Fed-LAMB: Layer-wise and Dimension-wise Locally Adaptive Federated Learning

Belhal Karimi, Ping Li, Xiaoyun Li

Cognitive Computing Lab
Baidu Research
10900 NE 8th St, Bellevue, WA 98004, USA
{belhal.karimi, pingli98, lixiaoyun996}@gmail.com

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. Combining (*dimension-wise*) adaptive gradient methods (e.g., Adam, AMSGrad) with FL has been an active direction, which is shown to outperform traditional SGD based FL in many cases. In this paper, we focus on the problem of training federated deep neural networks, and propose a novel FL framework which further introduces *layer-wise* adaptivity to the local model updates to accelerate the convergence of adaptive FL methods. Our framework includes two variants based on two recent locally adaptive federated learning algorithms. Theoretically, we provide a convergence analysis of our layer-wise FL methods, coined Fed-LAMB and Mime-LAMB, which match the convergence rate of state-of-the-art results in adaptive FL and exhibits linear speedup in terms of the number of workers. Experimental results on various datasets and models, under both IID and non-IID local data settings, show that both Fed-LAMB and Mime-LAMB achieve faster convergence speed and better generalization performance, compared to various recent adaptive FL methods.

1 INTRODUCTION

A growing and important task while learning models on observed data, is the ability to train over a large number of clients which could either be personal devices or distinct entities. In the paradigm of Federated Learning (FL) [Konečnỳ et al., 2016, McMahan et al., 2017], a central server orchestrates the optimization over those clients under the constraint that the data can neither be gathered nor shared among the clients. This is computationally more efficient, since more distributed computing resources are used; also, this is a very practical scenario which allows individual data holders (e.g., mobile devices) to train a model jointly without leaking private data. In this paper, we consider the following optimization problem:

$$\min_{\theta} f(\theta) := \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\xi\sim\mathcal{X}_i}[F_i(\theta;\xi)], \quad (1)$$

where the nonconvex function (e.g., deep networks) f_i represents the average loss over the local data samples for worker $i \in [\![n]\!]$, and $\theta \in \mathbb{R}^d$ the global model parameter. $\mathcal{X}_i$ is the data distribution on each client i. There are two general scenarios of FL [Yang et al., 2019]: (i) *cross-silo* setting where n is small/moderate and the clients can be, e.g., different data servers; (ii) *cross-device* setting, where n can be large (e.g., millions) and the clients are mobile devices. While (1) reminds that of standard distributed optimization, the principle and setting of FL are different from the classical distributed paradigm. Two of the main differences are: (i) Local updates: FL allows clients to perform multiple updates on the local models before the global aggregation, which improves the computational resource efficiency and reduces the frequency of communication; (ii) Data heterogeneity: in FL, the local data distributions $\mathcal{X}_i$ are usually different across workers, hindering the convergence of the global model. Federated learning aims at finding a global solution of (1) in fewest number of communication rounds.

One of the standard and most popular frameworks for FL is called Fed-SGD [McMahan et al., 2017]: we adopt multiple local Stochastic Gradient Descent (SGD) steps in each device, send those local models to the server that computes the average over the received local model parameters, and broadcasts it back to the devices. Moreover, momentum can be added to local SGD training for faster convergence and better learning performance [Yu et al., 2019]. On the other hand, adaptive gradient methods (e.g., Adam [Kingma and Ba, 2015], AMSGrad [Reddi et al., 2018]) have shown great success in many deep learning tasks. For instance, the

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

update rule of Adam at step t reads as

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t}}, \quad m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \\ v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2, \tag{2}$$

where α is the learning rate and g_t is the gradient at time t. We note that the effective learning rate of Adam is $\alpha/\sqrt{v}$, which is different across dimensions, i.e., *dimension-wise* adaptive. Recently, we have seen growing research efforts in the design of FL frameworks that adopt adaptive gradient methods as the protocols for local model training instead of SGD. Examples include federated AMSGrad (Fed-AMS) [Chen et al., 2020] and Mime [Karimireddy et al., 2020] with Adam updates. Specifically, in both methods, in each round the global server not only aggregates the local models, but also broadcasts to the workers a "global" second moment estimation to reconcile the dimension-wise adaptive learning rates across the clients. Therefore, this step can be regarded as a natural mitigation to data heterogeneity, which is a common and important practical scenario that affects the performance of FL algorithms [Li et al., 2020a, Liang et al., 2019, Karimireddy et al., 2019]. Adaptive-optimizer based FL have been shown to outperform many SGD based FL methods on various tasks, making it a promising direction in FL system design.

In this work, we specifically focus on further improving the convergence speed and learning performance of locally adaptive FL algorithms. Our construction is based on introducing a special learning rate schedule into the local training of FL, which has not been proposed in the literature before. For (single-machine) training of deep neural networks using Adam, [You et al., 2020] proposed a *layer-wise* adjusted learning rate scheme called LAMB, where in each update, the effective update $m_t/\sqrt{v_t}$ is further normalized by the weight of each layer in the deep neural network, respectively. In [You et al., 2020], the authors proved that LAMB matches the convergence rate of Adam theoretically, and demonstrated the superior performance of LAMB empirically. With this weight-dependent adjusted learning rates, LAMB allows large-batch training which could in particular speed up training large datasets and models like ImageNet [Deng et al., 2009] and BERT [Devlin et al., 2019].

Contributions. Despite that layer-wise learning rate has been successfully implemented in (single-machine) model learning, one question that has not been explored is: can we also use methods like LAMB in the local training in federated learning? Is it able to also speedup the global model convergence? In this paper, we propose an improved framework for locally adaptive FL algorithms, integrating both *dimension-wise* and *layer-wise* adaptive learning rates in each device's local update. We provide theoretical and empirical justification on the efficacy of such layer-wise adaptivity in local federated training. More specifically, our contributions are summarized as follows:

- We develop Fed-LAMB and Mime-LAMB, two instances of our layer-wise adaptive optimization framework for FL, following a layer-wise adaptive strategy to accelerate the training of deep neural networks.
- We show that our algorithm converges at the rate of $\mathcal{O}\left(\frac{1}{\sqrt{n\mathsf{h}R}}\right)$ to a stationary point, where h is the number of layers of the network, n is the number of clients and R is the number of communication rounds. This matches the convergence rate of LAMB, AMSGrad, as well as the state-of-the-art results in federated learning. The theoretical communication efficiency matches that of Fed-AMS [Chen et al., 2020].
- We empirically compare several recent adaptive FL methods under both homogeneous and heterogeneous data setting on various benchmark datasets. Our results confirm the accelerated empirical convergence of Fed-LAMB and Mime-LAMB over the baseline methods, including Fed-AMS and Mime. In addition, Fed-LAMB and Mine-LAMB can also reach similar, or better, test accuracy than their corresponding baselines.

2 BACKGROUND

We summarize some relevant work on adaptive optimization, layer-wise adaptivity and federated learning.

Adaptive gradient methods. Adaptive methods have proven to be the spearhead for many nonconvex optimization tasks. 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. Common used examples include RMSprop [Tieleman and Hinton, 2012], Adadelta [Zeiler, 2012], Adam [Kingma and Ba, 2015], Nadam [Dozat, 2016] and AMSGrad [Reddi et al., 2018]. Their popularity owes to their great performance in training deep neural networks. They generally combine the idea of adaptivity from AdaGrad [Duchi et al., 2011, McMahan and Streeter, 2010], as explained above, and the idea of momentum from Nesterov's Method [Nesterov, 2003] or Heavy ball method [Polyak, 1964] using past gradients. AdaGrad displays superiority when the gradient is sparse compared to other classical methods [Duchi et al., 2011]. Yet, when applying AdaGrad to train deep neural networks, it is observed that the learning rate might decay too fast. Consequently, [Kingma and Ba, 2015] developed Adam whose updating rule is presented in (2). A variant, called AMSGrad [Reddi et al., 2018], forces v to be monotone to fix the convergence issue. [Loshchilov and Hutter, 2019] proposed AdamW that combines weight decay with Adam. The convergence and generalization of adaptive methods and their application in decentralized learning are studied in, e.g., [Zhou et al., 2018, Chen et al., 2019, Zhou et al., 2020, Wang et al., 2021, Chen et al., 2022], among others.

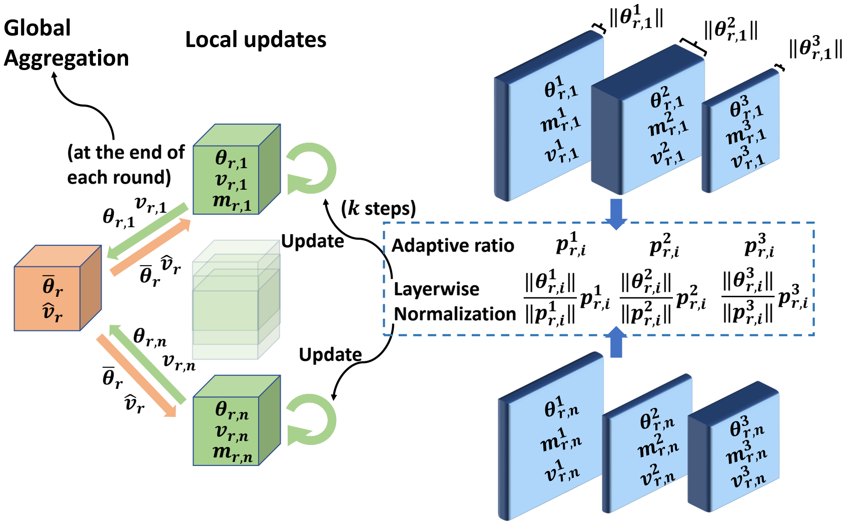
Global Aggregation
Local updates
(at the end of each round)
$\theta_{r,1}$ $v_{r,1}$ $m_{r,1}$
$\theta_{r,1}$ $v_{r,1}$
$\bar{\theta}_r$ $\hat{v}_r$
$\theta_{r,n}$ $v_{r,n}$
$\theta_{r,n}$ $v_{r,n}$ $m_{r,n}$
(k steps)
Update
Update
$\|\theta^1_{r,1}\|$
$\|\theta^2_{r,1}\|$
$\|\theta^3_{r,1}\|$
$\theta^1_{r,1}$ $m^1_{r,1}$ $v^1_{r,1}$
$\theta^2_{r,1}$ $m^2_{r,1}$ $v^2_{r,1}$
$\theta^3_{r,1}$ $m^3_{r,1}$ $v^3_{r,1}$
Adaptive ratio $p^1_{r,i}$ $p^2_{r,i}$ $p^3_{r,i}$
Layerwise Normalization $\frac{\|\theta^1_{r,i}\|}{\|p^1_{r,i}\|}p^1_{r,i}$ $\frac{\|\theta^2_{r,i}\|}{\|p^2_{r,i}\|}p^2_{r,i}$ $\frac{\|\theta^3_{r,i}\|}{\|p^3_{r,i}\|}p^3_{r,i}$
$\theta^1_{r,n}$ $m^1_{r,n}$ $v^1_{r,n}$
$\theta^2_{r,n}$ $m^2_{r,n}$ $v^2_{r,n}$
$\theta^3_{r,n}$ $m^3_{r,n}$ $v^3_{r,n}$

