Layer-wise and Dimension-wise 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. 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. Our framework can be applied to many locally adaptive FL methods, including two recent algorithms, Mime (Karimireddy et al., 2020) and Fed-AMS (Chen et al., 2020). Theoretically, we provide a thorough finite-time convergence analysis of our layer-wise FL methods, coined Fed-LAMB and Mime-LAMB, which demonstrates the acceleration effect of layer-wise adaptivity in FL, leading to improved communication efficiency compared with the baseline method. Experimental results on various datasets and models, under both iid and non-iid settings, show that both Fed-LAMB and Mime-LAMB achieve faster convergence speed and better generalization performance, compared to the state-of-the-art.

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. While (1) reminds that of standard distributed optimization, the principle and setting of FL are different from the classical distributed paradigm: (i) Local updates: FL allows clients to perform multiple updates on the local models before the global aggregation; (ii) Data heterogeneity: in FL, the local data distributions $\mathcal{X}_i$ are usually different across the workers, hindering the convergence of the global model. FL aims at finding a solution of (1) in the fewest number of communication rounds.

One of the most popular framework 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 update rule of Adam reads as

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

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.,

[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.

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).

In this paper, we focus on improving adaptive FL algorithms. 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 ratio $m_t/\sqrt{v_t}$ is further normalized by the weight of the deep network, respectively for each layer. 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). Inspired by the acceleration effect of LAMB, 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. 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 federated learning, following a principled 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 state-of-the-art methods in federated learning. Moreover, we show that it also improves the theoretical communication efficiency compared with the baseline, namely Fed-AMS (Chen et al., 2020).
- We empirically compare various FL methods under both homogeneous and heterogeneous data setting on various benchmark datasets. Our results confirm the accelerated 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 and Related Work

Below, we summarize some relevant works 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, 2004) 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 described in Reddi et al. (2018), forces v to be monotone to fix the theoretical convergence issue. The convergence and generalization of adaptive methods have been studied in, e.g., (Zhou et al., 2018b; Chen et al., 2019; Zhou et al., 2020).

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 preservative 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 analogue layer-wise adaptive variant of Adam. The update rule of LAMB for the ℓ-th layer can be expressed as

$$\theta_t^\ell = \theta_{t-1}^\ell - \frac{\alpha\|\theta_{t-1}^\ell\|}{\|\psi_t^\ell\|}\psi_t^\ell, \text{ with } \psi_t^\ell = m_t^\ell/\sqrt{v_t^\ell},$$

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. It was shown in You et al. (2020) that LAMB can significantly accelerate the convergence of Adam, allowing the use of large mini-batch size with much fewer training iterations.

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 (FL) (Konečnỳ 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. (2020b); Woodworth et al. (2020); Wang et al. (2020).

Recently, several works have considered integrating adaptive gradient methods with FL. Reddi et al. (2021) proposed Adp-Fed where the central server applies Adam-type updates. However, the local clients still perform SGD updates. Chen et al. (2020); Karimireddy et al. (2020) proposed Fed-AMS and Mime[1], 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 rate): in Fed-AMS, a global v is computed and synchronized in each round by averaging the v_i's, $i = 1, ..., n$ from the local clients; in Mime, a global v is directly calculated using full-batches (averaged over all clients) and maintained at the central server. Hence, Mime requires at least twice computation as Fed-AMS. On many tasks, these methods outperform Fed-SGD and other popular methods like SCAFFOLD (Karimireddy et al., 2019) and FedProx (Sahu et al., 2018).

3. Layer-wise 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). For conciseness, we mainly consider AMSGrad (Reddi et al., 2018) as the prototype method. We assume the loss function $f(\cdot)$ is induced by a multi-layer neural network, which includes a broad class of network architectures that can be parameterized layer-by-layer (e.g., MLP, CNN, ResNet, Transformers). Some notations are summarized below.

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. We denote $\theta_{r,i}^{\ell,t}$ as the model parameter of layer ℓ at round r, local iteration t and for worker i.

In general, our proposed algorithm can be viewed as a "federated LAMB". Based on the two recent works regarding locally adaptive FL mentioned above, we present the framework by 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 green (Mime-LAMB) boxes surrounding the text. Both methods use layer-wise adaptive LAMB for local updates (Line 13). The update rule in (3) on local clients 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. The main difference between 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 communication round, clients i communicates the local v_i; the server updates the global $\hat{v}$ by max operation with the averaged v, and sends back the $\hat{v}$.
- 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.

