# Layerwise and Dimensionwise Local Adaptive Method for Federated Learning

**Anonymous Authors**[1]

## Abstract

In the emerging paradigm of Federated Learning (FL), large amount of clients, such as mobile devices, are used to train possibly high-dimensional models on their respective data. Under the orchestration of a central server, the data needs to remain decentralized, as it can not be shared among clients or with the central server. Then, due to the low bandwidth of mobile devices, decentralized optimization methods need to shift the computation burden from those clients to the computation server while preserving *privacy* and reasonable *communication cost*. In the particular case of training Deep, as in multilayered Neural Networks, under such settings, we propose in this paper, FED-LAMB, a novel Federated Learning method based on a Layerwise and Dimensionwise updates of the local models. A periodic averaging is added to obtain estimates of the desired global model parameters. We provide a thorough finite time convergence analysis for our algorithm, substantiated by numerical runs on benchmark datasets.

## 1. Introduction

A growing and important task while learning models on observed data, is the ability to train the latter over a large number of clients which could either be devices or distinct entities. In the paradigm of Federated Learning (FL) (Konečnỳ et al., 2016; McMahan et al., 2017), the focus of our paper, a central server orchestrates the optimization over those clients under the constraint that the data can neither be centralized nor shared among the clients. Most modern machine learning tasks can be casted as a large finite-sum optimization problem written as:

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \tag{1}$$

where $n$ denotes the number of workers, $f_i$ represents the average loss for worker $i$ and $\theta$ the global model parameter taking value in $\Theta$ a subset of $\mathbb{R}^d$. While this formulation recalls that of distributed optimization, the core principle of FL is different that standard distributed paradigm.

FL currently suffers from two bottlenecks: communication efficiency and privacy. We focus on the former in this paper. While local updates, updates during which each client learn their local models, can reduce drastically the number of communication rounds between the central server and devices, new techniques must be employed to tackle this challenge. Some quantization (Alistarh et al., 2017; Wangni et al., 2018) or compression (Lin et al., 2017) methods allow to decrease the number of bits communicated at each round and are efficient method in a distributed setting. The other approach one can take is to accelerate the local training on each device and thus sending a better local model to the server at each round.

Under the important setting of heterogenous data, i.e. the data among each device can be distributed according to different distributions, current local optimization algorithms are perfectible. The most popular method for FL is using multiple local Stochastic Gradient Descent (SGD) steps in each device, sending those local models to the server that computes the average over those received local vector of parameters and broadcasts it back to the devices. This is called FEDAVG and has been introduced in (McMahan et al., 2017).

In (Chen et al., 2020), the authors motivate the usage of adaptive gradient optimization methods as a better alternative to the standard SGD inner loop in FEDAVG. They propose an adaptive gradient method, namely LOCAL AMSGRAD, with communication cost sublinear in $T$ that is guaranteed to converge to stationary points in $\mathcal{O}(\sqrt{d/Tn})$, where T is the number of iterations.

Based on recent progress in adaptive methods for accelerating the training procedure, see (You et al., 2019), we propose

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

a variant of LOCAL AMSGRAD integrating dimensionwise and layerwise adaptive learning rate in each device's local update. Our contributions are as follows:

- We develop a novel optimization algorithm for federated learning, namely FED-LAMB, following a principled layerwise adaptation strategy to accelerate training of deep neural networks.
- We provide a rigorous theoretical understanding of the non asymptotic convergence rate of FED-LAMB. Based on the recent progress on nonconvex stochastic optimization, we derive for a any finite number of rounds performed by our method, a characterization of the rate at which the classical suboptimilality condition, *i.e.,* , the second order moment of the gradient of the objective function, decreases. Our bound in $\mathcal{O}\left(\sqrt{\frac{p}{n}}\frac{1}{\mathsf{L}R}\right)$ matches state of the art methods in Federated Learning reaching a sublinear convergence in $R$, the total number of rounds.
- We exhibit the advantages of our method on several benchmarks supervised learning methods on both homogeneous and heterogeneous settings.

The plan of our paper is as follows. After having established a literature review of both realms of federated and adaptive learning in subsection 1.1, we develop in Section 2, our method, namely FED-LAMB, based on the computation per layer and per dimension, of a scaling factor in the traditional stepsize of AMSGrad. Theoretical understanding of our method's behaviour with respect to convergence towards a stationary point is developed in Section 3. We present numerical illustrations showing the advantages of our method in Section 4.

## 1.1. Related Work

**Adaptive gradient methods.** In classical stochastic nonconvex optimization, adaptive methods have proven to be the spearhead in many applications. Those gradient based optimization algorithms alleviate the possibly high nonconvexity of the objective function by adaptively updating each coordinate of their learning rate using past gradients. Most used examples AMSGRAD (Reddi et al., 2018), ADAM (Kingma & Ba, 2015), RMSPROP (Tieleman & Hinton, 2012), ADADELTA (Zeiler, 2012), and NADAM (Dozat, 2016).

Their popularity and efficiency are due to their great performance at training deep neural networks. They generally combine the idea of adaptivity from ADAGRAD (Duchi et al., 2011; McMahan & Streeter, 2010), as explained above, and the idea of momentum from NESTEROV'S METHOD (Nesterov, 2004) or HEAVY BALL method (Polyak, 1964) using past gradients. ADAGRAD displays a great edge when the gradient is sparse compared to other classical methods. Its update has a notable feature: it leverages an anisotropic learning rate depending on the magnitude of the gradient for each dimension which helps in exploiting the geometry of the data.

The anisotropic nature of this update represented a real breakthrough in the training of high dimensional and nonconvex loss functions. This adaptive learning rate helps accelerating the convergence when the gradient vector is sparse (Duchi et al., 2011), yet, when applying ADAGRAD to train deep neural networks, it is observed that the learning rate might decay too fast, see (Kingma & Ba, 2015) for more details. Consequently, (Kingma & Ba, 2015) develops ADAM leveraging a moving average of the gradients divided by the square root of the second moment of this moving average (element-wise multiplication). A variant, called AMSGRAD described in (Reddi et al., 2018) ought to fix ADAM failures and is presented in Algorithm 1.

**Algorithm 1** AMSGRAD (Reddi et al., 2018)

```
1: Required: parameter $\beta_1$, $\beta_2$, and $\eta_t$.
2: Init: $w_1 \in \Theta \subseteq \mathbb{R}^d$ and $v_0 = \epsilon 1 \in \mathbb{R}^d$.
3: for $t = 1$ to $T$ do
4:    Get mini-batch stochastic gradient $g_t$ at $w_t$.
5:    $\theta_t = \beta_1 \theta_{t-1} + (1 - \beta_1) g_t$.
6:    $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$.
7:    $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$.
8:    $w_{t+1} = w_t - \eta_t \frac{\theta_t}{\sqrt{\hat{v}_t}}$. (element-wise division)
9: end for
```

The difference between ADAM and AMSGRAD lies in Line 1 of Algorithm 1.

A natural extension of Algorithm 1 has been developed in (You et al., 2019) specifically for multi layered neural network. A principled layerwise adaptation strategy to accelerate training of deep neural networks using large mini-batches is proposed using either a standard stochastic gradient update or a generalized adaptive method under the setting of a classical single server empirical risk minimization problem.

**Federated learning.** An extension of the well known parameter server framework, where a model is being trained on several servers in a distributed manner, is called Federated Learning, see (Konečnỳ et al., 2016). Here, the central server only plays the role of compute power for aggregation and global update of the model. Compared with the distributed learning paradigm, in Federated Learning, the data stored in each worker must not be seen by the central server – preserving privacy is key – and the nature of those workers, which can be mobile devices, combined with their usually large amount make communication between the devices and the central server less appealing – communication cost needs to be controlled.