Figure 1: Illustration of Fed-LAMB framework (Algorithm 1), with a three-layer network and $\phi(x) = x$ as an example. For device i and each local iteration in round r, the adaptive ratio of j-th layer $\psi^j_{r,i}$ is normalized according to $\|\theta^j_{r,i}\|$, and then used for updating the local model. At the end of each round r, client i sends $\theta_{r,i} = [\theta^\ell_{r,i}]_{\ell=1}^{\mathrm{h}}$ and $v_{r,i}$ to the central server, which transmits back aggregated θ and $\hat{v}$ to devices to complete a round of training.

Layer-wise Adaptivity. When training deep networks, in many cases the scale of gradients differs a lot across the network layers. When we use the same learning rate for the whole network, the update might be too conservative for some specific layers (with large weights) which may slow down the convergence. Based on this observation, [You et al., 2018] proposed LARS, an extension of SGD with layer-wise adjusted scaling, whose performance, however, is not consistent accross tasks. Later, [You et al., 2020] proposed LAMB, an analogous layer-wise adaptive variant of Adam. The update rule of LAMB for the ℓ-th layer of the network can be expressed as

$$\theta^\ell_t = \theta^\ell_{t-1} - \frac{\alpha\|\theta^\ell_{t-1}\|}{\|\psi^\ell_t\|}\psi^\ell_t, \text{ with } \psi^\ell_t = m^\ell_t/\sqrt{v^\ell_t},$$

where m_t and v_t are defined in (2). Intuitively, for the ℓ-th layer, when the gradient magnitude is too small compared to the scale of the model parameter, we increase the effective learning rate to make the model move sufficiently far. Theoretically, [You et al., 2020] showed that LAMB achieves the same convergence rate as Adam; empirically, LAMB can significantly accelerate the convergence of Adam, allowing the use of large mini-batch size with fewer training iterations for large datasets.

Federated learning. An extension of the classic distributed training paradigm is called Federated Learning (FL) [Konečný et al., 2016, McMahan et al., 2017] which has seen many applications in various fields [Yang et al., 2019, Leroy et al., 2019, Bonawitz et al., 2019, Niknam et al., 2020, Xu et al., 2021]. For Fed-SGD (where clients perform SGD-based updates), recent variants and theoretical analysis on the convergence can be found in [Yu et al., 2019, Karimireddy et al., 2019, Khaled et al., 2020, Li et al., 2020c, Woodworth et al., 2020, Wang et al., 2020].

Many works have considered adaptive gradient methods in FL. [Reddi et al., 2021] proposed Adp-Fed where the central server applies Adam-type updates and the local clients perform SGD updates. [Li et al., 2022, Li and Li, 2023] studied distributed and federated adaptive method under communication compression. [Chen et al., 2020, Karimireddy et al., 2020] proposed Fed-AMS and Mime respectively, to adopt Adam/AMSGrad at the client level. Both works mitigate the influence of data heterogeneity by "sharing" the second moment v which controls the effective learning rates (more details will be provided later). Locally adaptive FL has also been applied to decentralized training [Zhao et al., 2022]. On many tasks, these methods outperform Fed-SGD and other popular methods like SCAFFOLD [Karimireddy et al., 2019] and FedProx [Li et al., 2020b, Yuan and Li, 2022]. Charles et al. [2021] empirically tested a FL method where LARS (i.e., layer-wise SGD) [You et al., 2018] is applied at the central server in local SGD, which is very different from our work in that the layer-wise adjustment happens locally with AMSGrad as the local optimizer. That is, our local models are trained with *dual* adaptivity.

3 LAYER-WISE LOCALLY ADAPTIVE FEDERATED LEARNING

In this section, we introduce our proposed FL framework, admitting both *dimension-wise* adaptivity (of adaptive learning rate) and *layer-wise* adaptivity (of layer-wise scaling).

Algorithm 1 Fed-LAMB and Mime-LAMB

```
1: Input: $0 < \beta_1, \beta_2 < 1$; learning rate $\alpha$; weight decaying rate $\lambda \in [0,1]$; frequency parameter $Z$.
2: Initialize: $\theta_{0,i} \in \Theta \subseteq \mathbb{R}^d$; $m^0_{0,i} = \hat{v}^0_{0,i} = v^0_{0,i} = 0$, $\forall i \in [\![n]\!]$; $\bar{\theta}_0 = \frac{1}{n}\sum_{i=1}^n \theta_{0,i}$; $\hat{v}_0 = \epsilon$
3: for $r = 1$ to $R$ do
4:     Sample a set of clients $D^r$
5:     for parallel for device $i \in D^r$ do
6:         Set $\theta^0_{r,i} = \bar{\theta}_{r-1}$, $m^0_{r,i} = m^T_{r-1,i}$ , $v^0_{r,i} = \hat{v}_{r-1}$
7:         for $t = 1$ to $T$ do
8:             Sample a mini-batch from the local data
9:             Compute stochastic gradient $g^t_{r,i}$ at $\theta^{t-1}_{r,i}$
10:            $m^t_{r,i} = \beta_1 m^{t-1}_{r,i} + (1-\beta_1) g^t_{r,i}$
11:            $v^t_{r,i} = \beta_2 v^t_{r-1,i} + (1-\beta_2)(g^t_{r,i})^2$
12:            Compute the ratio $\psi^t_{r,i} = m^t_{r,i}/\sqrt{\hat{v}_{r-1}}$.
13:            Update local model for each layer $\ell \in [\![\mathsf{h}]\!]$:
```

$$\theta^{\ell,t}_{r,i} = \theta^{\ell,t-1}_{r,i} - \frac{\alpha_r \phi(\|\theta^{\ell,t-1}_{r,i}\|)(\psi^{\ell,t}_{r,i} + \lambda\theta^{\ell,t-1}_{r,i})}{\|\psi^{\ell,t}_{r,i} + \lambda\theta^{\ell,t-1}_{r,i}\|} \quad (3)$$

```
14:        end for
15:        Communicate $\theta^T_{r,i} = [\theta^{\ell,T}_{r,i}]^{\mathsf{h}}_{\ell=1}$ to server
16:        Communicate $v^T_{r,i}$ to server
17:        Communicate $\nabla f_i(\bar{\theta}_{r-1})$ using full local data
18:    end for
19:    Server compute $\bar{\theta}_r = \frac{1}{|D^r|}\sum_{i\in D^r}\theta^T_{r,i}$
20:    Server compute $\hat{v}_r = \max(\hat{v}_{r-1}, \frac{1}{|D^r|}\sum_{i\in D^r} v^T_{r,i})$
21:    Compute $\nabla f(\bar{\theta}_{r-1}) = \frac{1}{|D^r|}\sum_{i\in D_r}\nabla f_i(\bar{\theta}_{r-1})$
22:    Compute $v_r = \beta_2 v_{r-1} + (1-\beta_2)\nabla f(\bar{\theta}_{r-1})^2)$
23:    Update $\hat{v}_r = \max(\hat{v}_{r-1}, v_r)$
24: end for
```

We mainly consider AMSGrad [Reddi et al., 2018] as the prototype adaptive gradient method. We assume the loss function $f(\cdot)$ is induced by a multi-layer neural network, which includes a broad class of network architectures like MLP, CNN, ResNet and Transformers.

Notations. We denote by θ the vector of parameters taking values in $\mathbb{R}^p$. Suppose the neural network has h layers, each with size p_ℓ (thus, $p = \sum_{\ell=1}^{\mathsf{h}} p_\ell$). For each layer $\ell \in [\![\mathsf{h}]\!]$, denote θ^ℓ as the sub-vector corresponding to the ℓ-th layer. Let R be the number of communication rounds and T be the number of local iterations per round. Moreover, $\theta^{\ell,t}_{r,i}$ is the model parameter of layer ℓ at round r, local iteration t and for worker i.

Algorithm. In general, our proposed algorithm can be viewed as a novel extension of LAMB to the more complicated federated learning setting. Based on the two recent works regarding locally adaptive FL mentioned above, we present the framework with two instances, Fed-LAMB and Mime-LAMB, as summarized in Algorithm 1 and depicted in Figure 1. We differentiate the steps of these two methods by blue (Fed-LAMB) and red (Mime-LAMB) boxes surrounding the text. Both methods use layer-wise adaptive LAMB for local updates (Line 13). The update in (3) on local workers can be expressed as

$$\theta \leftarrow \theta - \alpha\frac{\phi(\|\theta\|)}{\|\psi + \lambda\theta\|}(\psi + \lambda\theta),$$

where $\phi(\cdot) : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is a scaling function (usually chosen to be the identity function in practice) and λ is the weight decay rate. This way, the gradients are effectively normalized by the magnitude of layer weights, forcing the model move sufficiently far at every layer. Such normalization effect may accelerate the convergence of the model.

