated according to (11), where the transmission powers p_k and the post-transmission scalar $\sqrt{\eta}$ are selected according to Proposition 1. Upon calculating the estimate, it is compared to the true arithmetic mean of $\Delta\tilde{w}_k$ according to

$$\text{MSE} = (\frac{1}{K}\sum_{k=1}^{K}\Delta\tilde{w}_k - \Delta\hat{\tilde{w}})^2. \tag{40}$$

This process is repeated 20,000 times for each value of σ_z^2. The resulting MSEs are averaged to form the plot in Fig. 2. In the left plot of the figure, the estimation error using different number of transmissions M are illustrated. The plot demonstrates that the mean squared estimation error is approximately linear with the variance of the additive noise, regardless of M. In the right plot of Fig. 2, we compare the AirReComp power control scheme to that proposed in [21], [23]. The baseline is optimal if there are no retransmissions, but the numerical results demonstrate that our scheme has a significantly lower estimation error when $M > 1$. The gap between AirReComp and the baseline is also increasing with M, which demonstrates the importance of designing the power control scheme with retransmissions in mind. From the left plot, it is clear that the reduction in estimation error is worse than proportional to M. Instead, the system using $M = 8$ achieves approximately three times lower estimation error that the baseline of $M = 1$. Compared to using a forward error-correcting code, this result is significantly worse. However, since such codes are not compatible with analog communication, retransmissions are a good first step towards enabling a communication-estimation trade-off.

B. Federated Learning Convergence

In this subsection, we have two goals: To verify that the convergence rate is increasing in M for non-convex loss functions and to show that $M > 1$ can achieve improved performance over $M = 1$ without exceeding the cost budget. For the FL simulation, the network setup is identical to Subsection VI-A, except that the additive noise variance is fixed to $\sigma_z^2 = \sqrt{2K}$. The ML task is multi-label classification on the MNIST dataset [39] with $|\mathcal{D}_k| = 6000$ training samples per user device. The classifier is a DNN which consists of an input layer of 784 nodes, a hidden layer with 100 nodes, and an output layer of 10 nodes. The network is trained with a static learning rate of $\beta = 0.05$ with no dropout. We run 2 epochs ($E = 2$) per communication round, with a budget of $\overline{C} = 150$, training cost $C_t = 4$ and communication cost $C_u = 1$. The whole training process is repeated 50 times for each value of M considered, these results are then averaged to get the plot in Fig. 3.

The results clearly demonstrate that the convergence rate is improved as additional retransmissions are introduced, even for non-convex loss functions. Notice that all systems stop training when the cost budget $\overline{C}$ is consumed, e.g. the system with $M = 1$ stops after 50 communication rounds and the system with $M = 2$ stops after 25 communication rounds. As such, we can see that AirReComp can outperform the baseline of using no retransmissions ($M = 1$) in terms of classification accuracy, without incurring additional costs. Specifically, the system with $M = 1$ achieved an average classification accuracy of 91.3%, while the best system with $M = 4$ achieved an average classification accuracy of 92.1%.

In addition to training a classifier on the MNIST dataset, we also train a regression model to predict the conductivity of freshwater based on its temperature, percentage of dissolved oxygen, pH value, oxidation-reduction potential, and salinity. We have 30,000 datapoints in the dataset, of which we use 25,000 randomly selected data points for training and the remaining 5,000 as test data. The dataset used is the iWater dataset from "Digital Demo Stockholm" which is collected by sensors installed in lake Mälaren, Sweden [40], [41]. For this simulation, we use exactly the same setup as the previous, except that the DNN is reshaped to fit the new dataset. There is now an input layer of 4 nodes and an output layer of 1 node. Once again, we repeat the whole training process 50 times for each value of M and the results are averaged to generate the plot in Fig. 4. Similarly to the MNIST simulation, we see that the convergence rate is improved with increasing M, but now for a regression problem. Once again, the system stops after the cost budget $\overline{C}$ is consumed. The optimal system (lowest MSE) with $M = 4$ has a normalized MSE of 0.0008, whereas the baseline of $M = 1$ achieved a normalized MSE of 0.001, so $M = 4$ achieves approximately 20% lower MSE than the baseline.

C. Selection of M

In Section V, we developed a heuristic for finding the M which would yield the lowest loss without exceeding a cost budget $\overline{C}$. To verify the efficacy of using this heuristic, we have compared the final losses of the simulations in Section VI-B to the objective function of (37), we refer to this objective function as the *bound heuristic*. In all plots of this section, we scale and translate the loss functions to give a better visual