Thus, while traditional distributed gradient methods (Recht et al., 2011; Li et al., 2014; Zhao et al., 2020) do not respect those constraints, it has been proposed in (McMahan et al., 2017), an algorithm called Federated Averaging – FED-AVG – extending parallel SGD with local updates performed on each device. In FED-AVG, each worker updates their own model parameters locally using SGD, and the local models are synchronized by periodic averaging on the central parameter server.

## 2. Layerwise and Dimensionwise Adaptive Methods

Beforehand, it is important to provide useful and important notations used throughout our paper.

**Notations:** We denote by $\theta$ the vector of parameters taking values in $\mathbb{R}^d$. For each layer $\ell \in [\![\mathsf{L}]\!]$, where $\mathsf{L}$ is the total number of layers of the neural networks, and each coordinate $j \in [\![p_\ell]\!]$ where $p_\ell$ is the dimension per layer $\ell$, we note $\theta^{\ell,j}$ its $j$th coordinate. The gradient of $f$ with respect to $\theta^\ell$ is denoted by $\nabla_\ell f(\theta)$. The index $i \in [\![n]\!]$ denotes the index of the worker $i$ in our federated framework. $r$ and $t$ are used as the round and local iteration numbers respectively. The smoothness per layer is denoted by $L_\ell$ for each layer $\ell \in [\![\mathsf{L}]\!]$. We note for each communication $r > 0$, the set of randomly drawn devices $D^r$ performing local updates.

### 2.1. AMSGrad, Local AMSGrad and Periodic Averaging

Under our Federated setting, we stress on the important of reducing the communication cost at each round between the central server, used mainly for aggregation purposes, and the many clients used for gradient computation and local updates. Using Periodic Averaging after few local epochs, updating local models on each device, as developed in (McMahan et al., 2017) is the gold standard for achieving such communication cost reduction. Intuitively, one rather shift the computation burden from the many clients to the central server as much as possible. This allows for fewer local epochs and a better global model, from a loss minimization (or model fitting) perspective.

The premises of that new paradigm are SGD updates performed locally on each device then averaged periodically, see (Konečnỳ et al., 2016; Zhou & Cong, 2017). The heuristic efficiency of local updates using SGD and periodic averaging has been studied in (Stich, 2018; Yu et al., 2019) and shown to reach a similar sublinear convergence rate as in the standard distributed optimization settings.

Then, with the growing need of training far more complex models, such as deep neural networks, several efficient methods, built upon adaptive gradient algorithms, such as Local AMSGrad in (Chen et al., 2020), extended both empirically and theoretically, the benefits of performing local updates coupled with periodic averaging.

### 2.2. Layerwise and Dimensionwise Learning with Periodic Averaging

Recall that our original problem is the following optimization task:

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

where $f_i(\theta)$ is the loss function associated to the client $i \in [\![n]\!]$ and is parameterized, in our paper, by a deep neural network. The multilayer and nonconvex nature of the loss function implies having recourse to particular optimization methods in order to efficiently train our model. Besides, the distributed and clients low bandwidth constraints are strong motivations for improving existing methods performing (1).

Based on the periodic averaging and local AMSGrad algorithms, presented prior, we propose a layerwise and dimensionwise local AMS algorithm described in the following:

**Algorithm 2** FED-LAMB for Federated Learning

```
1:  Input: parameter $\beta_1$, $\beta_2$, and learning rate $\alpha_t$.
2:  Init: $\theta_0 \in \Theta \subseteq \mathbb{R}^d$, as the global model and $\hat{v}_0 = v_0 = \epsilon 1 \in \mathbb{R}^d$ and $\bar{\theta}_0 = \frac{1}{n}\sum_{i=1}^{n} \theta_0$.
3:  for $r = 1$ to $R$ do
4:    Set $\theta^0_{r,i} = \bar{\theta}_{r-1}$
5:    for parallel for device $d \in D^r$ do
6:      Compute stochastic gradient $g_{r,i}$ at $\theta_r$.
7:      for $t = 1$ to $T$ do
8:        $m^t_{r,i} = \beta_1 m^{t-1}_{r-1,i} + (1-\beta_1) g_{r,i}$.
9:        $m^t_{r,i} = m^t_{r,i} / (1-\beta_1^r)$.
10:       $v^{t,i}_r = \beta_2 v^t_{r-1,i} + (1-\beta_2) g^2_{r,i}$.
11:       $v^t_{r,i} = v^t_{r,i} / (1-\beta_2^r)$.
12:       Compute ratio $p_{r,i} = \frac{m^t_{r,i}}{\sqrt{\hat{v}_r + \epsilon}}$.
13:       Update local model for each layer $\ell \in [\![\mathsf{L}]\!]$:
            $\theta^{\ell,t}_{r,i} = \theta^{\ell,t-1}_{r,i} - \alpha_r \phi(\|\theta^{\ell,t-1}_{r,i}\|) \frac{p^\ell_{r,i} + \lambda\theta^{\ell,t-1}_{r,i}}{\|p^\ell_{r,i} + \lambda\theta^{\ell,t-1}_{r,i}\|}$
14:     end for
15:     Devices send $\theta^T_{r,i} = [\theta^{\ell,T}_{r,i}]^{\mathsf{L}}_{\ell=1}$ and $v^T_{r,i}$ to server.
16:   end for
17:   Server computes the averages of the local models $\bar{\theta}^\ell_r = \frac{1}{n}\sum_{i=1}^{n} \theta^{\ell,T}_{r,i}$ and $\hat{v}_{r+1} = \max(\hat{v}_r, \frac{1}{n}\sum_{i=1}^{n} v^T_{r,i})$ and send them back to the devices.
18: end for
```

Algorithm 2 is a natural adaptation of the vanilla AMSGrad method, for *multilayer* neural networks under the *distributed* settings. In particular, while Line 2 and Line 2 corresponds to the standard approximation of the first and second moments, via the smooth updates allowed by the tuning parameters $\beta_1$ and $\beta_2$ respectively and that both Lines 2-2 are correct the biases of those estimates, the final local update in Line 2 is novel and corresponds to the specialization per layer of our federated method. Note that a scaling factor is applied to the learning rate $\alpha_r$ at each round $r > 0$ via the quantity $\phi(\|\theta_{r,i}^{\ell,t-1}\|)$ depending on the dimensionwise and layerwise quantity computed in Line 2. This function is user designed and can be set to the identity function.

The adaptivity of our federated method is thus manifold. There occurs a per dimension normalization with respect to the square root of the second moment used in adaptive gradient methods and a layerwise normalization obtained via the final local update (Line 2).

## 3. On The Convergence of FED-LAMB

We develop in this section, the theoretical analysis of Algorithm 2. Based on classical result for stochastic nonconvex optimization, we provide a collection of results that aims to providing a better understanding of the convergence behavior of our distributed optimization method under the federated learning framework. The main challenges we ought to overcome are manifold:

- the large amount of decentralized workers working solely on their own data stored locally.
- a periodic averaging occurs on the central server pushing each of those clients to send local models after some local iterations.
- each clients computes a backpropagation of the main model, *i.e.,* the deep neural network, and then update its local version of the model via an adaptive gradient method: the distinctiveness being that those updates are done *dimensionwise* and *layerwise*.

Our analysis encompasses the consideration of those challenges and leads to a informative convergence rates depending on the quantities of interest in our problem: the number of layers of the DNN, the number of communications rounds and the number of clients used under our federated settings.