The main difference between the two variants, Fed-LAMB and Mime-LAMB, is the way the second moment $\hat{v}$ is synchronized, i.e., the dimension-wise adaptive learning rate. Both methods maintain a global $\hat{v}$ at the central server:

- Fed-LAMB (Line 20) : at the end of each round, the i-th client communicates the local v_i; the server updates the global $\hat{v}$ by the max operation with the averaged v among all clients, and sends back the global $\hat{v}$.
- Mime-LAMB (Line 21-23) : in each round r, the client computes and transmits the gradient at the global model $\bar{\theta}_r$ using full local data; the server updates the global v and $\hat{v}$ in the same manner as AMSGrad.

When implementing the algorithms, note that in Mime-LAMB, the global v is directly calculated using full-batches (averaged over all clients). As a result, Mime-LAMB needs to calculate the gradients twice, leading to twice the computational cost as Fed-LAMB. We also note that, the local update of Fed-LAMB (Line 13 of Algorithm 1) also incorporates the “decoupled ”weight decay, which is same as the weight decay mechanism used in the AdamW algorithm [Loshchilov and Hutter, 2019].

Data Heterogeneity: Conceptually, both the two approaches aim at alleviating the impact of data heterogeneity by globally reconciling the adaptive learning rates. We call this “moment sharing”. Therefore, in some sense, Algorithm 1 is naturally capable of balancing the heterogeneity in different local data distributions. Indeed, in [Chen et al., 2020] and [Karimireddy et al., 2020], the authors have shown that Fed-AMS and Mime would perform much worse, or even diverge, without aggregating and sharing the second moment $\hat{v}$ (please refer to the papers for details). Intuitively, synchronizing $\hat{v}$ makes all the clients “on the same pace” which is crucial for the convergence of locally adaptive FL methods.

Extension: skip synchronization of $\hat{v}_t$. In practice, when trained with the same number of rounds R and local iterations T, Mime, Fed-LAMB and Fed-Mime all require communicating two tensors (the local model update, and second moment v), while Fed-SGD [McMahan et al., 2017] and Adp-Fed [Reddi et al., 2021] only communicate one local update tensor. Hence, locally adaptive methods in general tend to require more communication. We now discuss a simple implementation trick of our algorithm that reduces this extra cost. Note that, as long as $\hat{v}_t$ is consistent across clients, we may not need to update and broadcast it in every round. To reduce the extra communication overhead of transmitting $\hat{v}$, one trick is to reduce the aggregation frequency of $\hat{v}$ in Algorithm 1 (e.g., we synchronize $\hat{v}$ every Z rounds). It can be shown that this "skip" aggregation of the second moment does not affect the convergence rate of our Fed-LAMB (see Theorem 5). Yet it can effectively reduce the communication of $\hat{v}$ by a factor of Z, which to a great extent alleviates the extra communication cost of locally adaptive methods. We will also show empirical evidence of this strategy in our experiments.

4 THEORETICAL ANALYSIS

In the context of nonconvex stochastic optimization for federated learning, we will make the following standard analytical assumptions.

Assumption 1 (Smoothness)**.** For all $i \in [\![n]\!]$ and $\ell \in [\![\mathsf{h}]\!]$, the local loss function is L_ℓ-smooth: $\left\|\nabla f_i(\theta^\ell) - \nabla f_i(\vartheta^\ell)\right\| \leq L_\ell \left\|\theta^\ell - \vartheta^\ell\right\|$.

Assumption 2 (Unbiased and bounded gradient)**.** The stochastic gradient is unbiased for $\forall r, t, i$: $\mathbb{E}[g_{r,i}^t] = \nabla f_i(\theta_r^t)$ and bounded by $\left\|g_{r,i}^t\right\| \leq M$.

Assumption 3 (Bounded variance)**.** The stochastic gradient admits (*locally*) $\mathbb{E}[|g_{r,i}^j - \nabla f_i(\theta_r)^j|^2] \leq \sigma^2$, and (*globally*) $\frac{1}{n}\sum_{i=1}^n ||\nabla f_i(\theta_r) - \nabla f(\theta_r)||^2] \leq G^2$.

Assumption 1 and Assumption 2 are commonly used in the analysis of adaptive gradients methods [Reddi et al., 2018, Chen et al., 2019, Reddi et al., 2021, Karimireddy et al., 2020]. Assumption 3 characterizes the data heterogeneity among local devices, and $G = 0$ when local data are IID.

Same as in [You et al., 2020], we further make the following assumption on the scaling function ϕ.

Assumption 4 (Bounded scaling function)**.** For any $a > 0$, there exist $\phi_m > 0, \phi_M > 0$ such that $\phi_m \leq \phi(a) \leq \phi_M$.

Assumption 4 can be satisfied when, for example, we let $\phi(a) = \min\{a + \zeta, \phi_M\}$ be the identity map plus a small constant ζ with an upper clipping threshold at some ϕ_M.

We now state our main result regarding the convergence rate of the proposed Algorithm 1.

Theorem 5. *Under Assumption 1-Assumption 4, consider $\{\overline{\theta_r}\}_{r>0}$ obtained from Algorithm 1 with a constant learning rate α. Suppose $\lambda = 0$. Then the squared gradient of the global model uniformly chosen from round $1, ..., R$ is bounded by*

$$
\begin{aligned}
&\frac{1}{R}\sum_{r=1}^{R} \mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2\right] \\
&\leq \sqrt{\frac{M^2 p}{n}}\frac{\triangle}{\mathsf{h}\alpha R} + \frac{4\alpha^2 \overline{L} M^2 T^2 \phi_M^2 (1-\beta_2)p}{\sqrt{\epsilon}} \\
&+ 4\alpha^2 \frac{M^2}{\sqrt{\epsilon}} + \frac{\phi_M \sigma^2}{Rn}\sqrt{\frac{1-\beta_2}{M^2 p}} + 4\alpha^2\left[\phi_M^2\sqrt{M^2 + p\sigma^2}\right] \\
&+ 4\frac{\alpha^2 \overline{L}}{\sqrt{\epsilon}} M^2 T^2 G^2 (1-\beta_2)p + 4\alpha\left[\phi_M \frac{\mathsf{h}\sigma^2}{\sqrt{n}}\right],
\end{aligned}
\tag{4}
$$

where $\triangle = \mathbb{E}[f(\bar{\theta}_1)] - \min_{\theta \in \Theta} f(\theta)$ and $\overline{L} = \sum_{\ell=1}^{\mathsf{h}} L_\ell$.

Remark 6. *Theorem 5 applies to both Fed-LAMB and Mime-LAMB variants. Also, the manifestation of p in the rate is because the variance bound is assumed on each dimension in Assumption 3. This dependency on p can be removed when Assumption 3 is assumed globally, which is also common in optimization literature. Moreover, this result also holds for Algorithm 1 with skip synchronization of $\hat{v}_t$ as discussed earlier.*

Using the uniform boundedness of the second moment accumulator $\|\hat{v}_r\|$ (which can be shown by Assumption 2) and by choosing a suitable decreasing learning rate, we have the following simplified statement.

Corollary 7. *Under the same setting as Theorem 5, with $\alpha = \mathcal{O}(\frac{1}{\sqrt{\mathsf{h}R}})$, it holds that*

$$
\begin{aligned}
\frac{1}{R}\sum_{r=1}^{R} \mathbb{E}\left[\left\|\nabla f(\overline{\theta_r})\right\|^2\right] \\
\leq \mathcal{O}\left(\frac{\sqrt{p}}{\sqrt{n\mathsf{h}R}} + \frac{\sqrt{\mathsf{h}}\sigma^2}{\sqrt{nR}} + \frac{G^2 T^2 p}{R\mathsf{h}}\right).
\end{aligned}
\tag{5}
$$

The leading two terms display a dependence of the convergence rate of Fed-LAMB on the initialization and the local variance of the stochastic gradients (Assumption 3). The last term involves the number of local updates T, and the global variance G^2 characterizing the data heterogeneity. Next, we provide detailed discussion and comparison of our result to related prior works.

LAMB bound in [You et al., 2020]: We start our discussion with the comparison of our convergence rate with that of LAMB, Theorem 3 in [You et al., 2020]. In the single-machine setting, the convergence rate of LAMB is $\mathcal{O}(\sqrt{p}\sqrt{\mathsf{h}T})$ where T is the number of training iterations. Note the convergence rate of Fed-LAMB is different from that of LAMB in the sense that, the convergence criterion is given at the averaged parameters (global model) at the end of each round. In Corollary 7, our rate would match LAMB if we take number of local step $T = 1$. This is also true for any fixed T and R sufficiently large. In addition, the $\mathcal{O}(\frac{1}{\sqrt{nR}})$ rate of Fed-LAMB implies an important *linear speedup* effect: the number of iterations to reach a δ-stationary point of Fed-LAMB decreases linearly in n, which displays the merit of distributed (federated) learning.

Fed-AMS bound in [Chen et al., 2020]: We now compare our method theoretically with Fed-AMS, the baseline distributed adaptive method developed in [Chen et al., 2020]. Their results state that when $T \leq \mathcal{O}(R^{1/3})$, the convergence rate of Fed-AMS is $\mathcal{O}(\frac{1}{\sqrt{nR}})$. Firstly, when the number of rounds R is sufficiently large, both our rate (5) and the rate of Fed-AMS are dominated by $\mathcal{O}(\frac{1}{\sqrt{nR}})$, improving the convergence rate of the standard AMSGrad, e.g. [Zhou et al., 2018] by $\mathcal{O}(1/\sqrt{n})$ (i.e., linear speedup). Secondly, in (5), the last term containing the number of local updates T is small as long as $T^4 \leq \mathcal{O}(\frac{Rh}{G^2})$. If we further assume $h \simeq T$, then we get the same rate of convergence as Fed-AMS with $T \leq \mathcal{O}(R^{1/3})$ local iterations, identical to the condition of Fed-AMS. Under these analytic settings and conditions , the convergence rate of Fed-LAMB also matches many popular federated learning methods in nonconvex optimization, e.g., Fed-SGD [McMahan et al., 2017], Mime [Karimireddy et al., 2020] and Adp-Fed [Reddi et al., 2021]. Moreover, when G is small (less data heterogeneity), the bound on T would increase, i.e., we can conduct more local updates. This is intuitive, for example, when $G = 0$ in the IID data setting, T can be very large.