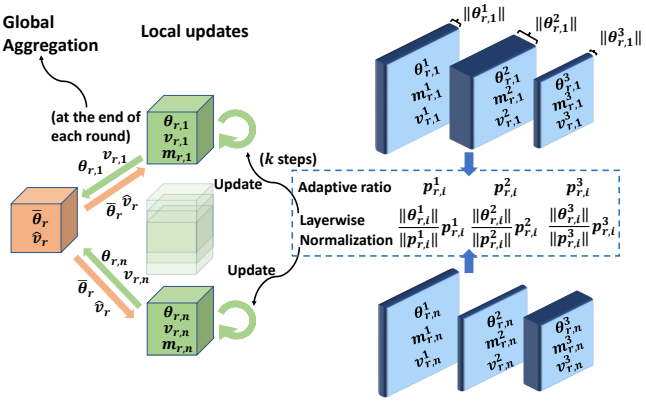

Figure 1: Illustration of Fed-LAMB framework (Algorithm 1), with a three-layer network and $\phi(x) = x$ as an example. For device i and each local iteration in round r, the adaptive ratio of j-th layer $\psi_{r,i}^j$ is normalized according to $\|\theta_{r,i}^j\|$, and then used for updating the local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

Conceptually, both approaches aim at alleviating the impact of data heterogeneity by "globally" reconciling the adaptive learning rates. However, Mime-LAMB needs to calculate the gradients twice, leading to double the computational cost. Moreover, we should note that (3) only uses $\hat{v}_r$, which is fixed per round, hence for all T local updates. We choose this strategy here since it has been shown to perform better than the one allowing $\hat{v}_r^t$ to change during local model training (Karimireddy et al., 2020).

To our knowledge, such two-fold adaptivity (dimension-wise and layer-wise) as in Algorithm 1 has not been con-

[1] "Mime" in our paper is equivalent to "MimeLite" in (Karimireddy et al., 2020). The original "Mime" in (Karimireddy et al., 2020) uses a SVRG-type variance reduction, and has very similar empirical performance as "MimeLite" without SVRG.

Algorithm 1 Fed-LAMB and Mime-LAMB

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

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

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

sidered in federated learning literature before. It turns out that, the significant acceleration of LAMB over Adam in the single-machine setting is also beneficial under the federated settings. Next, we will demonstrate the advantages of our scheme, both theoretically and empirically.

4. Convergence Analysis

We now present the theoretical analysis for Algorithm 1. Based on classical stochastic nonconvex optimization results, we derive a collection of bounds to understand the convergence behavior of our layer-wise adaptive optimization method under the federated learning framework. The main challenges we ought to overcome are: (i) the large amount of decentralized workers working solely on their own data stored locally; (ii) a periodic averaging occurs on the central server pushing each of those clients to send local models after some local iterations; (iii) both *dimension-wise* and *layer-wise* adaptivity. Our analysis encompasses those challenges and leads to an informative convergence rates depending on the quantities of interest: the number of layers of the neural network, the number of communications rounds and the number of clients.

4.1. Finite time analysis of Algorithm 1

In the sequel, the analysis of our scheme we provide is *global*, in the sense that it does not depend on the initialization of our algorithm, and *finite-time*, as in it is true for any arbitrary number of communication rounds. In the context of nonconvex stochastic optimization for federated clients, we need the following common assumptions:

Assumption 4.1. (Smoothness per layer) For $i \in [\![n]\!]$ and $\ell \in [\![L]\!]$: $\left\|\nabla f_i(\theta^\ell) - \nabla f_i(\vartheta^\ell)\right\| \leq L_\ell \left\|\theta^\ell - \vartheta^\ell\right\|$.

Assumption 4.2. (Unbiased and bounded gradient) The stochastic gradient is unbiased $\forall r, t, i$: $\mathbb{E}[g^t_{r,i}] = \nabla f_i(\theta^t_r)$ and bounded by $\left\|g^t_{r,i}\right\| \leq M$.

Assumption 4.3. (Bounded variance) The stochastic gradient admits (*locally*) $\mathbb{E}[|g^j_{r,i} - \nabla f_i(\theta_r)^j|^2] < \sigma^2$, and (*globally*) $\frac{1}{n}\sum_{i=1}^n ||\nabla f_i(\theta_r) - \nabla f(\theta_r)||^2] < G^2$.

Assumption 4.3 characterizes the data heterogeneity on each device. In a classical distributed settings we would have $G = 0$. Also, following You et al. (2020), we use the following assumption on the scaling function ϕ.

Assumption 4.4. (Bounded scale) For any $a \in \mathbb{R}_+$, there exist $\phi_m > 0, \phi_M > 0$ such that $\phi_m \leq \phi(a) \leq \phi_M$.

We now state our main result regarding the non-asymptotic convergence rate of Fed-LAMB and Mime-LAMB in Algorithm 1. The convergence rate for any round reads:

Theorem 4.5. *Under Assumption 4.1-Assumption 4.4, Consider $\{\overline{\theta_r}\}_{r>0}$ obtained from Algorithm 1 with a constant learning rate α. Let $\lambda = 0$. Then, for any round $R > 0$, we have*

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

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

Note that the manifestation of p in the rate is because the variance bound is on each dimension in Assumption 4.3. This dependency on p can be removed when Assumption 4.3 is assumed globally, which is also common in optimization literature. Two important Lemmas are required in the proof of the Theorem above (see complete proof of our bound in the Appendix). The first result gives a characterization of the gap between the averaged model, that is computed by the central server in a periodic manner, and each of the local models stored in each client $i \in [\![n]\!]$.