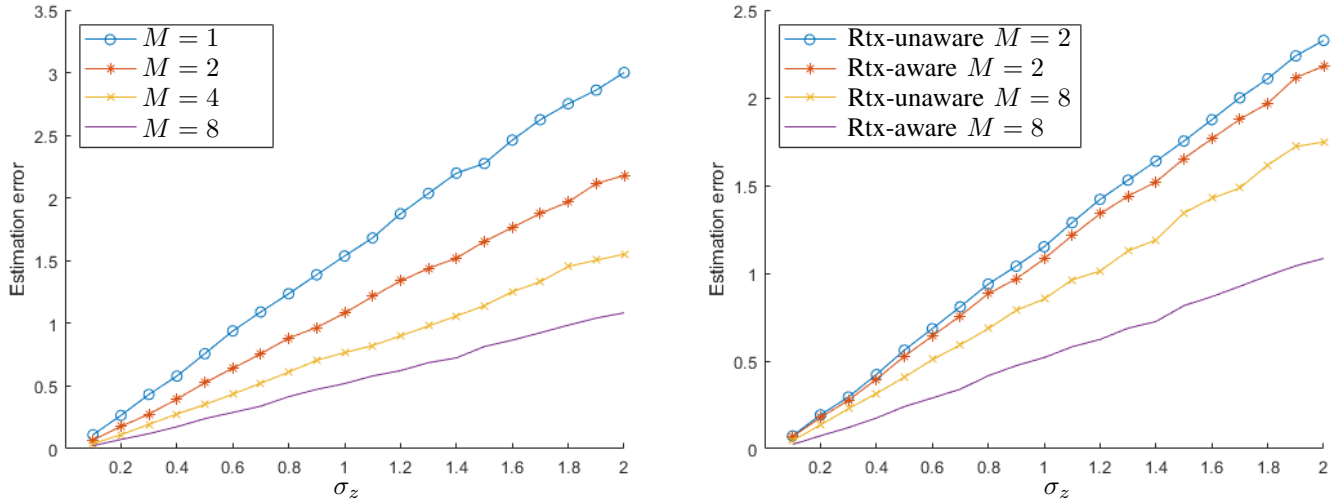

Fig. 2. Estimation error evaluation of AirReComp. We consider $K = 20$ devices and evaluate the squared estimation error. Left: The estimation error of a single transmission ($M = 1$) is compared to using retransmissions ($M > 1$). Note that even though SNR scales linearly with the number of transmissions, the estimation error is not reduced as drastically. Right: The estimation error of optimal retransmission-aware power control is compared to a retransmission-unaware baseline. The results demonstrate the importance of designing the power control scheme with retransmissions in mind.

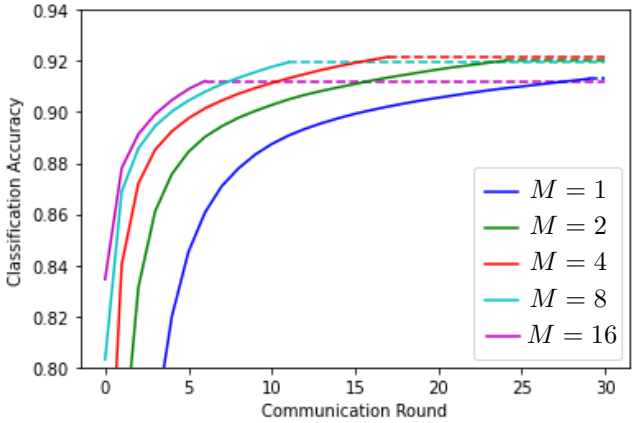

Fig. 3. Classification accuracy of MNIST hand-written digit recognition problem. We consider $K = 10$ devices and train fully-connected DNNs over a multiple access channel with fading. We compare five systems with different numbers of uplink retransmissions, where $M = 1$ corresponds to using zero retransmissions. The results demonstrate an increased rate of convergence as the number of retransmissions increase.

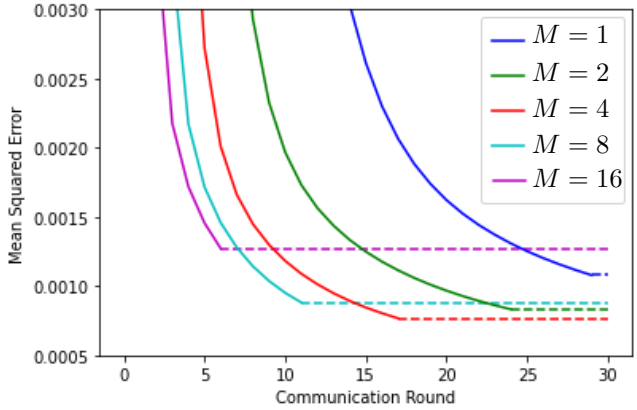

Fig. 4. Mean-squared error loss of "Digital Demo Stockholm" water monitoring problem. We train fully-connected DNNs with MSE loss to predict the conductivity level of the water based on other measurements (temperature, salinity, etc...). The plot shows the normalized MSE between the test dataset conductivity and the predicted conductivity as a function of the communication rounds. This result demonstrates that AirReComp does well for non-convex regression in addition to classification.

experience. Since we are only interested in the identification of the minimum value, this does not affect the result.

First for the MNIST dataset, we see in Fig. 5 that the shape of the bound heuristic resembles that of the empirical result. In this scenario, the heuristic would have selected $M = 4$, which also ended up being the optimal choice for training. To generate this plot, we multiplied the empirical loss by 10 and then subtracted a constant of 2. The bound heuristic line was left without scaling or translation.

For the iWater dataset, we have a similar comparison in Fig. 6. Both the bound and empirical loss function has a similar shape to Fig. 5, despite being for a completely different ML problem. However, the empirical loss function is of very different scale. In this figure, we multiplied the empirical loss by 1000, and did not perform any translation. The reason for this difference is because the regression problem is much easier than the classification problem and we therefore end up with lower losses in general. Since the shapes of the plots are the same, the bound heuristic correctly selects $M = 4$, which is the optimal number of retransmissions. This result illustrates that the bound heuristic works well for both classification and regression problems.

There are cases where one should not employ retransmissions, because the cost of doing so outweighs the benefit of reduced estimation errors. To illustrate this, we ran an MNIST simulation with the same setup as the previous MNIST simulation but reduced the cost of computation to $C_t = 0$, so that the the communication cost is relatively higher. Since the

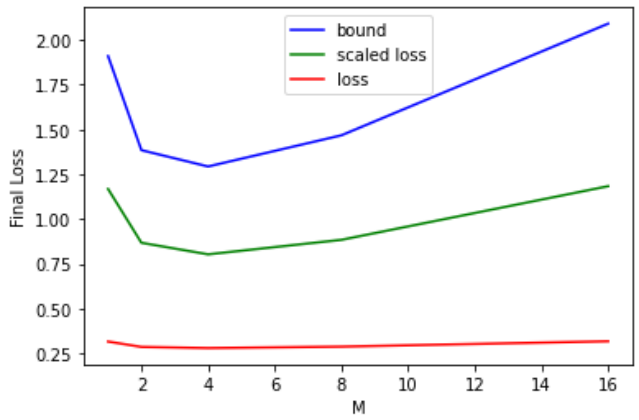

Fig. 5. Final loss after consuming the cost budget for the same systems as in Fig. 3. The empirical loss is compared to the diminishing term of our upper bound in (37). The heuristic matches the bound quite closely and selects the optimal M.