As a brief summary, Fed-LAMB achieves the same asymptotic convergence rate as Fed-AMS in the federated (distributed) learning setting. Our method also exhibits the favorable linear speedup property regarding the number of clients in the system. Next, we will show that Fed-LAMB and its variants provide impressive acceleration empirically in our experimental study presented next.

5 EXPERIMENTS

In this section, we conduct experiments on benchmark datasets with various network architectures to justify the effectiveness of our proposed method in practice. Our method empirically confirms its merit in terms of convergence speed. Basically, Fed-LAMB and Mime-LAMB reduce the number of rounds and thus the communication cost required to achieve a similar stationary point (or test accuracy) than the baseline methods. In many cases, Fed-LAMB also brings notable improvement in generalization over baselines.

Methods. We evaluate the following five FL algorithms, mainly focusing on recent federated optimization approaches based on adaptive gradient methods:

1. Fed-SGD [McMahan et al., 2017], standard federated averaging with local SGD updates.
2. Adp-Fed (*Adaptive Federated Optimization*, see Appendix for more details), the federated adaptive algorithm proposed by [Reddi et al., 2021]. Adp-Fed performs local SGD updates. In each round r, the changes in local models, $\triangle_i = w_{r,i}^T - w_{r,i}^0, i = 1, ..., n$, are sent to the central server for an aggregated Adam update.
3. Fed-AMS [Chen et al., 2020], locally adaptive AMSGrad algorithm.
4. Mime [Karimireddy et al., 2020] with AMSGrad, which performs adaptive local updates with central-server-guided global adaptive learning rate.
5. Our proposed Fed-LAMB and Mime-LAMB methods (Algorithm 1).

For all the adaptive gradient methods, we set $\beta_1 = 0.9$, $\beta_2 = 0.999$ by default [Reddi et al., 2018]. We present the results of $n = 50$ clients with 0.5 participation rate, i.e., we randomly pick half of the clients to be active for training in each round, and the local mini-batch size is set as 128. In each round, the training samples are allocated to the active devices, and one local epoch is completed after all the local devices run one pass over their received samples via mini-batch training. Results with more clients can be found in the Appendix, which give the same conclusions as what we will present below.

We tune the learning rate α for each algorithm over a fine grid. For Adp-Fed, there are two learning rates involved (global and local), both of which are tuned. More tuning details can be found in the Appendix. For Fed-LAMB and Mime-LAMB, the weight decay rate λ is tuned from $\{0, 0.01, 0.1\}$, and $\phi(x) = x$ is the identity mapping. For each run, we report the best test accuracy. The results are averaged over 3 runs each from a same initialization point.

Datasets and models. We experiment with four popular benchmark image classification datasets: MNIST [LeCun, 1998], Fashion MNIST (FMNIST) [Xiao et al., 2017], CIFAR-10 [Krizhevsky and Hinton, 2009] and TinyImageNet [Deng et al., 2009]. For MNIST, we apply 1) a simple multi-layer perceptron (MLP), which has one hidden layer containg 200 cells; 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. This CNN is also implemented for FMNIST. For CIFAR-10 and TinyImageNet, we use ResNet-18 [He et al., 2016].

5.1 COMPARISON UNDER IID SETTINGS

In Figure 2, we report the test accuracy of MLP trained on MNIST, and of CNN trained on MNIST and FMNIST, where the data are IID allocated among the clients. We test

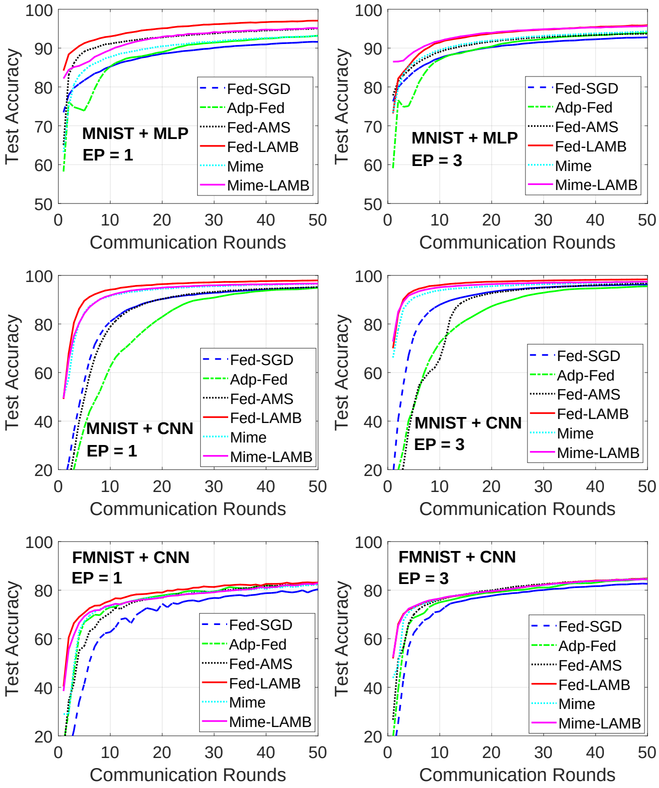

Figure 2: **IID data setting**. Test accuracy against the number of communication rounds.

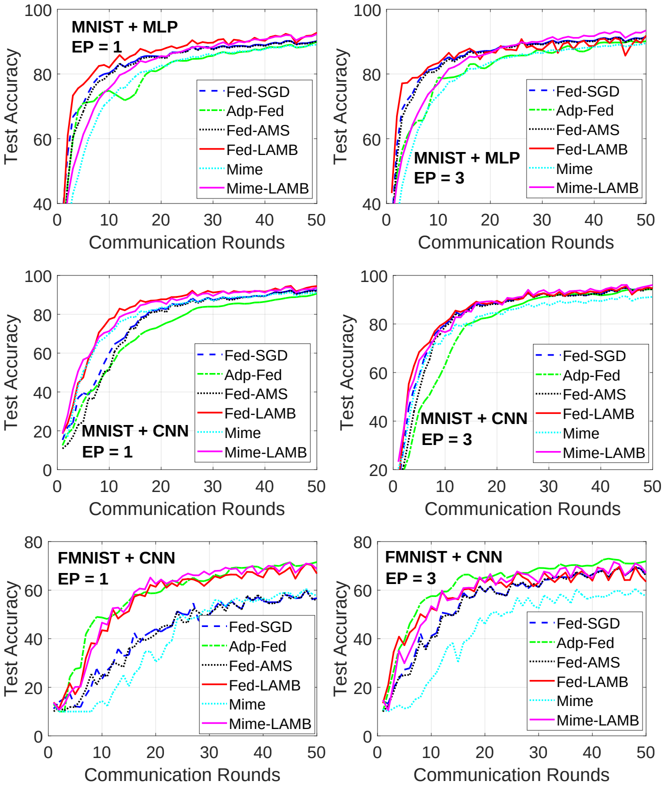

Figure 3: **non-IID data setting.** Test accuracy against the number of communication rounds.

1 local epoch and 3 local epochs. In all the figures, we observe a clear advantage of Fed-LAMB over the competing methods in terms of the convergence speed. In particular, we can see that Fed-LAMB is able to achieve the same accuracy with fewest number of communication rounds, thus improving the model training efficiency. For instance, this can be observed as follows: on MNIST + CNN (1 local epoch), Fed-AMS requires 20 rounds to achieve 90% accuracy, while Fed-LAMB only takes 5 rounds. This implies a 75% reduction in the communication cost and training time. Moreover, on MNIST, Fed-LAMB also leads to improved generalisation performance, i.e., test accuracy. We can draw same conclusions with 3 local epochs. Also, similar comparison holds for Mime-LAMB vs. Mime. In general, the Mime-LAMB and Fed-LAMB perform similarly.

5.2 COMPARISON UNDER NON-IID SETTINGS

In Figure 3, we provide the results on MNIST and FMNIST with non-IID local data distribution. In particular, in each round of federated training, every local device only receives samples from one or two classes (out of ten). We see that for experiments with 1 local epoch, in all cases our proposed Fed-LAMB outperforms all the baseline methods. Similar to the IID data setting, Fed-LAMB provides faster convergence speed and achieves higher test accuracy than Fed-SGD and Fed-AMS. The advantage is especially significant for the CNN model, e.g., it improves the accuracy of Fed-SGD and Fed-AMS by more than 10% on FMNIST at the 50-th round. The other baseline method, Adp-Fed, performs as good as our Fed-LAMB on FMNIST, but worse than other methods on MNIST. Mime-LAMB also considerably improves Mime, in all the runs, see Figure 3.

The relative comparison is basically the same for 3 local epochs, but the advantage of Fed-LAMB becomes less significant than what we observed in Figure 2 with IID data. One plausible reason is that when the local data is highly non-IID. Intuitively, with more local steps, learning the local models fast might not always do good to the global model, as local models target at different loss functions.