Lemma 4.6. *Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 1. Then for $i \in [\![n]\!]$ and $r > 0$, the gap $\|\overline{\theta_r} - \theta_{r,i}\|^2$ satisfies:*

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

where ϕ_M is defined in Assumption 4.4, $p = \sum_{\ell=1}^{\mathsf{h}} p_\ell$.

The gap is provably bounded by quantities such as the total dimension of the multi-layered model p, the learning rate and the upper bound of the gradient that we assumed via Assumption 4.2.

Lemma 4.7. *Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of the global model. For $r > 0$:*

$$\left\| \frac{\overline{\nabla f(\theta_r)}}{\sqrt{v_r}} \right\|^2 \geq \frac{1}{2} \left\| \frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r}} \right\|^2 - \overline{L} \alpha^2 M^2 \phi_M^2 \frac{(1-\beta_2)p}{\epsilon},$$

where M is defined in Assumption 4.2 and ϕ_M is defined in Assumption 4.4.

Note that the end goal is to characterize how fast the gradient of the averaged/global parameter $\overline{\theta_r}$ goes to zero, but not the averaged local gradient. Hence, we use Lemma 4.7 to bound the desired suboptimality conditon $\left\| \frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r}} \right\|$. Besides, using a uniform bound on the moment $\|\hat{v}_r\| \leq M^2$ and by choosing a suitable decreasing learning rate, we have the following simplified corollary:

Corollary 4.8. *Under the same setting as Theorem 4.5, with $\alpha = \mathcal{O}(\frac{1}{\sqrt{\mathsf{h}R}})$, it holds that*

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

The leading two terms display a dependence of the convergence rate of Fed-LAMB on the initialization and the variance of the stochastic gradient (see Assumption 4.3), which are common in distributed optimization. The last term involves the number of local updates which relates to the communication efficiency. More discussion will be provided in the sequel.

4.2. Comparisons

We dedicate the following paragraph to a discussion on the bound (and implications) derived above in comparison with known results most relevant to our interest in literature.

LAMB bound in You et al. (2020): We first start our discussion with the comparison of convergence rate of Fed-LAMB with that of LAMB, Theorem 3 in You et al. (2020). The convergence rates of Fed-LAMB and LAMB differ in two ways: (i) First, note that the characterization, on the suboptimality, or convergence criterion, is given at the averaged parameters noted $\overline{\theta}_r$ due to our distributed settings. It is thus natural to consider the evolution of our objective function, precisely its gradient, 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 (4) 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 and Lan (2013). (ii) Assuming that the convergence criterion in both Theorems is of similar order (which happens for a large enough number of rounds), the convergence rate of Fed-LAMB displays a similar $\mathcal{O}(1/R)$ behaviour for the initialization term. That said, despite the distributed (federated) setting, our dimension-wise and layer-wise method benefits from the double adaptivity phenomenon explained above and exhibited in LAMB (You et al., 2020), under a central server setting.

Fed-AMS bound in Chen et al. (2020): We now discuss the similarities and differences between Fed-AMS, the baseline distributed adaptive method developed in Chen et al. (2020), and our Fed-LAMB. For clarity, we restate their main result (Theorem 4.9) under our notations.

Theorem 4.9 (Chen et al. (2020))**.** *Under some regularity conditions on the local losses and similar assumption as ours, with some properly chosen learning rate, when $R \geq \mathcal{O}(\frac{16nL^2}{p})$, Fed-AMS has convergence rate*

$$\frac{1}{R} \sum_{r=1}^{R} \mathbb{E}\left[\left\| \nabla f(\overline{\theta_r}) \right\|^2 \right] \leq \mathcal{O}(\frac{\sqrt{p}}{\sqrt{nR}}). \tag{6}$$

Firstly, when the number of rounds R is sufficiently large, both rates (5) and (6) are dominated by $\mathcal{O}(\frac{\sqrt{p}}{\sqrt{nR}})$, matching the convergence rate of the standard AMSGrad, e.g. (Zhou et al., 2018a). The acceleration of our layer-wise scheme is exhibited in the $\mathcal{O}(1/(nR))$ term and the dependence on the number of layers $\mathcal{O}(1/(h^3 R^{3/2}))$ term.

Secondly, in (5), the last term containing the number of local updates T is small as long as $T \leq \mathcal{O}(\frac{R^{1/2}h^{5/4}}{(np)^{1/4}})$. Treating $p^{1/4}/h = \mathcal{O}(1)$, the result implies that for a given T, we can get the same rate of convergence as vanilla AMSGrad with $R \geq \mathcal{O}(T^2)$ rounds of communication. From Theorem 4.9, we know that Fed-AMS requires $R \geq \mathcal{O}(T^3)$ rounds to achieve the same rate. This implies that, our layer-wise framework reduces the number of communication rounds needed to reach an δ-stationary point compared with Chen et al. (2020). Therefore, by leveraging the layer-wise acceleration on local models, our class of methods improve the communication cost of Fed-AMS.