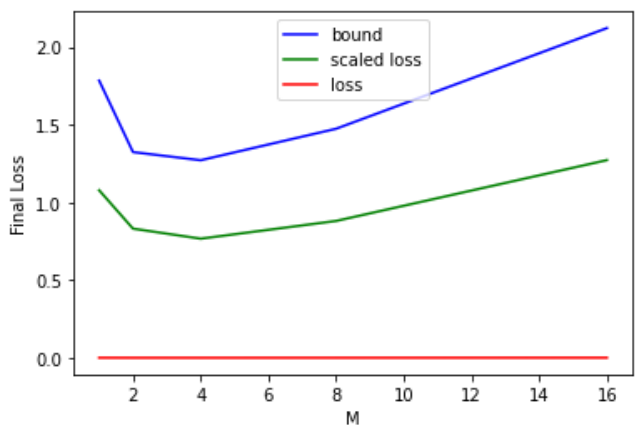

Fig. 6. Final loss after consuming the cost budget for a regression system for predicting the conductivity of freshwater. The heuristic selects $M = 4$ which is not the optimal $M^* = 8$. However, the difference in empirical performance between $M = 4$ and $M = 8$ is negligible, so $M = 4$ is not a poor choice.

computation cost is lower, we also reduced the total budget to $\overline{C} = 30$ so that the $M = 1$ system gets to train for the same number of rounds (30 communication rounds). The result is illustrated in Fig. 7. Here, we can see that the heuristic is able to correctly identify that retransmissions should not be used, since it would harm the final classification accuracy. Like the previous MNIST simulation, the empirical loss function is multiplied by 10 and then subtracted by 2 to generate the scaled loss.

Finally, we consider the relationship between the optimal number of transmissions M^* and the variance of the additive noise σ_z^2. To illustrate this, we calculate M^* using (39) for all $\sigma_z = \{0, 0.1, 0.2, ..., 9.9, 10\}$ to generate the plot in Fig. 8. In this simulation, we used the same parameters as in the simulation of Fig. 3 and as such we see $M^* = 4$ for $\sigma_z = \sqrt{2K} \approx 6.3$. This result demonstrates that the optimal number of retransmissions increases with the power of the noise, which also seems intuitively correct.

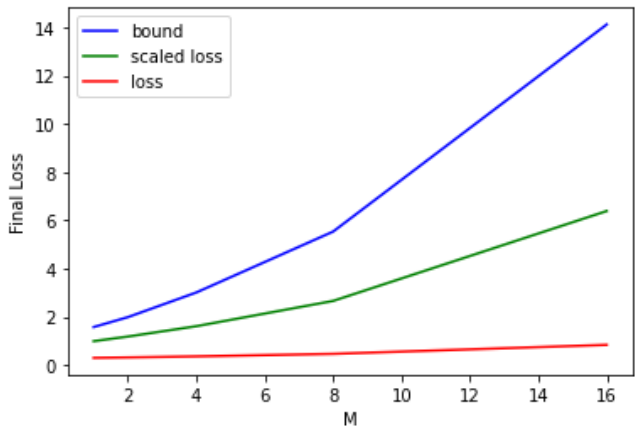

Fig. 7. Final loss after consuming the cost budget for a system where $M = 1$ is the optimal choice. The heuristic is able to identify $M^* = 1$ and can therefore avoid hurting the accuracy by introducing unwanted retransmissions.

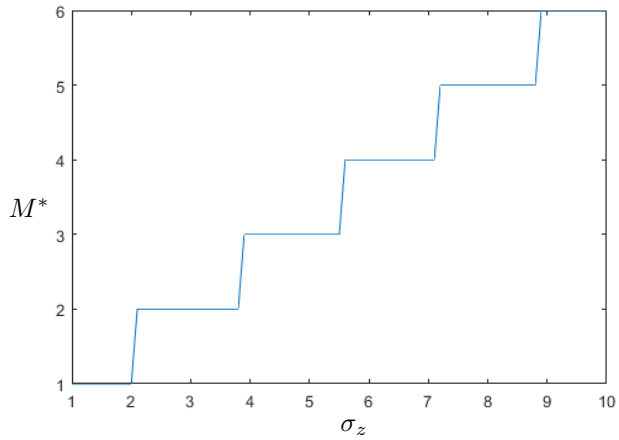

Fig. 8. Optimal M as predicted by our heuristic for different σ_z^2. We see that the optimal number of retransmissions are increasing with the additive noise power.

VII. Conclusion

In this paper, we propose retransmissions for Over-the-Air FL, in a system we call AirReComp. Arguably, this is the first work to enable a trade-off between communication resources and convergence speed for Over-the-Air FL. To improve the estimation error of AirReComp, we find a closed-form solution for optimal power control in the uplink. This power control solution shows that the number of retransmissions must be known by the transmitters to realize the MMSE estimator. We also prove two upper bounds on the FL loss for the AirReComp system, both for strongly-convex and convex loss functions. These bounds show that the convergence rate of FL is strictly increasing in the number of retransmissions. Finally, we provide a heuristic for selecting the optimal number of retransmissions based on the upper bounds. All of these results are verified numerically for non-convex loss functions by training DNNs with AirReComp, both in classification and regression tasks. The simulations also demonstrate that AirReComp can significantly outperform single uplink transmissions. There are conditions for which retransmissions are

not desirable, we demonstrate that the heuristic can detect such conditions before training starts, which avoids introducing harmful retransmissions.