In Figure 4, we present the results on CIFAR-10 and TinyImageNet datasets trained by ResNet-18. When training these two models, we decrease the learning rate to $1/10$ at the 30-th and 70-th communication round. From Figure 4, we can draw similar conclusion as before: the proposed Fed-LAMB is the best method in terms of both convergence speed and generalization accuracy. In particular, on TinyImageNet, we see that Fed-LAMB has a significant advantage over all the four baselines without layer-wise acceleration. Although Adp-Fed performs better than Fed-SGD and Fed-AMS, it is

	Fed-SGD	Adp-Fed	Fed-AMS	**Fed-LAMB**	Mime	**Mime-LAMB**
CIFAR-10	90.75 ± 0.48	91.57 ± 0.38	90.93 ± 0.22	**92.44 ± 0.53**	90.94 ± 0.13	**92.00 ± 0.21**
TinyImageNet	67.58 ± 0.21	74.17 ± 0.43	64.86 ± 0.83	**76.00 ± 0.26**	67.82 ± 0.24	**73.46 ± 0.25**

Table 1: Test accuracy with ResNet-18 network after 100 communication rounds.

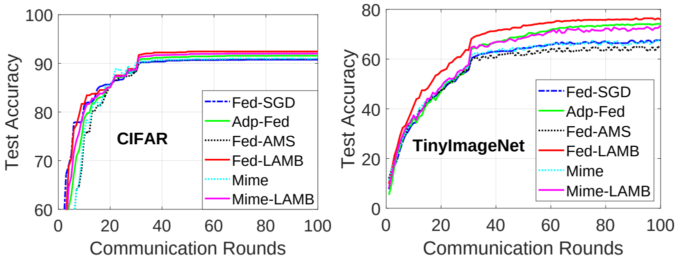

Figure 4: **non-IID data.** Test accuracy of CIFAR-10 and TinyImagenet on ResNet-18.

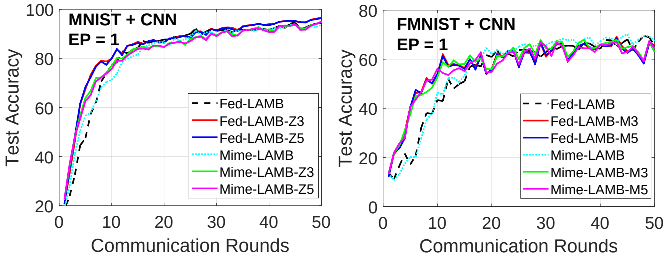

Figure 5: **non-IID data.** Fed-LAMB and Mime-Fed with skip synchronization of $\hat{v}$: the global $\hat{v}_t$ is synchronized every $Z = 3$ or 5 rounds.

considerably worse than Fed-LAMB. We report the test accuracy at the end of training in Table 1. Fed-LAMB achieves the highest accuracy on both datasets. Mime-LAMB also substantially improves Mime.

Skip synchronization. In Figure 5, we further present the results of methods with skip synchronization of $\hat{v}$, where the server updates and broadcasts $\hat{v}$ every $Z = 3, 5$ rounds, instead of in very single round. This reduces the communication cost of transmitting the second moment v by a factor of 3 or 5. We see that, the empirical performance of skip synchronization is similar to the standard design; some times it may converge even faster. Our results demonstrate the efficacy of this more efficient strategy in practice.

5.3 SUMMARY OF EMPIRICAL FINDINGS

Here, we provide a brief summary of our empirical results. On all the datasets, in terms of both convergence and generalization, the primary comparisons between our proposed methods and their baselines appear evident:

Fed-LAMB≈ Mime-LAMB>Fed-AMS≈Mime.

The proposed scheme (with two variants Fed-LAMB and Mime-LAMB) exhibits faster convergence and better generalisation accuracy than recently proposed adaptive FL algorithms. Our results suggest that, using layer-wise acceleration in the local training can speedup the overall model performance of locally adaptive federated learning. Moreover, in practice we may adopt the skip aggregation strategy to further reduce the additional communication required for our proposed approach, without losing utility. As discussed earlier, Mime-LAMB typically requires more gradient computation than Fed-LAMB. Therefore, with similar performance as Mime-LAMB, the Fed-LAMB protocol might be more efficient and convenient in practical applications.

6 CONCLUSION

We study a doubly adaptive method in the particular framework of federated learning (FL). Built upon the acceleration effect of layer-wise learning rate scheduling and of state-of-the-art adaptive gradient methods, we derive a locally layer-wise FL framework that performs local updates using adaptive AMSGrad on each worker and periodically averages local models stored on each device. The core of our Fed-LAMB scheme, is to speedup up local training by adopting layer-wise adaptive learning rates. To our knowledge, this is the first FL algorithm in literature that possess both the *dimension-wise* adaptivity (by AMSGrad) and *layer-wise* adaptivity (by layer-wise adjusted learning rate).

We provide the convergence analysis of Fed-LAMB that matches many existing methods, with a linear speedup against the number of clients. We also provide a skip aggregation trick to further reduce the communication overhead. Extensive experiments on various datasets and models, under both IID and non-IID data settings, validate that both Fed-LAMB and Mime-LAMB are able to provide faster convergence which in turn leads to reduced communication and training time to reach a certain accuracy. In many cases, our framework also improves the overall performance of federated learning over prior methods.

Adaptive FL (at central server) with communication compression has been studied in [Li et al., 2022, Li and Li, 2023]. In the future, we may also study Fed-LAMB type locally adaptive algorithms with communication compression.

References

Kallista A. Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In *Proceedings of Machine Learning and Systems (MLSys)*, Stanford, CA, 2019.

Zachary Charles, Zachary Garrett, Zhouyuan Huo, Sergei Shmulyian, and Virginia Smith. On large-cohort training for federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 20461–20475, virtual, 2021.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of A class of adam-type algorithms for non-convex optimization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, 2019.

Xiangyi Chen, Xiaoyun Li, and Ping Li. Toward communication efficient adaptive gradient method. In *Proceedings of the ACM-IMS Foundations of Data Science Conference (FODS)*, pages 119–128, Virtual Event, USA, 2020.

Xiangyi Chen, Belhal Karimi, Weijie Zhao, and Ping Li. On the convergence of decentralized adaptive gradient methods. In *Proceedings of The 14th Asian Conference on Machine Learning (ACML)*, Hyderabad, India, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, FL, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, Minneapolis, MN, 2019.

Timothy Dozat. Incorporating nesterov momentum into Adam. In *Proceedings of the 4th International Conference on Learning Representations (ICLR Workshop)*, San Juan, Puerto Rico, 2016.

John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, 2016.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4519–4529, Online [Palermo, Sicily, Italy], 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6341–6345, Brighton, UK, 2019.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.*, 37(3):50–60, 2020a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, Austin, TX, 2020b.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020c.

Xiaoyun Li and Ping Li. Analysis of error feedback in federated non-convex optimization with biased compression: Fast convergence and partial participation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, Honolulu, HI, 2023.

Xiaoyun Li, Belhal Karimi, and Ping Li. On distributed adaptive optimization with gradient compression. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*, Virtual Event, 2022.

Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local SGD with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, 2019.

Brendan McMahan and Matthew J. Streeter. Adaptive bound optimization for online convex optimization. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pages 244–256, Haifa, Israel, 2010.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, Fort Lauderdale, FL, 2017.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. 2003.

Solmaz Niknam, Harpreet S. Dhillon, and Jeffrey H. Reed. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Commun. Mag.*, 58(6):46–51, 2020.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Virtual Event, Austria, 2021.

Tijmen Tieleman and Geoffrey Hinton. RmsProp: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Networks Mach. Learn*, 17, 2012.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael G. Rabbat. SlowMo: Improving communication-efficient distributed SGD with slow momentum. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

Jun-Kun Wang, Xiaoyun Li, Belhal Karimi, and Ping Li. An optimistic acceleration of AMSGrad for nonconvex optimization. In *Proceedings of the Asian Conference on Machine Learning (ACML)*, pages 422–437, Virtual Event, 2021.

Blake E. Woodworth, Kumar Kshitij Patel, Sebastian U. Stich, Zhen Dai, Brian Bullins, H. Brendan McMahan, Ohad Shamir, and Nathan Srebro. Is local SGD better than minibatch SGD? In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 10334–10343, Virtual Event, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jie Xu, Benjamin S. Glicksberg, Chang Su, Peter B. Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *J. Heal. Informatics Res.*, 5(1):1–19, 2021.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):12:1–12:19, 2019.

Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, and Kurt Keutzer. Imagenet training in minutes. In *Proceedings of the 47th International Conference on Parallel Processing (ICPP)*, pages 1:1–1:10, Eugene, OR, 2018.

Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 7184–7193, Long Beach, CA, 2019.

Xiaotong Yuan and Ping Li. On convergence of FedProx: Local dissimilarity invariant bounds, non-smoothness and beyond. In *Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, 2022.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Weijie Zhao, Xuewu Jiao, Mingqing Hu, Xiaoyun Li, Xiangyu Zhang, and Ping Li. PaddleBox: Communication-efficient terabyte-scale model training framework for online advertising. In *Proceedings of the IEEE International Conference on Big Data (IEEE BigData)*, pages 1401–1408, Osaka, Japan, 2022.

Dongruo Zhou, Jinghui Chen, Yuan Cao, Yiqi Tang, Ziyan Yang, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.

Yingxue Zhou, Belhal Karimi, Jinxing Yu, Zhiqiang Xu, and Ping Li. Towards better generalization of adaptive gradient methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, virtual, 2020.

A EXPERIMENT DETAILS AND RESULTS

A.1 THE ADP-FED ALGORITHM

The Adp-Fed (Adaptive Federated Optimization) is one of the baseline methods compared with Fed-LAMB in our paper. The algorithm is given in Algorithm 2. The key difference between Adp-Fed and Fed-AMS [Chen et al., 2020] is that, in Adp-Fed, each client runs local SGD (Line 8), and an Adam optimizer is maintained for the global adaptive optimization (Line 15). In the Fed-AMS framework (as well as our Fed-LAMB), each clients runs local (adaptive) AMSGrad method, and the global model is simply obtained by averaging the local models. [Li and Li, 2023] proposed a variant of Adp-Fed algorithm with communication compression.