4.3. Extended discussions

We provide more discussion on the algorithmic and theoretical properties of Algorithm 1 from the following aspects:

Communication and Client Efficiency: The (sublinear) dependence on the number of communication rounds of our bound matches that of most recent methods in federated learning, e.g., SCAFFOLD (Karimireddy et al., 2019), a solution to the problem posed by heterogeneity of the data in each client. Yet, contrary to SCAFFOLD, our method only sends bits once per communication round while SCAFFOLD needs to send two vectors, including an additional control variate term from the clients to the central server. Our result also matches the communication bound of Reddi et al. (2021) which adapts Adam (Kingma and Ba, 2015) to the federated setting. The algorithm of Reddi et al. (2021) performs adaptive updates only at the central server, while SGD is still used for local updates. In addition, the $1/\sqrt{n}$ term in convergence rate of our method implies a linear speedup in the number of clients, which also matches the dependency on n of most federated learning methods.

Data Heterogeneity: To demonstrate the effect of non-iid data distribution theoretically, some related works pose assumptions on the global variance and the local variance of the stochastic gradients of the objective function (1) separately, such that both variances appears in the convergence rate. Our analysis can also be easily extended to incorporate the global variance term. While some works, including Karimireddy et al. (2019), target on designing specific strategies to alleviate the negative influence of data heterogeneity, we note that our LAMB FL methods are, in some sense, naturally capable of balancing the heterogeneity in different local data distributions. As mentioned before, this is largely due to the "moment averaging" step (line 20 in Algorithm 1), where the adaptive learning rates guided by the second moment estimation are aggregated among clients periodically. In our experiments, we highlight that the advantage of our class of methods is consistent under both homogeneous and heterogeneous settings.

Dependence on the dimension p: The $\sqrt{p}$ term appearing in our bound is due to the assumption on coordinate-wise bounded variance. Recent efficient techniques aim at reducing the number of bits transmitted at each round through sketches or compression techniques, see for instance Haddadpour et al. (2020); Ivkin et al. (2019); Li et al. (2019) to name a few. This is naturally compatible with our scheme, but is not the main focus of our contribution.

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 main objective is to validate the benefit of dimension-wise adaptive learning rate when integrated with the locally adaptive FL method. 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 brings notable improvement in generalization over baselines.

Methods. We evaluate the following five FL algorithms:

1. Fed-SGD (McMahan et al., 2017), standard federated averaging with local SGD updates.
2. Adp-Fed (*Adaptive Federated Optimization*, see Appendix A), the federated adaptive algorithm proposed by (Reddi et al., 2021). Adp-Fed performs local SGD updates. At 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 with adaptive learning rate averaging.
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 (Algorithm 1), layer-wise accelerated local AMSGrad.

For all the adaptive gradient methods, we set the hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ as recommended (Reddi et al., 2018). For the federated learning environment, we use $n = 50$ clients with 0.5 participation rate, i.e., we randomly pick half of the clients to be active for training in each round. We set the local mini-batch size as 128. At 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.

We tune the initial learning rate α for each algorithm over a fine grid. For Adp-Fed, there are two learning rates involved.

We tune the combination of local learning rate (for local SGD) and global learning rate (for global Adam) over a fine grid. The detailed learning rate tuning diagram can be found in Appendix A. For Fed-LAMB and Mime-LAMB, the weight decay rate λ is tuned from $\{0, 0.01, 0.1\}$, and we use the identity scaling function for $\phi(\cdot)$. For each run, we report the test accuracy with the best α and λ. The results are averaged over three independent runs, with same initialization for every method.

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, 2009) and TinyImageNet (Deng et al., 2009). The MNIST dataset contains 60000 training samples and 10000 test samples, from 10 classes of handwritten digits from 0 to 9. The FMNIST dataset has the same sample size and train/test split as MNIST, but the samples are fashion products (e.g., dress, bags) which makes it harder to train than MNIST. The CIFAR-10 dataset has 50000 training images and 10000 test images, from 10 classes. The TinyImageNet is a subset of the ImageNet dataset. It includes 100000 64×64 images for from 200 classes for training and 10000 for testing. For MNIST, we apply 1) a simple multi-layer perceptron (MLP), which has one hidden layer containing 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 i.i.d. settings

In Figure 2, we report the test accuracy of MLP trained on MNIST, as well as CNN trained on MNIST and FMNIST, where the data is i.i.d. allocated among the clients. We test 1 local epoch and 3 local epochs (more local iterations). 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 communication efficiency. For instance, this can be observed as follows: on MNIST + CNN (1 local epoch), Fed-AMS requires 20 rounds to achieves 90% accuracy, while Fed-LAMB only takes 5 rounds. This implies a 75% reduction in the communication cost. Moreover, on MNIST, Fed-LAMB also leads to improved generalisation performance, i.e., test accuracy. Note that, the result on MLP to a good extent provides a straightforward illustration on the benefit of layer-wise adaptivity in Fed-LAMB, since compared with Fed-AMS, the only difference is that the learning rate becomes adaptive to the scale of the single hidden layer in Fed-LAMB. We can draw same conclusions with 3 local epochs. Also,