### 3.1. Finite Time Analysis of FED-LAMB

In the context of nonconvex stochastic optimization for distributed devices, assume the following:

**H 1.** *For $i \in [\![n]\!]$ and $\ell \in [\![L]\!]$, $f_i$ is L-smooth:* $\|\nabla f_i(\theta) - \nabla f_i(\vartheta)\| \leq L_\ell \left\|\theta^\ell - \vartheta^\ell\right\|$.

We add some classical assumption in the unbiased stochastic optimization realm, on the gradient of the objective function:

**H2.** *The stochastic gradient is unbiased for any iteration $r > 0$: $\mathbb{E}[g_r] = \nabla f(\theta_r)$ and is bounded from above, i.e., $\|g_t\| \leq M$.*

**H3.** *The variance of the stochastic gradient is bounded for any iteration $r > 0$ and any dimension $j \in [\![d]\!]$: $\mathbb{E}[|g_r^j - \nabla f(\theta_r)^j|^2] < \sigma^2$.*

**H4.** *For any value $a \in \mathbb{R}_+^*$, there exists strictly positive constants such that $\phi_m \leq \phi(a) \leq \phi_M$.*

We now state our main result regarding the non asymptotic convergence analysis of our Algorithm 2:

**Theorem 1.** *Assume **H1-H4**. Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 2 with a decreasing learning rate $\alpha_r$. Then, if the number of local epochs is set to $T = 1$ and $\epsilon = \lambda = 0$, we have:*

$$\begin{aligned}
&\frac{1}{R}\sum_{r=1}^{R}\mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\|^2\right] \\
\leq&\sqrt{\frac{M^2 p}{n}}\frac{f(\bar{\vartheta}_1) - \mathbb{E}[f(\bar{\vartheta}_{R+1})]}{\mathsf{L}R} + \frac{\phi_M \sigma^2}{Rn}\sqrt{\frac{1-\beta_2}{M^2 p}} \\
&+ \alpha\phi_M\sigma\mathsf{L}p\sqrt{n} + \frac{\overline{L}\beta_1^2\mathsf{L}(1-\beta_2)M^2\phi_M^2 n}{2(1-\beta_1)^2 v_0} \\
&+ \frac{\alpha\beta_1}{1-\beta_1}\sqrt{(1-\beta_2)p}\frac{\mathsf{L}M^2}{\sqrt{v_0 <}} + \overline{L}\alpha^2 M^2\phi_M^2\frac{(1-\beta_2)p}{Rv_0}
\end{aligned} \tag{2}$$

We focus in the next subsection on two particular papers of utmost interest for our contribution.

### 3.2. Comparison with LAMB and Local-AMS

We dedicate the following paragraph to a discussion on the bound derived above in comparison with known results in the literature.

**LAMB bound in (You et al., 2019):** We first start our discussion with the comparison of convergence rate of FED-LAMBwith that of LAMB, Theorem 3 in (You et al., 2019). The convergence rates of FED-LAMBand LAMB differ in two ways: (i) the suboptimality, or convergence criterion, used here is $\frac{1}{R}\sum_{r=1}^{R}\mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\|^2\right]$ for a total number of rounds $R$ as opposed to $\mathbb{E}\left[\|\nabla f(\theta_{\mathcal{R}})\|^2\right]$ for some random termination round $\mathcal{R}$ uniformly drawn from $[\![R]\!]$. First, note that the characterization is given at the averaged parameters noted $\overline{\theta_{\mathcal{R}}}$ due to our distributed settings. It is thus natural to consider the evolution of our objective function, its gradient to be precise, evaluated at some global model values –as opposed to the outcome of a single step drift in the central server paradigm. Besides, for ease of interpretation, the LHS of (2) is summed over all rounds instead of a fictive random termination point. A simple calculation would lead to such characterization, found in several nonconvex stochastic optimization paper such as (Ghadimi & Lan, 2013). (ii) Assuming that the convergence criterion in both Theorems is of similar order (which happens for a large enough number of rounds), convergence rate of FED-LAMBdisplays a similar $\mathcal{O}(\frac{1}{R})$ behaviour for the initialization term, meaning that, despite the distributed (federated) settings, our dimensionwise and layerwise method benefits from the double adaptivity phenomenon explained above and exhibited in the LAMB method of (You et al., 2019), performed under a central server setting.

**Local-AMS bound in (Chen et al., 2020):** We now discuss the similarities and differences between the distributed adaptive method developed in (Chen et al., 2020) and named local-AMS, and our *deep federated* method, namely FED-LAMB. We first recall their main result:

**Theorem** (Theorem 5.1 in (Chen et al., 2020))**.** *Under some regularity conditions on the local losses and similar arguments as ours,* LOCAL-AMS *reaches a stationary point with the following rate:*

$$\frac{1}{R}\sum_{r=1}^{R}\mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\|^2\right] \leq 8\sqrt{\frac{p}{Rn}}\left[f(\bar{\vartheta}_1)-\mathbb{E}[f(\bar{\vartheta}_{R+1})]\right]+8L_s\sqrt{\frac{p}{Rn}}\frac{\sigma^2}{\epsilon}+cst. \tag{3}$$

*where $\epsilon$ corresponds to their initialization of the vector $\hat{v}_0$ and $L_s$ is the sum of the local smoothness constants.*

The first two terms of their results and ours, standard in convergence analysis, displays a strong dependence of the convergence rate on the initialization, which tends to be forgotten, and the bounded variance assumption of the stochastic gradient, see H3. The acceleration of our layerwise scheme is exhibited in the dependence on $\frac{1}{R}$ in those two terms, while results in (Chen et al., 2020) are of order $\frac{1}{\sqrt{R}}$. Note that the boundedness assumption is done on each dimension in H3 and leads to a manifestation of the term $\sqrt{p}$ in both rates. This can be handled for simplicity and clarity of the results when H3 is assumed globally. It is important to note that their rate include an analysis with respect to the number of local updates, noted $T$ in this paper. While our result is derived for a single local update $T=1$, we acknowledge that exhibiting a dependency on $T$ can be interesting, particularly in order to see the impact of several local updates which are, in practice, in different directions –as a byproduct of heterogeneous local training samples. We leave this investigation to future work.

In light of the prior remarks, we now give a simple variant of Theorem 1 in the following Corollary where we assume an upperbound on the norm of the second moment approximate $\sqrt{v_r^t}$:

**H5.** *For $t>0$ and $r>0$, there exists some constant such that $\|\sqrt{v_r^t}\| \leq V^2$.*

**Corollary 1.** *Assume **H1-H4**. Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 2 and set $\alpha_r = \mathcal{O}(\frac{1}{L\sqrt{R}})$. Then, if the number of local epochs is set to $T=1$, $\epsilon=\lambda=0$, and $R\geq$ we have, under **H5**:*

$$\frac{1}{R}\mathbb{E}\left[\left\|\nabla f(\overline{\theta_r})\right\|^2\right] \leq \mathcal{O}\left(\sqrt{\frac{p}{n}}\frac{1}{\mathsf{L}\sqrt{R}}\right) \tag{4}$$

We now discuss our bound regarding several aspects in order to gain understanding of the advantages of our method:

- Our (Karimireddy et al., 2019) (Reddi et al., 2020)
-
-

## 4. Numerical Study

In this section, we conduct numerical experiments on various datasets and network architectures to testify the effectiveness of our proposed method in practice.