Algorithm 2 Adp-Fed: Adaptive Federated Optimization [Reddi et al., 2021]

```
1: **Input**: parameter $0 < \beta_1, \beta_2 < 1$, and learning rate $\alpha_t$, weight decaying parameter $\lambda \in [0, 1]$.
2: **Initialize**: $\theta_{0,i} \in \Theta \subseteq \mathbb{R}^d$, $m_0 = 0$, $v_0 = \epsilon$, $\forall i \in [\![n]\!]$, and $\theta_0 = \frac{1}{n}\sum_{i=1}^n \theta_{0,i}$.
3: **for** $r = 1, \ldots, R$ **do**
4:     **parallel for device** $i$ **do**:
5:         Set $\theta_{r,i}^0 = \theta_{r-1}$.
6:         **for** $t = 1, \ldots, T$ **do**
7:             Compute stochastic gradient $g_{r,i}^t$ at $\theta_{r,i}^0$.
8:             $\theta_{r,i}^t = \theta_{r,i}^{t-1} - \eta_l g_{r,i}^t$
9:         **end for**
10:        Devices send $\triangle_{r,i} = \theta_{r,i}^T - \theta_{r,i}^0$ to server.
11:    **end for**
12:    Server computes $\bar{\triangle}_r = \frac{1}{n}\sum_{i=1}^n \triangle_{r,i}$
13:    $m_r = \beta_1 m_{r-1} + (1 - \beta_1)\bar{\triangle}_r$
14:    $v_r = \beta_2 v_{r-1} + (1 - \beta_2)\bar{\triangle}_r^2$
15:    $\theta_r = \theta_{r-1} + \eta_g \frac{m_r}{\sqrt{v_r}}$
16: **end for**
17: **Output**: Global model parameter $\theta_R$.
```

A.2 HYPER-PARAMETER TUNING

In our empirical study, we tune the learning rate of each algorithm carefully such that the best performance is achieved. The search grids in all our experiments are provided in Table 2.

Table 2: Search grids of the learning rate.

	Learning rate range
Fed-SGD	$[0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5]$
Fed-AMS	$[0.0001, 0.0003, 0.0005, 0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1]$
Fed-LAMB	$[0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5]$
Adp-Fed	Local η_l: $[0.0001, 0.0003, 0.0005, 0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5]$ Global η_g: $[0.0001, 0.0003, 0.0005, 0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1]$
Mime	$[0.0001, 0.0003, 0.0005, 0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1]$
Mime-LAMB	$[0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5]$

A.3 MORE WORKERS

In Figure 6, we provide additional figures with larger number of workers $n = 200$, on MNIST and FMNIST with non-IID data. The conclusions stay the same: we see that the proposed Fed-LAMB and Mime-LAMB perform much better than the baseline algorithms, with faster convergence and better accuracy at the end of 100 FL training rounds.

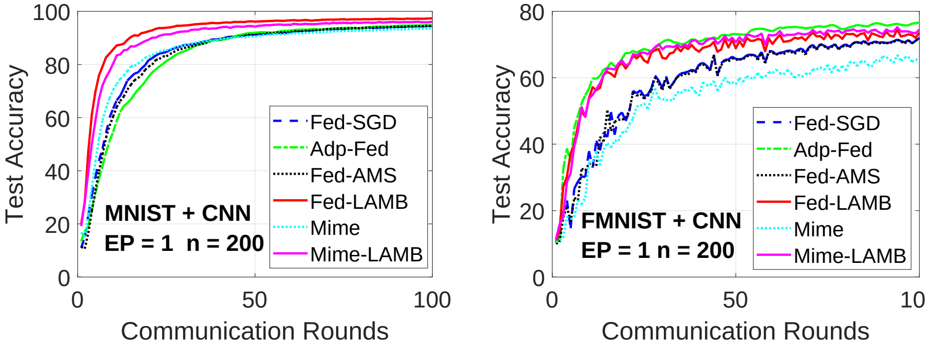

Figure 6: Test accuracy on MNIST and FMNIST with $n = 200$ workers, full participation, local batch size 64. Data are non-IID distributed among clients.

B THEORETICAL ANALYSIS

We first recall in Table 3 some important notations that will be used in our following analysis.

R, T	:=	Number of communications rounds and local iterations (resp.)
n, D, i	:=	Total number of clients, portion sampled uniformly and client index
h, ℓ	:=	Total number of layers in the DNN and its index
$\phi(\cdot)$	:=	Scaling factor in Fed-LAMB update
$\overline{\theta}$	:=	Global model (after periodic averaging)
$\psi_{r,i}^t$	:=	ratio computed at round r, local iteration t and for device i. $\psi_{r,i}^{\ell,t}$ denotes its component at layer ℓ

Table 3: Summary of notations used in the paper.

We now provide the proofs for the theoretical results of the main paper, including the intermediary Lemmas and the main convergence result, Theorem 5.

B.1 INTERMEDIARY LEMMA

We now develop the proof of the convergence rate of Fed-LAMB. We need a supporting Lemma 8 for this.

Lemma 8. *Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 1. Then for $i \in [\![n]\!]$:*

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

where ϕ_M is defined in Assumption 4 and p is the total number of dimensions $p = \sum_{\ell=1}^{\mathsf{h}} p_\ell$.

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

$$\theta^\ell_{r,i} = \overline{\theta_r}^\ell - \alpha \sum_{t=1}^{T} \phi(\|\theta^{\ell,t-1}_{r,i}\|)\psi^j_{r,i}/\|\psi^\ell_{r,i}\| = \overline{\theta_r}^\ell - \alpha \sum_{t=1}^{T} \phi(\|\theta^{\ell,t-1}_{r,i}\|)\frac{m^t_{r,i}}{\sqrt{v^t_r}}\frac{1}{\|\psi^\ell_{r,i}\|}$$

leading to

$$\|\overline{\theta_r} - \theta_{r,i}\|^2 = \sum_{\ell=1}^{\mathsf{h}} \|\overline{\theta_r}^\ell - \theta^\ell_{r,i}\|^2 \leq \alpha^2 M^2 T^2 \phi_M^2 \frac{(1-\beta_2)p}{\epsilon} \ ,$$

which concludes the proof. □

B.2 PROOF OF THEOREM 5

Theorem. Suppose **Assumption 1 - Assumption 4** holds. Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 1 with a constant learning rate α. Let the number of local epochs be $T \geq 1$ and $\lambda = 0$. Then, for any round $R > 0$, we have

$$\begin{aligned}\frac{1}{R}\sum_{r=1}^{R} \mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2\right] \leq \sqrt{\frac{M^2 p}{n}}\frac{\triangle}{\mathsf{h}\alpha R} + \frac{4\alpha^2 L M^2 T^2 \phi_M^2 (1-\beta_2)p}{\sqrt{\epsilon}} \\ + 4\alpha^2 \frac{M^2}{\sqrt{\epsilon}} + \frac{\phi_M \sigma^2}{Rn}\sqrt{\frac{1-\beta_2}{M^2 p}} + 4\alpha\left[\phi_M \frac{\mathsf{h}\sigma^2}{\sqrt{n}}\right] + 4\alpha^2\left[\phi_M^2\sqrt{M^2 + p\sigma^2}\right] ,\end{aligned} \tag{6}$$

where $\triangle = \mathbb{E}[f(\bar{\theta}_1)] - \min_{\theta\in\Theta} f(\theta)$.

Proof. Our proof will make use of an intermediary virtual sequence defined as

$$\bar{\vartheta}_r = \bar{\theta}_r + \frac{\beta_1}{1-\beta_1}(\bar{\theta}_r - \bar{\theta}_{r-1}) \ , \tag{7}$$

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

$$\begin{aligned}\bar{\vartheta}^\ell_{r+1} - \bar{\vartheta}^\ell_r &= \frac{1}{1-\beta_1}(\bar{\theta}^\ell_{r+1} - \bar{\theta}^\ell_r) - \frac{\beta_1}{1-\beta_1}(\bar{\theta}^\ell_r - \bar{\theta}^\ell_{r-1}) \\ &= \frac{\alpha_r}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta^\ell_{r,i}\|)}{\|\psi^\ell_{r,i}\|}\psi^\ell_{r,i} - \frac{\alpha_{r-1}}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta^\ell_{r-1,i}\|)}{\|\psi^\ell_{r-1,i}\|}\psi^\ell_{r-1,i} \\ &= \frac{\alpha\beta_1}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|\psi^\ell_{r,i}\|} - \frac{\phi(\|\theta^\ell_{r-1,i}\|)}{\sqrt{v^t_{r-1}}\|\psi^\ell_{r-1,i}\|}\right)m^t_{r-1} + \frac{\alpha}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta^\ell_{r,i}\|)}{\sqrt{v^t_r}\|\psi^\ell_{r,i}\|}g^t_{r,i} \ ,\end{aligned} \tag{8}$$

where we have assumed a constant learning rate α.