There are still interesting open problems on the reduction of estimation errors for Over-the-Air FL, including:

- **The consideration of fast-fading channels for AirReComp.** In this work, we consider a network with static channels, which restricts the applicability of the results to application areas with static sensors. In a fast-fading scenario, the problem changes substantially, especially the power control problem described in Section III.
- **Other methods of controlling the estimation error for Over-the-Air FL.** For instance, one could consider the possibility of a distributed channel code. With analog communication, this appears to be inapplicable, but with the ideas of one-bit digital Over-the-Air Computation, there might be possibilities to explore in this direction [42].
- **Tradeoff between transmission power and retransmissions.** Instead of focusing on adding retransmissions to improve the estimation error, one could consider changing the transmission power. Similar to the tradeoff explored in this work, there is a tradeoff between transmission power and convergence rate. Especially for low-powered IoT-devices, it would be interesting to analyze how much the transmission power could be reduced without significantly harming the FL performance.

Appendix A
Proof of Proposition 2

In Section IV we assumed that the model updates $\Delta\mathbf{w}_n$ are already of zero mean and with unit variance. Therefore, no normalization is required. This can equivalently be expressed by saying that $\mu_{n,k} = 0$ and $\sigma_{n,k} = 1$. As a consequence of this assumption, we have

$$\Delta\hat{\mathbf{w}}_n = \mathrm{Re}(\overline{\mathbf{y}}) = \sum_{k=1}^{K} \frac{|h_k|\Delta\mathbf{w}_{n,k}\sqrt{p_k}}{\sqrt{\eta}K} + \sum_{m=1}^{M} \frac{\mathrm{Re}(\mathbf{z}_m)}{M\sqrt{\eta}K}. \quad (41)$$

We start the proof by expressing the distance between the optimal global model $\mathbf{w}^*$ and the current global model $\mathbf{w}_n$ at communication round n as

$$r_n^2 \coloneqq ||\mathbf{w}_n - \mathbf{w}^*||^2. \quad (42)$$

This distance can be related to the FL loss function via Lemma 1 and Lemma 2. The plan for the proof is to utilize this relationship to form the upper bound. But before we get to that stage, we need to introduce the impact of AirReComp on the model update. To do so, we use (14) with (42) to express

$$\begin{aligned} r_{n+1}^2 &\coloneqq ||\mathbf{w}_n - \mathbf{w}^* - \beta\Delta\hat{\mathbf{w}}_n||^2 \\ &= r_n^2 - 2\beta\Delta\hat{\mathbf{w}}_n^T(\mathbf{w}_n - \mathbf{w}^*) + \beta^2||\Delta\hat{\mathbf{w}}_n||^2, \end{aligned} \quad (43)$$

where $\Delta\hat{\mathbf{w}}_n$ is the model update from (41). Next, we take the expectation of (43) with respect to $\Delta\mathbf{w}_{n,k}$ and $\mathbf{z}_m$. To do that, we first need to determine $\mathbb{E}\left[\Delta\hat{\mathbf{w}}_n\right]$ and $\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right]$. Beginning with $\mathbb{E}\left[\Delta\hat{\mathbf{w}}_n\right]$, we use (41) to get

$$\mathbb{E}\left[\Delta\hat{\mathbf{w}}_n\right] = \mathbb{E}\left[\sum_{k=1}^{K} \frac{|h_k|\Delta\mathbf{w}_{n,k}\sqrt{p_k}}{\sqrt{\eta}K} + \sum_{m=1}^{M} \frac{\mathrm{Re}(\mathbf{z}_m)}{M\sqrt{\eta}K}\right]. \quad (44)$$

Because of Assumption 1 and since the noise $\mathbf{z}_m$ is zero-mean, this simplifies to:

$$\mathbb{E}\left[\Delta\hat{\mathbf{w}}_n\right] = \mathbb{E}[\Delta\mathbf{w}_n]\sum_{k=1}^{K} \frac{|h_k|\sqrt{p_k}}{\sqrt{\eta}K}. \quad (45)$$

Since we assume there is only one epoch ($E = 1$), the model update is the gradient of the global loss function, i.e., $\mathbb{E}[\Delta\mathbf{w}_n] = \mathbb{E}[\nabla F(\mathbf{w}_n)]$, which gives us

$$\mathbb{E}\left[\Delta\hat{\mathbf{w}}_n\right] = \mathbb{E}[\nabla F(\mathbf{w}_n)]\sum_{k=1}^{K} \frac{|h_k|\sqrt{p_k}}{\sqrt{\eta}K}. \quad (46)$$

Next, we find $\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right]$, once again using (41)

$$\begin{aligned} &\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right] = \\ &\frac{1}{K^2\eta}\mathbb{E}\left[||\sum_{k=1}^{K}|h_k|\Delta\mathbf{w}_{n,k}\sqrt{p_k} + \sum_{m=1}^{M}\frac{\mathrm{Re}(\mathbf{z}_m)}{M}||^2\right]. \end{aligned} \quad (47)$$

The cross-multiplication term disappears because the noise is zero mean and independent from the model update

$$\begin{aligned} \mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right] = &\frac{1}{K^2\eta}\mathbb{E}\left[||\sum_{k=1}^{K}\sqrt{p_k}|h_k|\Delta\mathbf{w}_{n,k}||^2\right] \\ &+\frac{1}{M^2K^2\eta}\mathbb{E}\left[||\sum_{m=1}^{M}\mathrm{Re}(\mathbf{z}_m)||^2\right]. \end{aligned} \quad (48)$$

The first term of (48) can be upper-bounded by the Cauchy-Schwartz inequality as follows

$$||\sum_{k=1}^{K}\sqrt{p_k}|h_k|\Delta\mathbf{w}_{n,k}||^2 \leq K\sum_{i=1}^{d}\left(\sum_{k=1}^{K}p_k|h_k|^2(\Delta w_{n,k}^{(i)})^2\right). \quad (49)$$