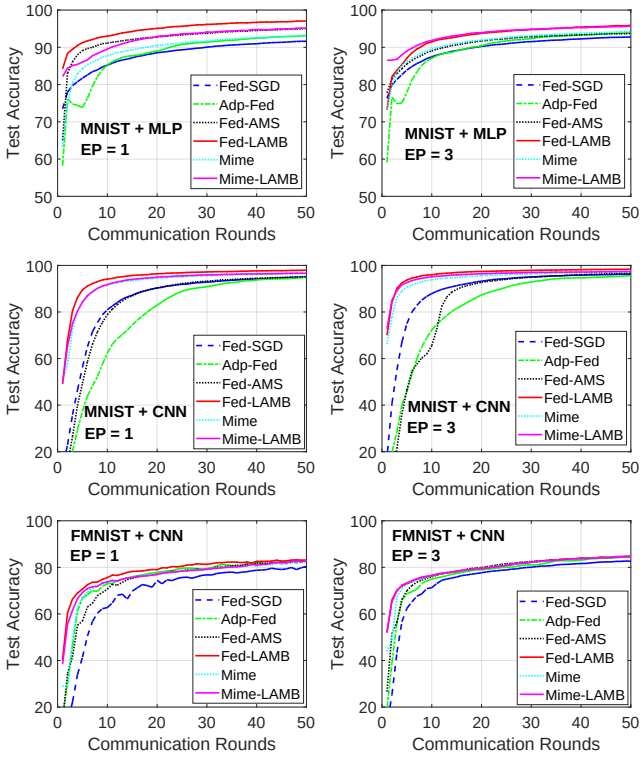

Figure 2: **i.i.d. data setting**. Test accuracy on MNIST and FMNIST against the number of communication rounds. **Left Column:** 1 local epoch. **Right Column:** 3 local epochs. **1st row:** MNIST + MLP. **2nd row:** MNIST + CNN. **3rd row:** FMNIST + CNN.

similar comparison holds for Mime-LAMB vs. Mime. In general, the Mime-LAMB variant matches the performance of Fed-LAMB closely.

5.2. Comparison under non-i.i.d. settings

In Figure 3, we provide the results on MNIST and FMNIST, when the local data has non-i.i.d. distribution (i.e., under strong data heterogeneity). In particular, in each round of federated training, every local device only receives samples from one or two class (out of ten). This is known to be the scenario where federated learning is harder to generalize well (McMahan et al., 2017), thus an important case for the empirical evaluation of FL methods. First of all, from Figure 3, we see that for experiments with 1 local epoch, in all cases our proposed Fed-LAMB outperforms all the baseline methods. Similar to the i.i.d. 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. On both two datasets, Fed-LAMB saves around 50% communication rounds to reach a same accuracy level as Fed-AMS. The other baseline method, Adp-Fed, performs as good as our Fed-LAMB on FMNIST, but worse than other methods on MNIST.

Table 1: Test Accuracy with ResNet-18 Network.

	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	$\mathbf{92.44 \pm 0.53}$	90.94 ± 0.13	$\mathbf{92.00 \pm 0.21}$
TinyImageNet	67.58 ± 0.21	74.17 ± 0.43	64.86 ± 0.83	$\mathbf{76.00 \pm 0.26}$	67.82 ± 0.24	$\mathbf{73.46 \pm 0.25}$

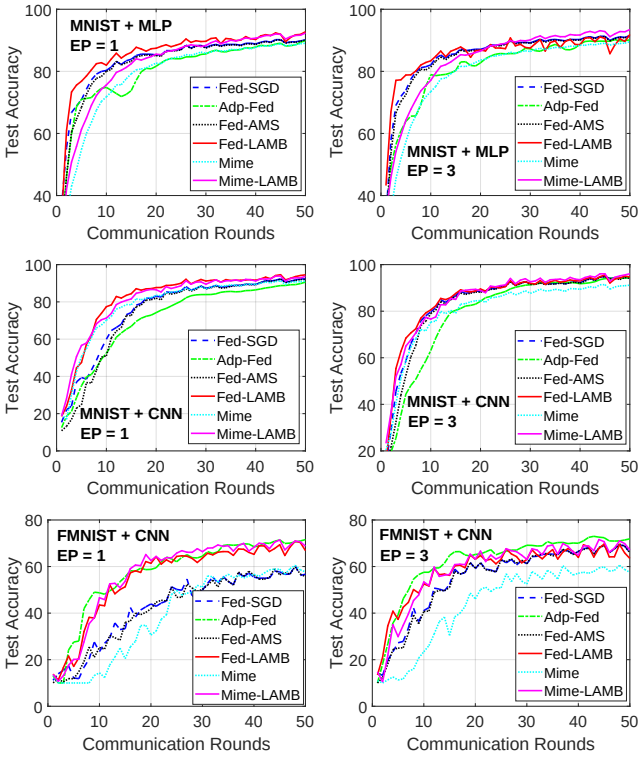

Figure 3: **non-i.i.d. data setting.** Test accuracy on MNIST and FMNIST against the number of communication rounds. **Left:** 1 local epoch. **Right:** 3 local epochs. **1st row:** MNIST + MLP. **2nd row:** MNIST + CNN. **3rd row:** FMNIST + CNN.