**Settings.** In our experiment, we will evaluate three federated learning algorithms: 1) Fed-SGD, 2) Fed-AMS and 3) our proposed Fed-LAMB (Algorithm 2), where the first two serve as the baseline methods. For adaptive methods 2) and 3), we set $\beta_1=0.9$, $\beta_2=0.999$ as default and recommended (Reddi et al., 2018). Regarding federated learning environment, we use 50 local workers with 0.5 participation rate. That means, we randomly pick half of the workers to be active for training in each round. To best accord with real scenarios where the local training batch size is usually limited, we set a relatively small local update batch size as 32. In each round, the training samples are allocated to the active devices, and one local epoch is finished after all the local devices run one epoch over their received samples by batch training. We test different number of local epochs in our experiments. For each dataset and number of local epochs, we tune the constant learning rate $\alpha$ for each algorithm over a fine grid in logarithm scale. For Fed-LAMB, the parameter $\lambda$ in Algorithm 2 controlling the overall scale of the layerwise gradients is tuned from $\{0, 0.01, 0.1\}$. For each run, we take the model performance with the best $\alpha$ and $\lambda$. The reported results are averaged over three independent runs each with same initialization.

**Models.** We test the performance of different federated learning algorithms on MNIST and CIFAR10 image classification datasets. For MNIST, we apply 1) a simple multilayer perceptron (MLP), which has one hidden layer containing 200 cells with dropout; 2) Convolutional Neural Network (CNN), which has two max-pooled convolutional layers followed by a dropout layer and two fully-connected layers with 320 and 50 cells respectively. For CIFAR10, we implement: 1) a CNN with three convolutional layers followed by two fully-connected layers, and 2) a ResNet-9 model proposed by (He et al., 2016). All the networks use ReLU as the activation function.

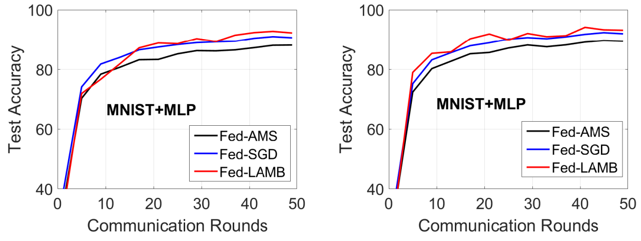

*Figure 1.* Test accuracy on MNIST+MLP, with non-iid data distribution. **Left panel:** 1 local epoch. **Right panel:** 5 local epochs.

### 4.1. MNIST with Multilayer Perceptron

In Figure 1, we start by presenting the test accuracy on MNIST dataset trained by MLP. We test heterogeneous (non-iid) data distribution, where each device only receives samples of one digit (out of ten). This is known to be the scenario where federated learning is harder to generalize well. Firstly, we observe that our proposed Fed-LAMB outperforms Fed-AMS and Fed-SGd with both 1 and 5 local epochs, illustrating its improvement over baseline federated method. In addition, though it is not the main comparison of this paper, Fed-SGD slightly generalizes better than Fed-AMS, which means that SGD might be sufficient for this rather simple model. Yet, Fed-LAMB beats both methods on this task.

### 4.2. MNIST with Convolutional Neural Network

We now evaluate the algorithms on larger models. Since Fed-LAMB is specifically designed for multi-layer deep learning networks, we expect it to show more substantial advantage on larger network architectures. In Figure 2, we present the results on MNIST with CNN, under non-iid data distribution. In general, we again see that Fed-LAMB outperforms Fed-AMS and Fed-SGD. The advantage is in particular significant with 1 local epoch, where Fed-SGD generalizes poorly. Importantly, we would like to address the acceleration effect of Fed-LAMB, in the early stage of training. We observe that Fed-LAMB converges faster than the vanilla Fed-AMS at first few communication rounds, in both cases. As a side note, the poor performance of Fed-SGD is, to some extent, consistent with prior literature and practical experience that adaptive gradient methods usually perform better than simple SGD in training large deep learning models [citation here].

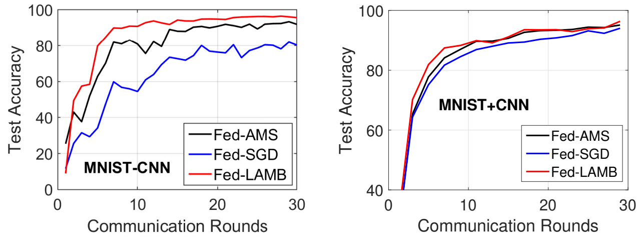

*Figure 2.* Test accuracy on MNIST+CNN, with non-iid data distribution. **Left panel:** 1 local epoch. **Right panel:** 5 local epochs.

### 4.3. CIFAR-10 with Convolutional Neural Network

In Figure 3, we report the test accuracy of CIFAR-10 dataset trained by CNN, where the data is iid allocated. When we run 1 local epoch, we observe clear advantage of Fed-LAMB over Fed-AMS on both test accuracy and convergence speed, and Fed-SGD again fails to achieve close performance. When local devices train 3 epochs per round, Fed-LAMB converges the fastest, with similar generalization error as Fed-AMS.

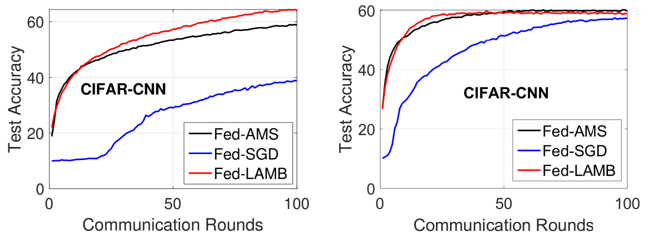

*Figure 3.* Test accuracy on CIFAR+CNN, with iid data distribution. **Left panel:** 1 local epoch. **Right panel:** 3 local epochs.

### 4.4. CIFAR-10 with Residual Neural Network

Lastly, we test the algorithms on CIFAR-10 and ResNet-9 model.

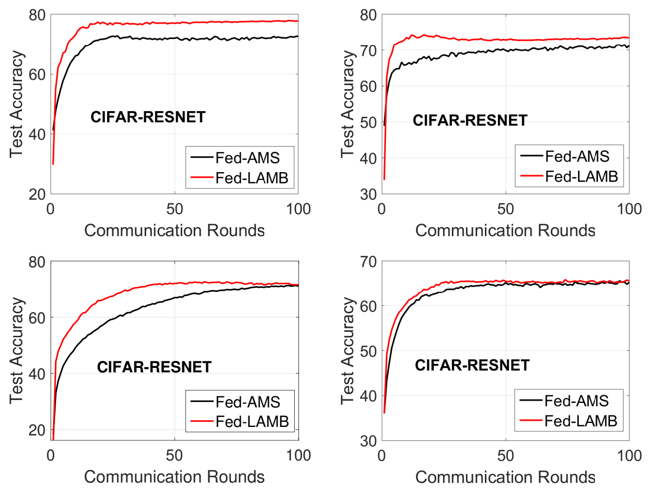

*Figure 4.* Test accuracy on CIFAR+CNN, with iid data distribution. **First row:** 10 clients. **Second row:** 50 clients. **Left panel:** 1 local epoch. **Right panel:** 3 local epochs.

## 5. Conclusion

We study in this contribution a doubly adaptive method in the particular framework of federated learning. Built upon the success of periodic averaging, and of state-of-the-arts adaptive gradient methods for single server non-convex stochastic optimization, we derive FED-LAMB, a distributed AMSGrad method that performs local updates on each worker and periodically averages local models. Besides, a core component of FED-LAMB, when the trained model is a deep neural network, is a layerwise update of each local model. Proved convergence guarantees of our scheme are provided in our contribution and exhibits a sub-linear dependence on the total number of communications rounds, the number of clients and the number of layers of the model. Multiple edges of periodic averaging, adaptive optimization methods and layerwise updates are displayed in our bounds. We empirically confirm the advantage of our algorithm over baselines methods on a panel of numerical experiments.