Using Assumption 1, we have

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

Taking expectations on both sides leads to

$$-\mathbb{E}[\left\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \right\rangle] \leq \mathbb{E}[f(\bar{\vartheta}_r) - f(\bar{\vartheta}_{r+1})] + \sum_{\ell=1}^{L}\frac{L_\ell}{2}\mathbb{E}[\|\bar{\vartheta}^\ell_{r+1} - \bar{\vartheta}^\ell_r\|^2] \ . \tag{9}$$

We note for all $\theta \in \Theta$, the majorant $G > 0$ such that $\phi(\|\theta\|) \leq G$. Then, following (9), we obtain

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

Developing the LHS of (10) using (8) leads to

$$\begin{aligned}
\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \rangle &= \sum_{\ell=1}^{\mathsf{h}} \sum_{j=1}^{p_\ell} \nabla_\ell f(\bar{\vartheta}_r)^j (\bar{\vartheta}_{r+1}^{\ell,j} - \bar{\vartheta}_r^{\ell,j}) \\
&= \frac{\alpha\beta_1}{1-\beta_1} \frac{1}{n} \sum_{\ell=1}^{\mathsf{h}} \sum_{j=1}^{p_\ell} \nabla_\ell f(\bar{\vartheta}_r)^j \left[\sum_{i=1}^{n} \left(\frac{\phi(\|\theta_{r,i}^\ell\|)}{\sqrt{v_r^t}\|\psi_{r,i}^\ell\|} - \frac{\phi(\|\theta_{r-1,i}^\ell\|)}{\sqrt{v_{r-1}^t}\|\psi_{r-1,i}^\ell\|} \right) m_{r-1}^t \right] \\
&\underbrace{- \frac{\alpha}{n} \sum_{\ell=1}^{\mathsf{h}} \sum_{j=1}^{p_\ell} \nabla_\ell f(\bar{\vartheta}_r)^j \sum_{i=1}^{n} \frac{\phi(\|\theta_{r,i}^\ell\|)}{\sqrt{v_r^t}\|\psi_{r,i}^\ell\|} g_{r,i}^{t,l,j}}_{=A_1} \;.
\end{aligned} \tag{11}$$

Suppose T is the total number of local iterations and R is the number of rounds. We can write (11) as

$$A_1 = -\alpha \langle \nabla f(\bar{\vartheta}_r), \frac{\bar{g}_r}{\sqrt{\hat{v}_r}} \rangle,$$

where $\bar{g}_r = \frac{1}{n}\sum_{i=1}^n \bar{g}_{t,i}$, with $\bar{g}_{t,i} = \left[\frac{\phi(\|\theta_{t,i}^1\|)}{\|\psi_{t,i}^1\|} g_{t,i}^1, ..., \frac{\phi(\|\theta_{t,i}^L\|)}{\|\psi_{t,i}^L\|} g_{t,i}^L \right]$ representing the normalized gradient (concatenated by layers) of the i-th device. It holds that

$$\langle \nabla f(\bar{\vartheta}_r), \frac{\bar{g}_r}{\sqrt{\hat{v}_r}} \rangle = \frac{1}{2}\|\frac{\nabla f(\bar{\vartheta}_r)}{\hat{v}_r^{1/4}}\|^2 + \frac{1}{2}\|\frac{\bar{g}_r}{\hat{v}_r^{1/4}}\|^2 - \|\frac{\nabla f(\bar{\vartheta}_r) - \bar{g}_r}{\hat{v}_r^{1/4}}\|^2. \tag{12}$$

To bound the last term on the RHS, we have

$$\begin{aligned}
\|\frac{\nabla f(\bar{\vartheta}_r) - \bar{g}_r}{\hat{v}_r^{1/4}}\|^2 = \|\frac{\frac{1}{n}\sum_{i=1}^n (\nabla f(\bar{\vartheta}_r) - \bar{g}_{t,i})}{\hat{v}_r^{1/4}}\|^2 &\leq \frac{1}{n}\sum_{i=1}^{n} \|\frac{\nabla f(\bar{\vartheta}_r) - \bar{g}_{t,i}}{\hat{v}_r^{1/4}}\|^2 \\
&\leq \frac{2}{n}\sum_{i=1}^{n} \Big(\|\frac{\nabla f(\bar{\vartheta}_r) - \nabla f(\bar{\theta}_r)}{\hat{v}_r^{1/4}}\|^2 + \|\frac{\nabla f(\bar{\theta}_r) - \bar{g}_{t,i}}{\hat{v}_r^{1/4}}\|^2 \Big).
\end{aligned}$$

By Lipschitz smoothness of the loss function, the first term admits

$$\begin{aligned}
\frac{2}{n}\sum_{i=1}^{n} \|\frac{\nabla f_i(\bar{\vartheta}_r) - \nabla f_i(\bar{\theta}_r)}{\hat{v}_r^{1/4}}\|^2 \leq \frac{2}{n\sqrt{\epsilon}} \sum_{i=1}^{n} L_\ell \|\bar{\vartheta}_r - \bar{\theta}_r\|^2 &= \frac{2L_\ell}{n\sqrt{\epsilon}} \frac{\beta_1^2}{(1-\beta_1)^2} \sum_{i=1}^{n} \|\bar{\theta}_r - \bar{\theta}_{t-1}\|^2 \\
&\leq \frac{2\alpha^2 L_\ell}{n\sqrt{\epsilon}} \frac{\beta_1^2}{(1-\beta_1)^2} \sum_{l=1}^{L}\sum_{i=1}^{n} \|\frac{\phi(\|\theta_{t,i}^l\|)}{\|\psi_{t,i}^l\|} \psi_{t,i}^l\|^2 \\
&\leq \frac{2\alpha^2 L_\ell p \phi_M^2}{\sqrt{\epsilon}} \frac{\beta_1^2}{(1-\beta_1)^2}.
\end{aligned}$$

For the second term,

$$\frac{2}{n}\sum_{i=1}^{n} \|\frac{\nabla f(\bar{\theta}_r) - \bar{g}_{t,i}}{\hat{v}_r^{1/4}}\|^2 \leq \frac{4}{n}\Big(\underbrace{\sum_{i=1}^{n} \|\frac{\nabla f(\bar{\theta}_r) - \nabla f(\theta_{t,i})}{\hat{v}_r^{1/4}}\|^2}_{B_1} + \underbrace{\sum_{i=1}^{n} \|\frac{\nabla f(\theta_{t,i}) - \bar{g}_{t,i}}{\hat{v}_r^{1/4}}\|^2}_{B_2} \Big). \tag{13}$$

Using the smoothness of f_i we can transform B_1 into consensus error by

$$\begin{aligned}
B_1 \leq \frac{L}{\sqrt{\epsilon}} \sum_{i=1}^{n} \|\bar{\theta}_r - \theta_{t,i}\|^2 &= \frac{\alpha^2 L}{\sqrt{\epsilon}} \sum_{i=1}^{n}\sum_{l=1}^{L} \| \sum_{j=\lfloor t \rfloor_r + 1}^{t} \Big(\frac{\phi(\|\theta_{j,i}^l\|)}{\|\psi_{j,i}^l\|} \psi_{j,i}^l - \frac{1}{n}\sum_{k=1}^{n} \frac{\phi(\|\theta_{j,k}^l\|)}{\|\psi_{j,k}^l\|} \psi_{j,k}^l \Big) \|^2 \\
&\leq n \frac{\alpha^2 L}{\sqrt{\epsilon}} M^2 T^2 \phi_M^2 (1-\beta_2) p,
\end{aligned} \tag{14}$$

where the last inequality stems from Lemma 8 in the particular case where $\theta_{t,i}$ are averaged every $ct+1$ local iterations for any integer c, since $(t-1)-(\lfloor t \rfloor_r + 1) + 1 \leq T-1$.

We now bound B_2 (under the simplification that $\beta_1 = 0$):

$$\begin{aligned}
\mathbb{E}[B_2] &= \mathbb{E}[\sum_{i=1}^{n} \| \frac{\nabla f(\theta_{t,i}) - \bar{g}_{t,i}}{\hat{v}_r^{1/4}} \|^2] \\
&\leq \frac{nM^2}{\sqrt{\epsilon}} + n\phi_M^2 \sqrt{M^2 + p\sigma^2} - 2\sum_{i=1}^{n} \mathbb{E}[\langle \nabla f(\theta_{t,i}), \bar{g}_{t,i} \rangle / \sqrt{\hat{v}_r}] \\
&= \frac{nM^2}{\sqrt{\epsilon}} + n\phi_M^2 \sqrt{M^2 + p\sigma^2} - 2\sum_{i=1}^{n} \sum_{\ell=1}^{L} \mathbb{E}[\langle \nabla_\ell f(\theta_{t,i}), \frac{\phi(\|\theta_{t,i}^l\|)}{\|\psi_{t,i}^l\|} g_{t,i}^l \rangle / \sqrt{\hat{v}_r^l}] \\
&= \frac{nM^2}{\sqrt{\epsilon}} + n\phi_M^2 \sqrt{M^2 + p\sigma^2} - 2\sum_{i=1}^{n} \sum_{l=1}^{L} \sum_{i=1}^{p_l} \mathbb{E}[\nabla_l f(\theta_{t,i})^j \frac{\phi(\|\theta_{t,i}^{l,j}\|)}{\sqrt{\hat{v}_r^{l,j}} \|\psi_{t,i}^{l,j}\|} g_{t,i}^{l,j}] \\
&\leq \frac{nM^2}{\sqrt{\epsilon}} + n\phi_M^2 \sqrt{M^2 + p\sigma^2} - 2\sum_{i=1}^{n} \sum_{l=1}^{L} \sum_{i=1}^{p_l} \mathbb{E}\left[\sqrt{\frac{1-\beta_2}{M^2 p_\ell}} \phi(\|\theta_{r,i}^{l,j}\|) \nabla_l f(\theta_{t,i})^j g_{t,i}^{l,j} \right] \\
&\quad - 2\sum_{i=1}^{n} \sum_{l=1}^{L} \sum_{j=1}^{p_l} E\left[\left(\phi(\|\theta_{r,i}^{l,j}\|) \nabla_l f(\theta_{t,i})^j \frac{g_{r,i}^{t,l,j}}{\|\psi_{r,i}^{l,j}\|} \right) 1 \left(\text{sign}(\nabla_l f(\theta_{t,i})^j \neq \text{sign}(g_{r,i}^{t,l,j}) \right) \right],
\end{aligned}$$

where we use Assumption 2, Assumption 3 and Assumption 4. Yet,

$$\begin{aligned}
-\mathbb{E}&\left[\left(\phi(\|\theta_{r,i}^{l,j}\|) \nabla_l f(\theta_{t,i})^j \frac{g_{r,i}^{t,l,j}}{\|\psi_{r,i}^{l,j}\|} \right) 1 \left(\text{sign}(\nabla_l f(\theta_{t,i})^j \neq \text{sign}(g_{r,i}^{t,l,j}) \right) \right] \\
&\leq \phi_M \nabla_l f(\theta_{t,i})^j \mathbb{P}\left[\text{sign}(\nabla_l f(\theta_{t,i})^j \neq \text{sign}(g_{r,i}^{t,l,j}) \right].
\end{aligned}$$