The relative comparison is basically the same when we conduct 3 local epochs. Yet, the advantage of Fed-LAMB becomes less significant than what we observed in Figure 2 with iid local data distribution. One plausible reason is that when the local data is highly non-i.i.d., the fast convergence of the local models as in Fed-LAMB might not be as beneficial when we allow too many local iterations. Intuitively, learning the local models "too fast" might not always be a good thing to the global model, since local models target at different loss functions. Finally, Mime-LAMB also considerably improves Mime, in all the runs, see Figure 3.

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 three baselines. Although Adp-Fed performs better

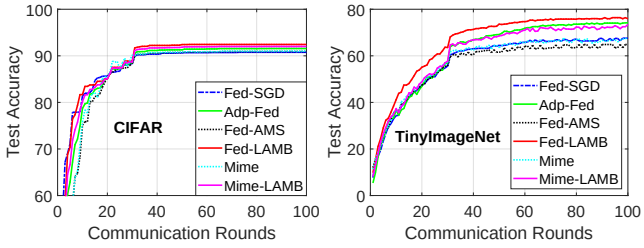

Figure 4: **non-i.i.d. data setting.** Test accuracy on CIFAR-10 + ResNet-18 and TinyImagenet + ResNet-18.

than Fed-SGD and Fed-AMS, it is 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. Again, Mime-LAMB greatly improves Mime, but performs slightly worse than Fed-LAMB.

6. Conclusion

We study a doubly adaptive method in the particular framework of federated learning. Built upon the success of periodic averaging, and of state-of-the-art adaptive gradient methods for single server nonconvex stochastic optimization, we derive a layer-wise FL framework, based on a distributed AMSGrad method, that performs local updates on each worker and periodically averages local models stored on each device. When the trained model is a deep neural network, a core component of our methods, namely Fed-LAMB and Mime-LAMB, is a *layer-wise* update of each local model. The main contribution of our paper is thus a federated learning optimization algorithm that leverages a double level of adaptivity: the first one stems from a *dimension-wise* adaptivity, inspired by adaptive gradient methods, extended to their distributed (and local) counterpart, the second one is due to a *layer-wise* adaptivity making use of the particular compositionality of the considered model. Proved convergence guarantees of our scheme are provided in our contribution, and exhibit a sublinear dependence on the total number of communications rounds, and a linear speedup against the number of clients. 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 could lead to reduced communication cost. In many cases, our framework also improves the overall performance of federated learning over prior methods.

References

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of A class of adam-type algorithms for non-convex optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 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 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 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.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23(4):2341–2368, 2013.

Farzin Haddadpour, Belhal Karimi, Ping Li, and Xiaoyun Li. Fedsketch: Communication-efficient and private federated learning via sketching. *arXiv preprint arXiv:2008.04975*, 2020.

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.

Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. Communication-efficient distributed SGD with sketching. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13144–13154, Vancouver, Canada, 2019.

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 *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 4519–4529. PMLR, 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. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, 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 *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 6341–6345. IEEE, 2019.

Tian Li, Zaoxing Liu, Vyas Sekar, and Virginia Smith. Privacy for free: Communication-efficient learning with differential privacy using sketches. *arXiv preprint arXiv:1911.00972*, 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.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b.

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.

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. *Springer*, 2004.

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.

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

Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *CoRR*, abs/1812.06127, 2018.

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

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael G. Rabbat. Slowmo: Improving communication-efficient distributed SGD with slow momentum. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

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 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10334–10343. PMLR, 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 2018, Eugene, OR, USA, August 13-16, 2018*, pages 1:1–1:10. ACM, 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 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7184–7193. PMLR, 2019.

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

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*, 2018a.

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*, 2018b.

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. Hyper-parameter Tuning and Algorithms

A.1. The Adp-Fed Algorithm (Reddi et al., 2021)

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.

Algorithm 2 Adp-Fed: Adaptive Federated Optimization (Reddi et al., 2021)

1: **Input**: parameter $0 < \beta_1, \beta_2 < 1$, and learning rate α_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 θ_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	$[0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5]$

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 4.5.

B.1. Intermediary Lemmas

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

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

where ϕ_M is defined in Assumption 4.4 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 $r > 0$:

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

leading to

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

which concludes the proof. □

Lemma. Consider $\{\overline{\theta_r}\}_{r>0}$, the sequence of parameters obtained running Algorithm 1. Then for $r > 0$:

$$\left\| \frac{\overline{\nabla} f(\theta_r)}{\sqrt{v_r}} \right\|^2 \geq \frac{1}{2} \left\| \frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r}} \right\|^2 - \overline{L}\alpha^2 M^2 \phi_M^2 \frac{(1-\beta_2)p}{\epsilon} ,$$

where M is defined in Assumption 4.2, $p = \sum_{\ell=1}^{\mathsf{h}} p_\ell$ and ϕ_M is defined in Assumption 4.4.