Then, we apply our assumption on the local model updates from (25) to (49) to get

$$||\sum_{k=1}^{K}\sqrt{p_k}|h_k|\Delta\mathbf{w}_{n,k}||^2 \leq K\sum_{k=1}^{K}p_k|h_k|^2(||\Delta\mathbf{w}_n||^2 + ||\boldsymbol{\sigma}||^2). \quad (50)$$

Combining (48) and (50) gives us

$$\begin{aligned} &\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right] \leq \\ &\frac{1}{K\eta}\mathbb{E}\left[\sum_{k=1}^{K}p_k|h_k|^2(||\Delta\mathbf{w}_n||^2 + ||\boldsymbol{\sigma}||^2)\right] + \frac{d\sigma_z^2}{MK^2\eta}, \end{aligned} \quad (51)$$

and since we assume $E = 1$, we get

$$\begin{aligned} &\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right] \leq \\ &\frac{1}{K\eta}\sum_{k=1}^{K}p_k|h_k|^2(||\boldsymbol{\sigma}||^2 + \mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right]) + \frac{d\sigma_z^2}{MK^2\eta}. \end{aligned} \quad (52)$$

With $\mathbb{E}\left[\Delta\hat{\mathbf{w}}_n\right]$ and $\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right]$ evaluated in (46) and (52), we go back to the model distance. Taking the expectation on both sides of (43) yields

$$\mathbb{E}[r_{n+1}^2] \leq \mathbb{E}[r_n^2] - \sum_{k=1}^{K} \frac{2\beta\sqrt{p_k}|h_k|\mathbb{E}[\nabla F(\mathbf{w}_n)^T(\mathbf{w}_n - \mathbf{w}^*)]}{K\sqrt{\eta}} + \frac{\beta^2}{K\eta}\sum_{k=1}^{K} p_k|h_k|^2(||\boldsymbol{\sigma}||^2 + \mathbb{E}[||\nabla F(\mathbf{w}_n)||^2]) + \frac{d\sigma_z^2\beta^2}{MK^2\eta}. \tag{53}$$

Now we are ready to introduce the FL loss by utilizing strong convexity and Lipschitz smoothness. We do this by rewriting Lemma 2 to

$$\mathbb{E}\left[\nabla F(\mathbf{w}_n)^T(\mathbf{w}_n - \mathbf{w}^*)\right] \geq \frac{\mu L}{\mu + L}\mathbb{E}[r_n^2] + \frac{1}{\mu + L}\mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right], \tag{54}$$

where we have utilized $\nabla F(\mathbf{w}^*) = 0$ for the final term on the RHS. Combining (53) and (54) yields

$$\mathbb{E}[r_{n+1}^2] \leq \mathbb{E}[r_n^2] - \frac{2\beta}{K\sqrt{\eta}}\sum_{k=1}^{K}\sqrt{p_k}|h_k|\left(\frac{\mu L}{\mu + L}\mathbb{E}[r_n^2] + \frac{1}{\mu + L}\mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right]\right) + \frac{\beta^2}{K\eta}\sum_{k=1}^{K} p_k|h_k|^2(||\boldsymbol{\sigma}||^2 + \mathbb{E}[||\nabla F(\mathbf{w}_n)||^2]) + \frac{d\sigma_z^2\beta^2}{MK^2\eta}. \tag{55}$$

Since this expression is getting long, we use three constants c_2, c_3, and c_4 to simplify it. The model distance is then

$$\mathbb{E}[r_{n+1}^2] \leq c_2\mathbb{E}[r_n^2] + c_3 + c_4\mathbb{E}[||\nabla F(\mathbf{w}_n)||^2], \tag{56}$$

where c_2 and c_3 were defined in (29) and (30) respectively. Because of our choice of learning rate in Assumption 2, we have the following inequality for c_4

$$c_4 \coloneqq \frac{\beta^2}{K\eta}\sum_{k=1}^{K} p_k|h_k|^2 - \frac{2\beta}{K\sqrt{\eta}(\mu + L)}\sum_{k=1}^{K}\sqrt{p_k}|h_k| < 0. \tag{57}$$

Since c_4 is less than zero, we can rewrite our bound in (56) as

$$\mathbb{E}[r_{n+1}^2] \leq c_2\mathbb{E}[r_n^2] + \beta^2 c_3. \tag{58}$$

At this point, the bound is almost complete. The only thing that remains is to find an inequality comparing $\mathbb{E}[r_n^2]$ and $\mathbb{E}[r_0^2]$ instead of comparing two adjacent communication rounds. As such, we reduce the iteration counter by one, which yields

$$\mathbb{E}[r_n^2] \leq c_2\mathbb{E}[r_{n-1}^2] + \beta^2 c_3. \tag{59}$$

We then combine (58) and (59) to get

$$\mathbb{E}[r_{n+1}^2] \leq c_2^2\mathbb{E}[r_{n-1}^2] + (c_2 + 1)\beta^2 c_3. \tag{60}$$

By induction we have

$$\mathbb{E}[r_n^2] \leq c_2^n\mathbb{E}[r_0^2] + \beta^2 c_3\sum_{i=0}^{n-1} c_2^i. \tag{61}$$

Then we apply $\sum_{i=0}^{n-1} c_2^i < \sum_{i=0}^{\infty} c_2^i = 1/(1 - c_2)$ to achieve

$$\mathbb{E}[r_n^2] \leq c_2^n\mathbb{E}[r_0^2] + \frac{\beta^2}{(1 - c_2)}c_3. \tag{62}$$