## References

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

Chen, X., Li, X., and Li, P. Toward communication efficient adaptive gradient method. In *ACM-IMS Foundations of Data Science Conference (FODS)*, Seattle, WA, 2020.

Dozat, T. Incorporating nesterov momentum into adam. *ICLR (Workshop Track)*, 2016.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 2011.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *ICLR*, 2015.

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

Li, M., Andersen, D. G., Park, J. W., Smola, A. J., Ahmed, A., Josifovski, V., Long, J., Shekita, E. J., and Su, B.-Y. Scaling distributed machine learning with the parameter server. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pp. 583–598, 2014.

Lin, Y., Han, S., Mao, H., Wang, Y., and Dally, W. J. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

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

McMahan, H. B. and Streeter, M. J. Adaptive bound optimization for online convex optimization. *COLT*, 2010.

Nesterov, Y. Introductory lectures on convex optimization: A basic course. *Springer*, 2004.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *Mathematics and Mathematical Physics*, 1964.

Recht, B., Re, C., Wright, S., and Niu, F. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *Advances in neural information processing systems*, 24: 693–701, 2011.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. *ICLR*, 2018.

Stich, S. U. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Tieleman, T. and Hinton, G. Rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 2012.

Wangni, J., Wang, J., Liu, J., and Zhang, T. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pp. 1299–1309, 2018.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

Yu, H., Jin, R., and Yang, S. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *arXiv preprint arXiv:1905.03817*, 2019.

Zeiler, M. D. Adadelta: An adaptive learning rate method. *arXiv:1212.5701*, 2012.

Zhao, W., Xie, D., Jia, R., Qian, Y., Ding, R., Sun, M., and Li, P. Distributed hierarchical gpu parameter server for massive scale deep learning ads systems. *arXiv preprint arXiv:2003.05622*, 2020.

Zhou, F. and Cong, G. On the convergence properties of a $k$-step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv preprint arXiv:1708.01012*, 2017.

## A. Theoretical Analysis

### A.1. Intermediary Lemmas

**Lemma 1.** *Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 2. Then for $i \in [\![n]\!]$:*

$$\|\overline{\theta_r} - \theta_{r,i}\| \leq \alpha^2 M^2 \phi_M^2 \frac{(1-\beta_2)p}{v_0} \tag{5}$$

*where $\phi_M$ is defined in H4 and $p$ is the total number of dimensions $p = \sum_{\ell=1}^{\mathsf{L}} p_\ell$.*

*Proof.* Assuming the simplest case when $T = 1$, i.e. one local iteration, then by construction of Algorithm 2, we have for all $\ell \in [\![\mathsf{L}]\!]$, $i \in [\![n]\!]$ and $r > 0$:

$$\theta_{r,i}^\ell = \overline{\theta_r}^\ell - \alpha\phi(\|\theta_{r,i}^{\ell,t-1}\|)p_{r,i}^j / \|p_{r,i}^\ell\| = \overline{\theta_r}^\ell - \alpha\phi(\|\theta_{r,i}^{\ell,t-1}\|)\frac{m_{r,i}^t}{\sqrt{v_r^t}}\frac{1}{\|p_{r,i}^\ell\|} \tag{6}$$

leading to

$$\begin{aligned}\|\overline{\theta_r} - \theta_{r,i}\|^2 &= \left\langle \overline{\theta_r}^\ell - \theta_{r,i}^\ell \,|\, \overline{\theta_r}^\ell - \theta_{r,i}^\ell \right\rangle \\ &\leq \alpha^2 M^2 \phi_M^2 \frac{(1-\beta_2)p}{v_0}\end{aligned} \tag{7}$$

which concludes the proof. □

### A.2. Proof of Theorem 1

**Theorem.** *Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 2. Then, if the number of local epochs is set to $T = 1$ and $\epsilon = \lambda = 0$, we have:*

$$\frac{1}{R}\sum_{r=1}^{R} \mathbb{E}[\|\nabla f(\overline{\theta_r})\|^2 \leq dd \tag{8}$$

**Case with $T = 1$, $\epsilon = 0$ and $\lambda = 0$:** Using H1, we have:

$$\begin{aligned}f(\bar{\vartheta}_{r+1}) &\leq f(\bar{\vartheta}_r) + \left\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \right\rangle + \sum_{\ell=1}^{L} \frac{L_\ell}{2}\|\bar{\vartheta}_{r+1}^\ell - \bar{\vartheta}_r^\ell\|^2 \\ &\leq f(\bar{\vartheta}_r) + \sum_{\ell=1}^{\mathsf{L}}\sum_{j=1}^{p_\ell} \nabla_\ell f(\bar{\vartheta}_r)^j (\bar{\vartheta}_{r+1}^{\ell,j} - \bar{\vartheta}_r^{\ell,j}) + \sum_{\ell=1}^{L} \frac{L_\ell}{2}\|\bar{\vartheta}_{r+1}^\ell - \bar{\vartheta}_r^\ell\|^2\end{aligned} \tag{9}$$

Taking expectations on both sides leads to:

$$-\mathbb{E}[\left\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \right\rangle] \leq \mathbb{E}[f(\bar{\vartheta}_r) - f(\bar{\vartheta}_{r+1})] + \sum_{\ell=1}^{L} \frac{L_\ell}{2}\mathbb{E}[\|\bar{\vartheta}_{r+1}^\ell - \bar{\vartheta}_r^\ell\|^2] \tag{10}$$

Yet, we observe that, using the classical intermediate quantity, used for proving convergence results of adaptive optimization methods, see for instance (Reddi et al., 2018), we have:

$$\bar{\vartheta}_r = \bar{\theta}_r + \frac{\beta_1}{1-\beta_1}(\bar{\theta}_r - \bar{\theta}_{r-1}) \tag{11}$$

where $\bar{\theta_r}$ denotes the average of the local models at round $r$. Then for each layer $\ell$,

$$\begin{aligned}
\bar{\vartheta}_{r+1}^{\ell} - \bar{\vartheta}_{r}^{\ell} &= \frac{1}{1-\beta_1}(\bar{\theta}_{r+1}^{\ell} - \bar{\theta}_{r}^{\ell}) - \frac{\beta_1}{1-\beta_1}(\bar{\theta}_{r}^{\ell} - \bar{\theta}_{r-1}^{\ell}) && (12)\\
&= \frac{\alpha_r}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\|p_{r,i}^{\ell}\|}p_{r,i}^{\ell} - \frac{\alpha_{r-1}}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta_{r-1,i}^{\ell}\|)}{\|p_{r-1,i}^{\ell}\|}p_{r-1,i}^{\ell} && (13)\\
&= \frac{\alpha\beta_1}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|} - \frac{\phi(\|\theta_{r-1,i}^{\ell}\|)}{\sqrt{v_{r-1}^t}\|p_{r-1,i}^{\ell}\|}\right)m_{r-1}^t + \frac{\alpha}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}g_{r,i} && (14)
\end{aligned}$$

where we have assumed a constant learning rate $\alpha$.