Then we have

$$\mathbb{E}[B_2] \leq \frac{nM^2}{\sqrt{\epsilon}} + n\phi_M^2 \sqrt{M^2 + p\sigma^2} - 2\phi_m \sqrt{\frac{1-\beta_2}{M^2 p}} \sum_{i=1}^{n} \mathbb{E}[\|[\nabla f(\theta_{t,i})\|^2] + \phi_M \frac{\mathsf{h}\sigma^2}{\sqrt{n}}$$

Thus, (13) becomes

$$\frac{2}{n} \sum_{i=1}^{n} \| \frac{\nabla f_i(\bar{\theta}_r) - \bar{g}_{t,i}}{\hat{v}_r^{1/4}} \|^2 \leq 4 \left[\frac{\alpha^2 L_\ell}{\sqrt{\epsilon}} \alpha^2 M^2 T^2 \phi_M^2 (1-\beta_2) p + \frac{\alpha M^2}{\sqrt{\epsilon}} + \phi_M^2 \sqrt{M^2 + p\sigma^2} + \alpha\phi_M \frac{\mathsf{h}\sigma^2}{\sqrt{n}} \right]$$

Substituting all ingredients into (12), we obtain

$$\begin{aligned}
-\alpha\mathbb{E}[\langle \nabla f(\bar{\vartheta}_r), \frac{\bar{g}_r}{\sqrt{\hat{v}_r}} \rangle] &\leq -\frac{\alpha}{2} \mathbb{E}[\| \frac{\nabla f(\bar{\vartheta}_r)}{\hat{v}_r^{1/4}} \|^2] - \frac{\alpha}{2} \mathbb{E}[\| \frac{\bar{g}_r}{\hat{v}_r^{1/4}} \|^2] + \frac{2\alpha^3 L_\ell p \phi_M^2}{\sqrt{\epsilon}} \frac{\beta_1^2}{(1-\beta_1)^2} \\
&\quad + 4\alpha \left[\frac{\alpha^2 L}{\sqrt{\epsilon}} M^2 T^2 \phi_M^2 (1-\beta_2) p + \frac{\alpha M^2}{\sqrt{\epsilon}} + \phi_M^2 \sqrt{M^2 + p\sigma^2} + \alpha\phi_M \frac{\mathsf{h}\sigma^2}{\sqrt{n}} \right].
\end{aligned}$$

To bound the second term on the RHS in above, we notice that

$$\begin{aligned}
\mathbb{E}[\| \frac{\bar{g}_r}{\hat{v}_r^{1/4}} \|^2] = \frac{1}{n^2} \mathbb{E}[\| \frac{\sum_{i=1}^{n} \bar{g}_{r,i}}{\hat{v}_r^{1/4}} \|^2] &= \frac{1}{n^2} \mathbb{E}[\sum_{l=1}^{L} \sum_{i=1}^{n} \| \frac{\phi(\|\theta_{r,i}^l\|)}{\hat{v}^{1/4} \|\psi_{r,i}^l\|} g_{r,i}^l \|^2] \\
&\geq \phi_m^2 (1-\beta_2) \mathbb{E}\left[\| \frac{1}{n} \sum_{i=1}^{n} \frac{\nabla f(\theta_{r,i})}{\hat{v}_r^{1/4}} \|^2 \right] \qquad (15) \\
&= \phi_m^2 (1-\beta_2) \mathbb{E}\left[\| \frac{\overline{\nabla} f(\theta_r)}{\hat{v}_r^{1/4}} \|^2 \right].
\end{aligned}$$

Regarding $\left\|\frac{\overline{\nabla} f(\theta_r)}{\hat{v}_r^{1/4}}\right\|^2$, we have

$$
\begin{aligned}
\left\|\frac{\overline{\nabla} f(\theta_r)}{\hat{v}_r^{1/4}}\right\|^2 &\geq \frac{1}{2}\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2 - \left\|\frac{\overline{\nabla} f(\theta_r) - \nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2 \\
&\geq \frac{1}{2}\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2 - \left\|\frac{\frac{1}{n}\sum_{i=1}^n (\nabla f_i(\theta_r) - \nabla f(\bar{\theta}_r))}{\hat{v}_r^{1/4}}\right\|^2 \\
&\geq \frac{1}{2}\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2 - \frac{\alpha^2 L_\ell}{\sqrt{\epsilon}} M^2 T^2 (\sigma^2 + G^2)(1-\beta_2)p,
\end{aligned}
$$

where the last line is due to (14) and Assumption 3. Therefore, we have obtained

$$
\begin{aligned}
A_1 \leq& -\frac{\alpha\phi_m^2(1-\beta_2)}{4}\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2 + \frac{\alpha^3 L_\ell}{\sqrt{\epsilon}} M^2 T^2 \phi_m^2 \phi_M^2 (1-\beta_2)^2 p + \frac{2\alpha^3 L_\ell p \phi_M^2}{\sqrt{\epsilon}} \frac{\beta_1^2}{(1-\beta_1)^2} \\
&+ 4\alpha\left[\frac{\alpha^2 L}{\sqrt{\epsilon}} M^2 T^2 (\sigma^2 + G^2)(1-\beta_2)p + \frac{M^2\alpha}{\sqrt{\epsilon}} + \alpha\phi_M^2\sqrt{M^2 + p\sigma^2} + \phi_M \alpha \frac{\mathsf{h}\sigma^2}{\sqrt{n}}\right], \\
\leq& -\frac{\alpha\phi_m^2(1-\beta_2)}{4}\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2 + \frac{\alpha^3 L_\ell}{\sqrt{\epsilon}} M^2 T^2 \phi_m^2 \phi_M^2 (1-\beta_2)^2 p + \frac{2\alpha^3 L_\ell p \phi_M^2}{\sqrt{\epsilon}} \frac{\beta_1^2}{(1-\beta_1)^2} \\
&+ 4\alpha\Big[\frac{\alpha^2 L}{\sqrt{\epsilon}} M^2 T^2 G^2 (1-\beta_2)p + \frac{M^2\alpha}{\sqrt{\epsilon}} + \alpha\phi_M^2\sqrt{M^2 + p\sigma^2} \\
&\qquad + \sigma^2\left(\frac{\alpha^2 L}{\sqrt{\epsilon}} M^2 T^2 (1-\beta_2)p + \phi_M \alpha \frac{\mathsf{h}}{\sqrt{n}}\right)\Big].
\end{aligned}
$$

Substitute back into (11), assuming $M \leq 1$, we have the following by taking the telescope sum

$$
\begin{aligned}
&\frac{1}{R}\sum_{t=1}^R \mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2\right] \\
&\lesssim \sqrt{\frac{M^2 p}{n}} \frac{f(\bar{\vartheta}_1) - \mathbb{E}[f(\bar{\vartheta}_{R+1})]}{\mathsf{h}\alpha R} + \frac{\alpha}{n^2}\sum_{r=1}^R\sum_{i=1}^n \sigma_i^2 \mathbb{E}\left[\left\|\frac{\phi(\|\theta_{r,i}^\ell\|)}{\sqrt{v_r}\|\psi_{r,i}^\ell\|}\right\|^2\right] + \frac{2\alpha^3 \overline{L} p \phi_M^2}{\sqrt{\epsilon}} \frac{\beta_1^2}{(1-\beta_1)^2} \\
&+ 4\Big[\frac{\alpha^2 \overline{L}}{\sqrt{\epsilon}} M^2 T^2 G^2 (1-\beta_2)p + \frac{\alpha M^2}{\sqrt{\epsilon}} + \alpha\phi_M^2\sqrt{M^2 + p\sigma^2} \\
&+ \sigma^2\left(\frac{\alpha^2 \overline{L}}{\sqrt{\epsilon}} M^2 T^2 (1-\beta_2)p + \phi_M \alpha \frac{\mathsf{h}}{\sqrt{n}}\right)\Big] + \frac{\alpha\beta_1}{1-\beta_1}\sqrt{(1-\beta_2)p}\frac{\mathsf{h}M^2}{\sqrt{\epsilon}} + \overline{L}\alpha^2 M^2 \phi_M^2 \frac{(1-\beta_2)p}{T\epsilon} \\
&\leq \sqrt{\frac{M^2 p}{n}} \frac{\mathbb{E}[f(\bar{\theta}_1)] - \min_{\theta\in\Theta} f(\theta)}{\mathsf{h}\alpha R} + \frac{\phi_M \sigma^2}{Rn}\sqrt{\frac{1-\beta_2}{M^2 p}} \\
&+ 4\Big[\frac{\alpha^2 \overline{L}}{\sqrt{\epsilon}} M^2 T^2 G^2 (1-\beta_2)p + \frac{M^2\alpha}{\sqrt{\epsilon}} + \phi_M^2 \alpha\sqrt{M^2 + p\sigma^2} \\
&+ \sigma^2\Big(\frac{\alpha^2 \overline{L}}{\sqrt{\epsilon}} M^2 T^2 (1-\beta_2)p + \phi_M \frac{\mathsf{h}}{\sqrt{n}}\Big)\Big] + \frac{\alpha\beta_1}{1-\beta_1}\sqrt{(1-\beta_2)p}\frac{\mathsf{h}M^2}{\sqrt{\epsilon}} \\
&\qquad\qquad + \overline{L}\alpha^2 M^2 \phi_M^2 \frac{(1-\beta_2)p}{T\epsilon} + \frac{2\alpha^3 \overline{L} p \phi_M^2}{\sqrt{\epsilon}} \frac{\beta_1^2}{(1-\beta_1)^2}.
\end{aligned}
$$

Organizing terms, we conclude the proof. □