Finally, we utilize convexity and Lipschitz smoothness from (23) to relate the LHS of (62) to the FL loss, which yields

$$\mathbb{E}\left[F(\mathbf{w}_n)\right] - F(\mathbf{w}^*) \leq \frac{L}{2}c_2^n\mathbb{E}[r_0^2] + \frac{\beta^2 L}{2(1 - c_2)}c_3, \tag{63}$$

which is the bound from Proposition 2. □

Remark 4. *The c_2 variable determines the convergence rate of the algorithm and must fulfill $c_2 < 1$ to guarantee convergence in expectation. Given that the static learning rate β follows Assumption 2, this inequality will always hold. It is easily proven by substituting the β from* (27) *into*

$$c_2 \coloneqq 1 - \frac{2\beta}{K\sqrt{\eta}}\frac{\mu L}{\mu + L}\sum_{k=1}^{K}\sqrt{p_k}|h_k|. \tag{64}$$

Appendix B
Proof of Proposition 3

Just as in the first proof, we utilize the properties of convexity and Lipschitz smoothness to relate distance between the optimal global model $\mathbf{w}^*$ and the current global model $\mathbf{w}_n$ to the FL loss function. In contrast to the first proof, we use these properties immediately. Specifically, we start with Lemma 1 and take the expectation on both sides to get

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq \mathbb{E}\left[F(\mathbf{w}_n)\right] + \mathbb{E}\left[\nabla F(\mathbf{w}_n)^T(\mathbf{w}_{n+1} - \mathbf{w}_n)\right] + \frac{L}{2}\mathbb{E}\left[||\mathbf{w}_{n+1} - \mathbf{w}_n||^2\right]. \tag{65}$$

Then, we add the global update via (14) to get

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq \mathbb{E}\left[F(\mathbf{w}_n)\right] - \beta\mathbb{E}\left[\nabla F(\mathbf{w}_n)^T\Delta\hat{\mathbf{w}}_n\right] + \frac{L\beta^2}{2}\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right]. \tag{66}$$

The impact of the wireless channel is introduced in (41). Then, following steps similar to (44)-(46) gives us

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq \mathbb{E}\left[F(\mathbf{w}_n)\right] - \frac{\beta}{K\sqrt{\eta}}\sum_{k=1}^{K}\sqrt{p_k}|h_k|\mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right] + \frac{L\beta^2}{2}\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right]. \tag{67}$$

Then, we insert the bound for $\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_n||^2\right]$ from (52) to get

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq \mathbb{E}\left[F(\mathbf{w}_n)\right] - \frac{\beta}{K\sqrt{\eta}}\sum_{k=1}^{K}\sqrt{p_k}|h_k|\mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right] + \frac{L\beta^2}{2K\eta}\left(\sum_{k=1}^{K} p_k|h_k|^2(||\boldsymbol{\sigma}||^2 + \mathbb{E}[||\nabla F(\mathbf{w}_n)||^2]) + \frac{d\sigma_z^2}{MK}\right). \tag{68}$$

Next, we recognize c_3 from (30) and substitute it into the bound

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq \mathbb{E}\left[F(\mathbf{w}_n)\right] - \frac{\beta}{K\sqrt{\eta}}\sum_{k=1}^{K}\sqrt{p_k}|h_k|\left(1-\frac{L\beta}{2\sqrt{\eta}}\frac{\sum_{k=1}^{K}p_k|h_k|^2}{\sum_{k=1}^{K}\sqrt{p_k}|h_k|}\right) \mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right] + \frac{L\beta^2}{2}c_3. \tag{69}$$

Then we do the same for c_1

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq \mathbb{E}\left[F(\mathbf{w}_n)\right] - \frac{\beta c_1}{K}\left(1-\frac{L\beta}{2\sqrt{\eta}}\frac{\sum_{k=1}^{K}p_k|h_k|^2}{\sum_{k=1}^{K}\sqrt{p_k}|h_k|}\right)\mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right] + \frac{L\beta^2}{2}c_3. \tag{70}$$

This expression can be simplified by using Assumption 3 to bound the second term on the RHS

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq \mathbb{E}\left[F(\mathbf{w}_n)\right] - \frac{\beta c_1}{2K}\mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right] + \frac{L\beta^2}{2}c_3. \tag{71}$$

Next, we are going to upper bound the first term on the RHS of (71). We use the following standard property of convexity (see equation 3.2 from [43]):

$$\mathbb{E}[F(\mathbf{w}_n)] \leq F(\mathbf{w}^*) + \mathbb{E}[\nabla F(\mathbf{w}_n)^T(\mathbf{w}_n - \mathbf{w}^*)]. \tag{72}$$

Plug this into (71)

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq F(\mathbf{w}^*) + \mathbb{E}[\nabla F(\mathbf{w}_n)^T(\mathbf{w}_n - \mathbf{w}^*)] - \frac{\beta c_1}{2K}\mathbb{E}\left[||\nabla F(\mathbf{w}_n)||^2\right] + \frac{L\beta^2}{2}c_3. \tag{73}$$

Some tedious algebraic manipulation transforms (73) to

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq F(\mathbf{w}^*) + \frac{K}{2\beta c_1}\mathbb{E}\left[||\mathbf{w}_n - \mathbf{w}^*||^2 - ||(\mathbf{w}_n - \mathbf{w}^*) - \frac{\beta c_1}{K}\nabla F(\mathbf{w}_n)||^2\right] + \frac{L\beta^2}{2}c_3. \tag{74}$$

Let $r_n^2 \coloneqq ||\mathbf{w}_n - \mathbf{w}^*||^2$ (same as (42)) and

$$\tilde{r}_{n+1}^2 \coloneqq ||\mathbf{w}_n - \mathbf{w}^* - \frac{\beta c_1}{K}\nabla F(\mathbf{w}_n)||^2. \tag{75}$$

Then (74) becomes

$$\mathbb{E}\left[F(\mathbf{w}_{n+1})\right] \leq F(\mathbf{w}^*) + \frac{K}{2\beta c_1}\mathbb{E}\left[r_n^2 - \tilde{r}_{n+1}^2\right] + \frac{L\beta^2}{2}c_3. \tag{76}$$