We note for all $\theta \in \Theta$, the majorant $G > 0$ such that $\phi(\|\theta\|) \leq G$. Then, following (10), we obtain:

$$-\mathbb{E}[\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \rangle] \leq \mathbb{E}[f(\bar{\vartheta}_r) - f(\bar{\vartheta}_{r+1})] + \sum_{\ell=1}^{L}\frac{L_\ell}{2}\mathbb{E}[\|\bar{\vartheta}_{r+1} - \bar{\vartheta}_r\|^2] \quad (15)$$

Developing the LHS of (15) using (12) leads to

$$\begin{aligned}
\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \rangle &= \sum_{\ell=1}^{\mathsf{L}}\sum_{j=1}^{p_\ell}\nabla_\ell f(\bar{\vartheta}_r)^j(\bar{\vartheta}_{r+1}^{\ell,j} - \bar{\vartheta}_{r}^{\ell,j}) && (16)\\
&= \frac{\alpha\beta_1}{1-\beta_1}\frac{1}{n}\sum_{\ell=1}^{\mathsf{L}}\sum_{j=1}^{p_\ell}\nabla_\ell f(\bar{\vartheta}_r)^j\left[\sum_{i=1}^{n}\left(\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|} - \frac{\phi(\|\theta_{r-1,i}^{\ell}\|)}{\sqrt{v_{r-1}^t}\|p_{r-1,i}^{\ell}\|}\right)m_{r-1}^t\right] && (17)\\
&\quad \underbrace{-\frac{\alpha}{n}\sum_{\ell=1}^{\mathsf{L}}\sum_{j=1}^{p_\ell}\nabla_\ell f(\bar{\vartheta}_r)^j\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}g_{r,i}}_{=A_1} && (18)
\end{aligned}$$

**Term $A_1$:** Since we have that $\|p_{r,i}^{\ell}\| \leq \sqrt{\frac{p_\ell}{1-\beta_2}}$ and $1/\sqrt{v_r^t} \leq 1/\sqrt{v_0}$, using H2, we develop the term $A_1$ as follows:

$$\begin{aligned}
A_1 &\leq -\frac{\alpha}{n}\sum_{\ell=1}^{\mathsf{L}}\sum_{j=1}^{p_\ell}\nabla_\ell f(\bar{\vartheta}_r)^j\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}g_{r,i} && (19)\\
&\leq -\frac{\alpha}{n}\sum_{\ell=1}^{\mathsf{L}}\sqrt{\frac{1-\beta_2}{M^2 p_\ell}}\sum_{i=1}^{n}\sum_{j=1}^{p_\ell}\phi(\|\theta_{r,i}^{\ell}\|)\nabla_\ell f(\bar{\vartheta}_r)^j g_{r,i}^{\ell,j} && (20)\\
&\quad -\frac{\alpha}{n}\sum_{\ell=1}^{\mathsf{L}}\sum_{i=1}^{n}\sum_{j=1}^{p_\ell}\left(\phi(\|\theta_{r,i}^{\ell}\|)\nabla_\ell f(\bar{\vartheta}_r)^j\frac{p_{r,i}^{\ell}}{\|p_{r,i}^{\ell}\|}\right)1\left(\text{sign}(\nabla_\ell f(\bar{\vartheta}_r)^j) \neq \text{sign}(p_{r,i}^{\ell})\right) && (21)
\end{aligned}$$

Taking the expectations on both sides yields and using H4:

$$\mathbb{E}[A_1] \leq -\alpha \sum_{\ell=1}^{\mathsf{L}} \sqrt{\frac{1-\beta_2}{M^2 p_\ell}} \sum_{i=1}^{n} \sum_{j=1}^{p_\ell} \mathbb{E}\left[\phi(\|\theta_{r,i}^{\ell}\|) \nabla_\ell f(\bar{\vartheta}_r)^j g_{r,i}^{\ell,j}\right] \tag{22}$$

$$- \frac{\alpha}{n} \sum_{\ell=1}^{\mathsf{L}} \sum_{i=1}^{n} \sum_{j=1}^{p_\ell} \mathbb{E}\left[\phi(\|\theta_{r,i}^{\ell}\|) \nabla_\ell f(\bar{\vartheta}_r)^j \frac{p_{r,i}^{\ell}}{\|p_{r,i}^{\ell}\|} \mathbb{1}\left(\text{sign}(\nabla_\ell f(\bar{\vartheta}_r)^j) \neq \text{sign}(p_{r,i}^{\ell})\right)\right] \tag{23}$$

$$\leq -\frac{\alpha}{n} \sum_{\ell=1}^{\mathsf{L}} \phi_m \sqrt{\frac{1-\beta_2}{M^2 p_\ell}} \sum_{i=1}^{n} \sum_{j=1}^{p_\ell} (\nabla_\ell f(\bar{\vartheta}_r)^j)^2 \tag{24}$$

$$- \frac{\alpha}{n} \sum_{\ell=1}^{\mathsf{L}} \sum_{i=1}^{n} \sum_{j=1}^{p_\ell} \phi_M \mathbb{E}\left[\left|\nabla_\ell f(\bar{\vartheta}_r)^j \frac{p_{r,i}^{\ell}}{\|p_{r,i}^{\ell}\|}\right| \mathbb{1}\left(\text{sign}(\nabla_\ell f(\bar{\vartheta}_r)^j) \neq \text{sign}(p_{r,i}^{\ell})\right)\right] \tag{25}$$

$$\tag{26}$$

where we have used assumption H4.

Since for any $\ell, i, j$, we have

$$\mathbb{E}\left[\left|\nabla_\ell f(\bar{\vartheta}_r)^j \frac{p_{r,i}^{\ell}}{\|p_{r,i}^{\ell}\|}\right| \mathbb{1}\left(\text{sign}(\nabla_\ell f(\bar{\vartheta}_r)^j) \neq \text{sign}(p_{r,i}^{\ell})\right)\right] \leq \left|\nabla_\ell f(\bar{\vartheta}_r)^j\right| \mathbb{P}\left(\text{sign}(\nabla_\ell f(\bar{\vartheta}_r)^j) \neq \text{sign}(p_{r,i}^{\ell})\right) \tag{27}$$

Then, we obtain

$$\mathbb{E}[A_1] \leq -\alpha\phi_m \sqrt{\frac{\mathsf{L}(1-\beta_2)}{M^2 p}} \mathbb{E}[\|\overline{\nabla f}(\bar{\vartheta}_r)\|^2] - \alpha\phi_M \sum_{\ell=1}^{\mathsf{L}} \sum_{i=1}^{n} \sum_{j=1}^{p_\ell} \frac{\sigma_i^{\ell,j}}{\sqrt{n}} \tag{28}$$

where for any $\theta \in \Theta$ we define $\overline{\nabla f}(\cdot) = \sum_{i=1}^{n} \nabla f_i(\cdot)$.

We now need to bound the following terms:

$$A_r^2 := \mathbb{E}[\|\bar{\vartheta}_{r+1} - \bar{\vartheta}_r\|^2] \tag{29}$$

$$A_r^3 := \frac{\alpha\beta_1}{1-\beta_1} \frac{1}{n} \sum_{\ell=1}^{\mathsf{L}} \sum_{j=1}^{p_\ell} \nabla_\ell f(\bar{\vartheta}_r)^j \left[\sum_{i=1}^{n} \left(\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|} - \frac{\phi(\|\theta_{r-1,i}^{\ell}\|)}{\sqrt{v_{r-1}^t}\|p_{r-1,i}^{\ell}\|}\right) m_{r-1}^t\right] \tag{30}$$