Proof. Consider the following sequence:

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

where the inequality is due to the Cauchy-Schwartz inequality.

Under the smoothness assumption Assumption 4.1 and using Lemma 4.6, we have

$$\left\| \frac{\overline{\nabla} f(\theta_r)}{\sqrt{v_r}} \right\|^2 \geq \frac{1}{2} \left\| \frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r}} \right\|^2 - \left\| \frac{\overline{\nabla} f(\theta_r) - \nabla f(\overline{\theta_r})}{\sqrt{v_r}} \right\|^2 \geq \frac{1}{2} \left\| \frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r}} \right\|^2 - \overline{L}\alpha^2 M^2 \phi_M^2 \frac{(1-\beta_2)p}{\epsilon} ,$$

which concludes the proof. □

B.2. Proof of Theorem 4.5

We now develop a proof for the two intermediary lemmas, Lemma 4.6 and Lemma 4.7, in the case when each local model is obtained after more than one local update. Then the two quantities, either the gap between the periodically averaged parameter and each local update, i.e., $\|\overline{\theta_r} - \theta_{r,i}\|^2$, and the ratio of the average gradient, more particularly its relation to the gradient of the average global model (i.e., $\left\| \frac{\overline{\nabla f(\theta_r)}}{\sqrt{v_r^t}} \right\|$ and $\left\| \frac{\nabla f(\overline{\theta_r})}{\sqrt{v_r^t}} \right\|$), are impacted.

Theorem. Assume **Assumption 4.1-Assumption 4.4**. 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

$$\frac{1}{R}\sum_{r=1}^{R}\mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2\right] \leq \sqrt{\frac{M^2 p}{n}}\frac{\triangle}{\mathsf{h}\alpha R} + \frac{4\alpha\alpha^2 L M^2 (T-1)^2 \phi_M^2 (1-\beta_2)p}{\sqrt{\epsilon}} + 4\alpha\frac{M^2}{\sqrt{\epsilon}} + \frac{\phi_M \sigma^2}{Rn}\sqrt{\frac{1-\beta_2}{M^2 p}} + 4\alpha\left[\phi_M \frac{\mathsf{h}\sigma^2}{\sqrt{n}}\right] + 4\alpha\left[\phi_M^2\sqrt{M^2 + p\sigma^2}\right] + cst, \tag{7}$$

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

Proof. Using Assumption 4.1, we have

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

Taking expectations on both sides leads to

$$-\mathbb{E}[\left\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \right\rangle] \leq \mathbb{E}[f(\bar{\vartheta}_r) - f(\bar{\vartheta}_{r+1})] + \sum_{\ell=1}^{L}\frac{L_\ell}{2}\mathbb{E}[\|\bar{\vartheta}_{r+1}^\ell - \bar{\vartheta}_r^\ell\|^2] \,. \tag{8}$$

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

$$\bar{\vartheta}_r = \bar{\theta}_r + \frac{\beta_1}{1-\beta_1}(\bar{\theta}_r - \bar{\theta}_{r-1}) \,, \tag{9}$$

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

$$\begin{aligned} \bar{\vartheta}_{r+1}^\ell - \bar{\vartheta}_r^\ell &= \frac{1}{1-\beta_1}(\bar{\theta}_{r+1}^\ell - \bar{\theta}_r^\ell) - \frac{\beta_1}{1-\beta_1}(\bar{\theta}_r^\ell - \bar{\theta}_{r-1}^\ell) \\ &= \frac{\alpha_r}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^\ell\|)}{\|\psi_{r,i}^\ell\|}\psi_{r,i}^\ell - \frac{\alpha_{r-1}}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta_{r-1,i}^\ell\|)}{\|\psi_{r-1,i}^\ell\|}\psi_{r-1,i}^\ell \\ &= \frac{\alpha\beta_1}{1-\beta_1}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\phi(\|\theta_{r,i}^\ell\|)}{\sqrt{v_r^t}\|\psi_{r,i}^\ell\|} - \frac{\phi(\|\theta_{r-1,i}^\ell\|)}{\sqrt{v_{r-1}^t}\|\psi_{r-1,i}^\ell\|}\right)m_{r-1}^t + \frac{\alpha}{n}\sum_{i=1}^{n}\frac{\phi(\|\theta_{r,i}^\ell\|)}{\sqrt{v_r^t}\|\psi_{r,i}^\ell\|}g_{r,i}^t \,, \end{aligned} \tag{10}$$

where we have assumed a constant learning rate α.