Now, just like in the previous proof, we want to form a bound with respect to r_0^2. However, instead of using induction we use a telescoping sum. To set it up, we start by taking a sum of (76) over n iterations to get

$$\sum_{i=1}^{n}\mathbb{E}\left[F(\mathbf{w}_i)\right] - nF(\mathbf{w}^*) \leq + \frac{K}{2\beta c_1}\sum_{i=1}^{n}\mathbb{E}\left[r_{i-1}^2 - \tilde{r}_i^2\right] + \frac{nL\beta^2}{2}c_3. \tag{77}$$

The sum $\sum_{i=1}^{n}\mathbb{E}\left[r_{i-1}^2 - \tilde{r}_i^2\right]$ can be rewritten as

$$\sum_{i=1}^{n}\mathbb{E}\left[r_{i-1}^2 - \tilde{r}_i^2\right] = \mathbb{E}[r_0^2] - \mathbb{E}[\tilde{r}_n^2] + \sum_{i=0}^{n-2}\mathbb{E}[r_{i+1}^2 - \tilde{r}_{i+1}^2] \tag{78}$$

and the middle terms $\sum_{i=0}^{n-2}\mathbb{E}[r_{i+1}^2 - \tilde{r}_{i+1}^2]$ will be upper bounded to a constant. We develop this bound next. To start, we plug in the definition of r_{i+1}^2 and $\tilde{r}_{i+1}^2$ into $\mathbb{E}[r_{i+1}^2 - \tilde{r}_{i+1}^2]$:

$$\mathbb{E}\left[r_{i+1}^2 - \tilde{r}_{i+1}^2\right] = \mathbb{E}\left[||\mathbf{w}_{i+1} - \mathbf{w}^*||^2 - ||\mathbf{w}_i - \mathbf{w}^* - \frac{\beta c_1}{K}\nabla F(\mathbf{w}_i)||^2\right]. \tag{79}$$

Applying (14) and doing some algebra yields

$$\mathbb{E}\left[r_{i+1}^2 - \tilde{r}_{i+1}^2\right] = \beta\mathbb{E}\left[\beta||\Delta\hat{\mathbf{w}}_i||^2 - 2(\mathbf{w}_i - \mathbf{w}^*)^T\Delta\hat{\mathbf{w}}_i\right] + \beta\mathbb{E}\left[\frac{2c_1}{K}(\mathbf{w}_i - \mathbf{w}^*)^T\nabla F(\mathbf{w}_i) - \frac{\beta c_1^2}{K^2}||\nabla F(\mathbf{w}_i)||^2\right]. \tag{80}$$

Then we apply (41) and after some algebra we have

$$\mathbb{E}\left[r_{i+1}^2 - \tilde{r}_{i+1}^2\right] = \beta^2\mathbb{E}\left[||\Delta\hat{\mathbf{w}}_i||^2 - \frac{c_1^2}{K^2}||\nabla F(\mathbf{w}_i)||^2\right]. \tag{81}$$

Next, we insert $||\Delta\hat{\mathbf{w}}_i||^2$ from (52) and do some algebra which yields

$$\mathbb{E}\left[r_{i+1}^2 - \tilde{r}_{i+1}^2\right] = \beta^2\mathbb{E}\left[\frac{1}{K^2\eta}\left(\sum_{k=1}^{K}\sqrt{p_k}|h_k|\right)^2||\Delta\mathbf{w}_{n,k}||^2 + \frac{d\sigma_z^2}{MK^2\eta} - \frac{c_1^2}{K^2}||\nabla F(\mathbf{w}_i)||^2\right]. \tag{82}$$

Assumption 1 together with some algebra gives us

$$\mathbb{E}\left[r_{i+1}^2 - \tilde{r}_{i+1}^2\right] \leq \beta^2\frac{c_1^2}{K^2}||\boldsymbol{\sigma}||^2 + \beta^2\frac{d\sigma_z^2}{MK^2\eta}. \tag{83}$$

Finally, we apply the Cauchy-Schwarz inequality to get

$$\mathbb{E}\left[r_{i+1}^2 - \tilde{r}_{i+1}^2\right] \leq \beta^2||\boldsymbol{\sigma}||^2\frac{\sum_{k=1}^{K}p_k|h_k|^2}{K\eta} + \beta^2\frac{d\sigma_z^2}{MK^2\eta} = \beta^2 c_3 \tag{84}$$

That concludes the upper bound on $\mathbb{E}\left[r_{i+1}^2 - \tilde{r}_{i+1}^2\right]$ so we plug it back into (77) to get

$$\sum_{i=1}^{n}\mathbb{E}\left[F(\mathbf{w}_i)\right] - nF(\mathbf{w}^*) \leq \frac{K}{2\beta c_1}\mathbb{E}\left[r_0^2 - \tilde{r}_n^2 + (n-1)\beta^2 c_3\right] + \frac{nL\beta^2}{2}c_3. \tag{85}$$

Since $\tilde{r}_n^2$ and c_3 are positive, we can add one of each to the expectation in the RHS of (85) without breaking the inequality, which yields

$$\sum_{i=1}^{n}\mathbb{E}\left[F(\mathbf{w}_i)\right] - nF(\mathbf{w}^*) \leq \frac{K}{2\beta c_1}\mathbb{E}\left[r_0^2\right] + \frac{n\beta}{2}\left(\frac{K}{c_1} + L\beta\right)c_3. \tag{86}$$

Finally, we note that $\mathbb{E}\left[F(\mathbf{w}_n)\right] \leq \mathbb{E}\left[F(\mathbf{w}_i)\right]$ for all $i \leq n$. Therefore

$$\mathbb{E}\left[F(\mathbf{w}_n)\right] - F(\mathbf{w}^*) \leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[F(\mathbf{w}_i)\right] - F(\mathbf{w}^*) \leq \frac{K}{2n\beta c_1}\mathbb{E}\left[r_0^2\right] + \frac{\beta}{2}\left(\frac{K}{c_1} + L\beta\right)c_3, \tag{87}$$

which is the bound from Proposition 3. □

References

[1] "Ericsson IoT Connections Outlook: Broadband IoT set to overtake 2G and 3G," https://www.ericsson.com/en/mobility-report/dataforecasts/iot-connections-outlook, accessed: 2021-08-24.