**Term $A_r^2$:** According to definition (11), for each layer $\ell \in [\![\mathsf{L}]\!]$, we have, using the Cauchy-Schwartz inequality, that:

$$\|\bar{\vartheta}_{r+1}^{\ell} - \bar{\vartheta}_r^{\ell}\|^2 = \left\|\frac{\alpha\beta_1}{1-\beta_1} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|} - \frac{\phi(\|\theta_{r-1,i}^{\ell}\|)}{\sqrt{v_{r-1}^t}\|p_{r-1,i}^{\ell}\|}\right) m_{r-1}^t + \frac{\alpha}{n} \sum_{i=1}^{n} \frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|} g_{r,i}\right\|^2 \tag{31}$$

$$\leq 2\frac{\alpha^2}{n^2} \left\|\frac{\beta_1}{1-\beta_1} \sum_{i=1}^{n} \left(\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|} - \frac{\phi(\|\theta_{r-1,i}^{\ell}\|)}{\sqrt{v_{r-1}^t}\|p_{r-1,i}^{\ell}\|}\right) m_{r-1}^t\right\|^2 + \frac{\alpha^2}{n^2} \left\|\sum_{i=1}^{n} \frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|} g_{r,i}\right\|^2 \tag{32}$$

Taking the expectation on both sides leads to:

$$
\begin{aligned}
\mathbb{E}[\|\bar{\vartheta}^\ell_{r+1} - \bar{\vartheta}^\ell_r\|^2] \leq 2\alpha^2 \mathbb{E}\left[\left\| \frac{\beta_1}{1-\beta_1} \sum_{i=1}^n \left( \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|} - \frac{\phi(\|\theta^\ell_{r-1,i}\|)}{\sqrt{v^t_{r-1}}\|p^\ell_{r-1,i}\|}\right) m^t_{r-1}\right\|^2\right] + \frac{\alpha^2}{n^2}\mathbb{E}\left[\left\|\sum_{i=1}^n \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|} g_{r,i}\right\|^2\right] \\
\leq 2\frac{\alpha^2}{n^2}\mathbb{E}\left[\left\| \frac{\beta_1}{1-\beta_1} \sum_{i=1}^n \left( \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|} - \frac{\phi(\|\theta^\ell_{r-1,i}\|)}{\sqrt{v^t_{r-1}}\|p^\ell_{r-1,i}\|}\right) m^t_{r-1}\right\|^2\right] \\
+ \frac{\alpha^2}{n^2}\mathbb{E}\left[\left|\sum_{i=1}^n\sum_{j=1}^p \left\langle \Gamma^j_{r,i}(\nabla f_i(\theta_r)^j + g^j_{r,i} - \nabla f_i(\theta_r)^j) \,|\, \Gamma^j_{r,i}(\nabla f_i(\theta_r)^j + g^j_{r,i} - \nabla f_i(\theta_r)^j)\right\rangle\right|\right] \\
\leq 2\frac{\alpha^2}{n^2}\mathbb{E}\left[\left\| \frac{\beta_1}{1-\beta_1} \sum_{i=1}^n \left( \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|} - \frac{\phi(\|\theta^\ell_{r-1,i}\|)}{\sqrt{v^t_{r-1}}\|p^\ell_{r-1,i}\|}\right) m^t_{r-1}\right\|^2\right] \\
+ \frac{\alpha^2}{n^2}\mathbb{E}\left[\left\|\sum_{i=1}^n \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|} \nabla f_i(\theta_r)\right\|^2\right] + \frac{\alpha^2}{n^2}\sum_{i=1}^n \sigma_i^2 \mathbb{E}\left[\left\| \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|}\right\|^2\right]
\end{aligned}
\tag{33}
$$

where the last line uses assumptions H2 and H3 (unbiased gradient and bounded variance of the stochastic gradient) and $\Gamma := \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|}$.

On the other hand, using the bound on the gradient H2,

$$
\begin{aligned}
&\sum_{r=1}^R \mathbb{E}\left[\left\| \frac{\beta_1}{1-\beta_1} \sum_{i=1}^n \left( \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|} - \frac{\phi(\|\theta^\ell_{r-1,i}\|)}{\sqrt{v^t_{r-1}}\|p^\ell_{r-1,i}\|}\right) m^t_{r-1}\right\|^2\right] \\
&\leq \frac{\beta_1^2}{(1-\beta_1)^2} M^2 \phi_M^2 \sum_{r=1}^R \mathbb{E}\left[\left\|\sum_{i=1}^n \left( \frac{1}{\sqrt{v^t_r}\|p^\ell_{r,i}\|} - \frac{1}{\sqrt{v^t_{r-1}}\|p^\ell_{r-1,i}\|}\right)\right\|^2\right] \\
&\leq \frac{\beta_1^2}{(1-\beta_1)^2} \frac{\mathsf{L}(1-\beta_2)}{p} M^2 \phi_M^2 \sum_{r=1}^R \mathbb{E}\left[\left\|\sum_{i=1}^n \left( \frac{1}{\sqrt{v^t_r}} - \frac{1}{\sqrt{v^t_{r-1}}}\right)\right\|^2\right] \\
&\leq \frac{\beta_1^2}{(1-\beta_1)^2} \frac{\mathsf{L}(1-\beta_2)}{p} M^2 \phi_M^2 \sum_{r=1}^R \mathbb{E}\left[\left|\sum_{i=1}^n \sum_{j=1}^p \left( \frac{1}{\sqrt{v^{t,j}_r}} - \frac{1}{\sqrt{v^{t,j}_{r-1}}}\right)\right|\right] \\
&\leq \frac{\beta_1^2}{(1-\beta_1)^2} \frac{\mathsf{L}(1-\beta_2)}{p} M^2 \phi_M^2 \frac{np}{v_0}
\end{aligned}
\tag{34}
$$

where, in the telescopic sum, we have used the initial value $v_0$ of the non decreasing sequence $\{v^t_r\}_{r>0}$ by construction (max operator).

Combining (34) into (33) and summing over the total number of rounds $R$ yields

$$
\begin{aligned}
\sum_{r=1}^R A_r^2 := \sum_{r=1}^R \mathbb{E}[\|\bar{\vartheta}^\ell_{r+1} - \bar{\vartheta}^\ell_r\|^2] \leq \frac{\beta_1^2}{(1-\beta_1)^2} \frac{\mathsf{L}(1-\beta_2)}{p} M^2 \phi_M^2 \frac{np}{v_0} \\
+ \sum_{r=1}^R \left[\frac{\alpha^2}{n^2}\mathbb{E}\left[\left\|\sum_{i=1}^n \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|} \nabla f_i(\theta_r)\right\|^2\right] + \frac{\alpha^2}{n^2}\sum_{i=1}^n \sigma_i^2 \mathbb{E}\left[\left\| \frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|p^\ell_{r,i}\|}\right\|^2\right]\right]
\end{aligned}
\tag{35}
$$

**Term $A_r^3$:** According to similar arguments on the non decreasing sequence involved in the algorithm as in the previous series of calculations, observe that

$$\sum_{r=1}^{R} A_r^3 \leq \frac{\alpha\beta_1}{1-\beta_1}\sqrt{(1-\beta_2)p}\frac{\mathsf{L}M^2}{\sqrt{v_0}} \tag{36}$$

Plugging (28) into (15) combined with (35) and (36) injected into the original smoothness definition (10) summed over the total number of rounds:

$$-\sum_{r=1}^{R}\mathbb{E}[\langle\nabla f(\bar{\vartheta}_r)\,|\,\bar{\vartheta}_{r+1}-\bar{\vartheta}_r\rangle] \leq \sum_{r=1}^{R}\mathbb{E}[f(\bar{\vartheta}_r)-f(\bar{\vartheta}_{r+1})] + \sum_{r=1}^{R}\sum_{\ell=1}^{L}\frac{L_\ell}{2}\mathbb{E}[\|\bar{\vartheta}_{r+1}^{\ell}-\bar{\vartheta}_r^{\ell}\|^2] \tag{37}$$

gives:

$$\begin{aligned}
&\sum_{r=1}^{R}\alpha\phi_m\sqrt{\frac{\mathsf{L}(1-\beta_2)}{M^2p}}\mathbb{E}[\|\overline{\nabla f}(\bar{\vartheta}_r)\|^2] - \alpha\phi_M\sum_{\ell=1}^{\mathsf{L}}\sum_{i=1}^{n}\sum_{j=1}^{p_\ell}\frac{\sigma_i^{\ell,j}}{\sqrt{n}} + \frac{\alpha\beta_1}{1-\beta_1}\sqrt{(1-\beta_2)p}\frac{\mathsf{L}M^2}{\sqrt{v_0}}\\
&\leq \sum_{r=1}^{R}\mathbb{E}[f(\bar{\vartheta}_r)-f(\bar{\vartheta}_{r+1})] + \sum_{\ell=1}^{L}\frac{L_\ell}{2}\frac{\beta_1^2}{(1-\beta_1)^2}\frac{\mathsf{L}(1-\beta_2)}{p}M^2\phi_M^2\frac{np}{v_0}\\
&\quad-\sum_{r=1}^{R}\left[\frac{\alpha^2}{n^2}\mathbb{E}\left[\left\|\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}\nabla f_i(\theta_r)\right\|^2\right] + \frac{\alpha^2}{n^2}\sum_{i=1}^{n}\sigma_i^2\mathbb{E}\left[\left\|\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}\right\|^2\right]\right]
\end{aligned} \tag{38}$$

Noting that $\sum_{r=1}^{R}\mathbb{E}[f(\bar{\vartheta}_r)-f(\bar{\vartheta}_{r+1})] = f(\bar{\vartheta}_1)-\mathbb{E}[f(\bar{\vartheta}_{R+1})]$, we obtain

$$\begin{aligned}
&\sum_{r=1}^{R}\alpha\phi_m\sqrt{\frac{\mathsf{L}(1-\beta_2)}{M^2p}}\mathbb{E}[\|\overline{\nabla f}(\bar{\vartheta}_r)\|^2] + \frac{1}{n^2}\sum_{r=1}^{R}\mathbb{E}\left[\left\|\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}\nabla f_i(\theta_r)\right\|^2\right]\\
&\leq f(\bar{\vartheta}_1)-\mathbb{E}[f(\bar{\vartheta}_{R+1})] + \frac{1}{n^2}\sum_{i=1}^{n}\sigma_i^2\mathbb{E}\left[\left\|\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}\right\|^2\right] + \alpha\phi_M\sum_{\ell=1}^{\mathsf{L}}\sum_{i=1}^{n}\sum_{j=1}^{p_\ell}\frac{\sigma_i^{\ell,j}}{\sqrt{n}} + \frac{\alpha\beta_1}{1-\beta_1}\sqrt{(1-\beta_2)p}\frac{\mathsf{L}M^2}{\sqrt{v_0}}\\
&+\sum_{\ell=1}^{L}\frac{L_\ell}{2}\frac{\beta_1^2}{(1-\beta_1)^2}\frac{\mathsf{L}(1-\beta_2)}{p}M^2\phi_M^2\frac{np}{v_0}
\end{aligned} \tag{39}$$

leading to

$$\begin{aligned}
\sum_{r=1}^{R}\mathbb{E}\left[\left\|\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}\nabla f_i(\theta_r)\right\|^2\right] &\leq \frac{1}{\alpha}\left[f(\bar{\vartheta}_1)-\mathbb{E}[f(\bar{\vartheta}_{R+1})]\right] + \sum_{r=1}^{R}\frac{\alpha}{n^2}\sum_{i=1}^{n}\sigma_i^2\mathbb{E}\left[\left\|\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}\right\|^2\right]\\
&+\alpha\phi_M\sigma\mathsf{L}p\sqrt{n} + \frac{\overline{L}\beta_1^2\mathsf{L}(1-\beta_2)M^2\phi_M^2 n}{2(1-\beta_1)^2v_0} + \frac{\alpha\beta_1}{1-\beta_1}\sqrt{(1-\beta_2)p}\frac{\mathsf{L}M^2}{\sqrt{v_0}}
\end{aligned} \tag{40}$$

where $\overline{L} = \sum_{\ell=1}^{\mathsf{L}} L_\ell$ is the sum of all smoothness constants.

Consider the following inequality:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^{\ell}\|)}{\sqrt{v_r^t}\|p_{r,i}^{\ell}\|}\nabla f_i(\theta_r) \leq \phi_M(1-\beta_2)\frac{\overline{\nabla}f(\theta_r)}{\sqrt{v_r^t}} \tag{41}$$

where $\overline{\nabla} f(\theta_r) := \frac{1}{n}\sum_{i=1}^n \nabla f_i(\theta_r)$. And using the Cauchy-Schwartz inequality we have

$$\left\|\frac{\overline{\nabla} f(\theta_r)}{\sqrt{v_r^t}}\right\| \geq \frac{1}{2}\left\|\frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\| - \left\|\frac{\overline{\nabla} f(\theta_r) - \nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\| \tag{42}$$

Using Lemma 1 and the smoothness assumption H1, we have

$$\begin{aligned}\left\|\frac{\overline{\nabla} f(\theta_r)}{\sqrt{v_r^t}}\right\| &\geq \frac{1}{2}\left\|\frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\| - \left\|\frac{\overline{\nabla} f(\theta_r) - \nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\| \\ &\geq \frac{1}{2}\left\|\frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\| - \overline{L}\alpha^2 M^2 \phi_M^2 \frac{(1-\beta_2)p}{v_0}\end{aligned} \tag{43}$$

Plugging the above inequality into (40) and dividing both sides by $R$ yields:

$$\begin{aligned}\frac{1}{R}\sum_{r=1}^R \mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r^t}}\right\|^2\right] \leq \sqrt{\frac{M^2 p}{n}}\frac{f(\bar{\vartheta}_1) - \mathbb{E}[f(\bar{\vartheta}_{R+1})]}{\mathsf{L}\alpha R} + \frac{\alpha}{n^2}\sum_{r=1}^R\sum_{i=1}^n \sigma_i^2 \mathbb{E}\left[\left\|\frac{\phi(\|\theta_{r,i}^\ell\|)}{\sqrt{v_r^t}\|p_{r,i}^\ell\|}\right\|^2\right] \\ + \alpha\phi_M\sigma\mathsf{L}p\sqrt{n} + \frac{\overline{L}\beta_1^2\mathsf{L}(1-\beta_2)M^2\phi_M^2 n}{2(1-\beta_1)^2 v_0} + \frac{\alpha\beta_1}{1-\beta_1}\sqrt{(1-\beta_2)p}\frac{\mathsf{L}M^2}{\sqrt{v_0}} + \overline{L}\alpha^2 M^2\phi_M^2\frac{(1-\beta_2)p}{Rv_0}\end{aligned} \tag{44}$$

concluding the proof of our main convergence result.

### A.3. Proof Corollary 1

**Corollary.** *Assume **H1-H4**. Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 2. Then, if the number of local epochs is set to $T = 1$, $\epsilon = \lambda = 0$, and $R \geq$ we have, under **H5**:*

$$\frac{1}{R}\mathbb{E}\left[\left\|\nabla f(\overline{\theta_r})\right\|^2\right] \leq \mathcal{O}\left(\sqrt{\frac{M^2 p}{n}}\frac{1}{\mathsf{L}R}\right) \tag{45}$$

*Proof.* From the bound in Theorem 1 and with assumption **H5** we obtain: □

## B. Additional Numerical Experiments