[2] "Pew Research Center: U.S. Technology Device Ownership 2015," https://www.pewresearch.org/internet/2015/10/29/technology-device-ownership-2015/, accessed: 2021-08-24.

[3] J. Jagannathan and U. Udaykumar, "Predictive Modeling for Improving Healthcare Using IoT: Role of Predictive Models in Healthcare Using IoT," in *Incorporating the Internet of Things in Healthcare Applications and Wearable Devices*. IGI Global, 2020, pp. 243–254.

[4] B. Brik, A. Ksentini, and M. Bouaziz, "Federated Learning for UAVs-enabled Wireless Networks: Use Cases, Challenges, and Open Problems," *IEEE Access*, vol. 8, pp. 53 841–53 849, 2020.

[5] L. Knoll, L. Breuer, and M. Bach, "Large Scale Prediction of Groundwater Nitrate Concentrations from Spatial Data Using Machine Learning," *Science of the total environment*, vol. 668, pp. 1317–1327, 2019.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and Open Problems in Federated Learning," *arXiv preprint arXiv:1912.04977*, 2019.

[8] 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*, 2017.

[9] H. Hellström, J. M. B. da Silva Jr, V. Fodor, and C. Fischione, "Wireless for Machine Learning," *arXiv preprint arXiv:2008.13492*, 2020.

[10] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an Intelligent Edge: Wireless Communication Meets Machine Learning," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 19–25, 2020.

[11] M. M. Amiri and D. Gündüz, "Machine Learning at the Wireless Edge: Distributed Stochastic Gradient Descent Over-the-Air," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[12] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A Joint Learning and Communications Framework for Federated Learning over Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

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

[14] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated Learning via Over-the-Air Computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[15] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing Interference for Analog Function Computation in Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 20, pp. 4893–4906, 2013.

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

[17] B. Hasırcıoğlu and D. Gündüz, "Private Wireless Federated Learning with Anonymous Over-the-Air Computation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5195–5199.

[18] 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.

[19] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[20] R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.

[21] X. Cao, G. Zhu, J. Xu, and K. Huang, "Optimized Power Control for Over-the-Air Computation in Fading Channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7498–7513, 2020.

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

[23] W. Liu, X. Zang, Y. Li, and B. Vucetic, "Over-the-Air Computation Systems: Optimization, Analysis and Scaling Laws," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5488–5502, 2020.

[24] X. Zang, W. Liu, Y. Li, and B. Vucetic, "Over-the-Air Computation Systems: Optimal Design with Sum-Power Constraint," *IEEE Wireless Communications Letters*, vol. 9, no. 9, pp. 1524–1528, 2020.

[25] X. Li, G. Zhu, Y. Gong, and K. Huang, "Wirelessly Powered Data Aggregation for IoT via Over-the-Air Function Computation: Beamforming and Power Control," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3437–3452, 2019.

[26] N. Zhang and M. Tao, "Gradient Statistics Aware Power Control for Over-the-Air Federated Learning," *IEEE Transactions on Wireless Communications*, 2021.

[27] D. Fan, X. Yuan, and Y.-J. A. Zhang, "Temporal-Structure-Assisted Gradient Aggregation for Over-the-Air Federated Edge Learning," *arXiv preprint arXiv:2103.02270*, 2021.

[28] T. Jiang and Y. Shi, "Over-the-air computation via intelligent reflecting surfaces," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[29] Z. Wang, Y. Shi, Y. Zhou, H. Zhou, and N. Zhang, "Wireless-powered over-the-air computation in intelligent reflecting surface-aided IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1585–1598, 2020.

[30] Z. Wang, J. Qiu, Y. Zhou, Y. Shi, L. Fu, W. Chen, and K. B. Letaief, "Federated learning via intelligent reflecting surface," *IEEE Transactions on Wireless Communications*, vol. Early Access, July 2021.

[31] A. J. Goldsmith and S.-G. Chua, "Adaptive Coded Modulation for Fading Channels," *IEEE Transactions on communications*, vol. 46, no. 5, pp. 595–602, 1998.

[32] G. Zhu, Y. Wang, and K. Huang, "Broadband Analog Aggregation for Low-Latency Federated Edge Learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2019.

[33] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated Learning over Wireless Networks: Optimization Model Design and Analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1387–1395.

[34] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2003, vol. 87.

[35] X. Cao, G. Zhu, J. Xu, and S. Cui, "Optimized Power Control for Over-the-Air Federated Edge Learning," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[36] H. Hellström, V. Fodor, and C. Fischione, "Over-the-Air Federated Learning with Retransmissions," in *2021 IEEE 22st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2021.

[37] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When Edge Meets Learning: Adaptive Control for Resource-constrained Distributed Machine Learning," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 63–71.

[38] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. J. Pappas, "Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks," *arXiv preprint arXiv:1906.04893*, 2019.

[39] 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.

[40] "iWater: Digital Demo Stockholm," https://www.digitaldemostockholm.com/en/demo-projects/iwater/, accessed: 2021-10-29.

[41] D. Maher, "IoT for Fresh Water Quality Monitoring," *DiVA: Digitala Vetenskapliga Arkivet*, 2018.

[42] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit Over-the-Air Aggregation for Communication-Efficient Federated Edge Learning: Design and Convergence Analysis," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 2120–2135, 2020.

[43] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.