We note for all $\theta \in \Theta$, the majorant $G > 0$ such that $\phi(\|\theta\|) \leq G$. Then, following (8), we obtain

$$-\mathbb{E}[\left\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \right\rangle] \leq \mathbb{E}[f(\bar{\vartheta}_r) - f(\bar{\vartheta}_{r+1})] + \sum_{\ell=1}^{L}\frac{L_\ell}{2}\mathbb{E}[\|\bar{\vartheta}_{r+1} - \bar{\vartheta}_r\|^2] \,. \tag{11}$$

Developing the LHS of (11) using (10) leads to

$$\begin{aligned}\langle \nabla f(\bar{\vartheta}_r) \,|\, \bar{\vartheta}_{r+1} - \bar{\vartheta}_r \rangle &= \sum_{\ell=1}^{\mathsf{h}} \sum_{j=1}^{p_\ell} \nabla_\ell f(\bar{\vartheta}_r)^j (\bar{\vartheta}_{r+1}^{\ell,j} - \bar{\vartheta}_r^{\ell,j}) \\ &= \frac{\alpha\beta_1}{1-\beta_1} \frac{1}{n} \sum_{\ell=1}^{\mathsf{h}} \sum_{j=1}^{p_\ell} \nabla_\ell f(\bar{\vartheta}_r)^j \left[\sum_{i=1}^{n} \left(\frac{\phi(\|\theta_{r,i}^\ell\|)}{\sqrt{v_r^t}\|\psi_{r,i}^\ell\|} - \frac{\phi(\|\theta_{r-1,i}^\ell\|)}{\sqrt{v_{r-1}^t}\|\psi_{r-1,i}^\ell\|} \right) m_{r-1}^t \right] \\ &\underbrace{- \frac{\alpha}{n} \sum_{\ell=1}^{\mathsf{h}} \sum_{j=1}^{p_\ell} \nabla_\ell f(\bar{\vartheta}_r)^j \sum_{i=1}^{n} \frac{\phi(\|\theta_{r,i}^\ell\|)}{\sqrt{v_r^t}\|\psi_{r,i}^\ell\|} g_{r,i}^{t,l,j}}_{=A_1} . \end{aligned} \tag{12}$$

Suppose T is the total number of local iterations and R is the number of rounds. We can write (12) 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

$$\langle \nabla f(\bar{\vartheta}_r), \frac{\bar{g}_r}{\sqrt{\hat{v}_r}} \rangle = \frac{1}{2} \| \frac{\nabla f(\bar{\vartheta}_r)}{\hat{v}_r^{1/4}} \|^2 + \frac{1}{2} \| \frac{\bar{g}_r}{\hat{v}_r^{1/4}} \|^2 - \| \frac{\nabla f(\bar{\vartheta}_r) - \bar{g}_r}{\hat{v}_r^{1/4}} \|^2. \tag{13}$$

To bound the last term on the RHS, we have

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

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

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

For the second term,

$$\frac{2}{n} \sum_{i=1}^n \| \frac{\nabla f(\bar{\theta}_r) - \bar{g}_{t,i}}{\hat{v}_r^{1/4}} \|^2 \le \frac{4}{n} \Big(\underbrace{\sum_{i=1}^n \| \frac{\nabla f(\bar{\theta}_r) - \nabla f(\theta_{t,i})}{\hat{v}_r^{1/4}} \|^2}_{B_1} + \underbrace{\sum_{i=1}^n \| \frac{\nabla f(\theta_{t,i}) - \bar{g}_{t,i}}{\hat{v}_r^{1/4}} \|^2}_{B_2} \Big). \tag{14}$$

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

$$\begin{aligned} B_1 \le \frac{L}{\sqrt{\epsilon}} \sum_{i=1}^n \|\bar{\theta}_r - \theta_{t,i}\|^2 = \frac{\alpha^2 L}{\sqrt{\epsilon}} \sum_{i=1}^n \sum_{l=1}^L \| \sum_{j=\lfloor t \rfloor_r + 1}^t \Big(\frac{\phi(\|\theta_{j,i}^l\|)}{\|\psi_{j,i}^l\|} \psi_{j,i}^l - \frac{1}{n} \sum_{k=1}^n \frac{\phi(\|\theta_{j,k}^l\|)}{\|\psi_{j,k}^l\|} \psi_{j,k}^l \Big) \|^2 \\ \le n \frac{\alpha^2 L}{\sqrt{\epsilon}} M^2 (T-1)^2 \phi_M^2 (1-\beta_2) p, \end{aligned} \tag{15}$$

where the last inequality stems from Lemma 4.6 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 develop the expectation of B_2 under the simplification that $\beta_1=0$:

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

where we use assumption Assumption 4.2, Assumption 4.3 and Assumption 4.4. Yet,

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

Then we have

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

Thus, (14) becomes

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

Substituting all ingredients into (13), we obtain

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

At the same time, we have

$$\begin{aligned}
\mathbb{E}\big[\|\frac{\bar{g}_r}{\hat{v}_r^{1/4}}\|^2\big]=\frac{1}{n^2}\mathbb{E}\big[\|\frac{\sum_{i=1}^{n}\bar{g}_{t,i}}{\hat{v}_r^{1/4}}\|^2\big] &= \frac{1}{n^2}\mathbb{E}\big[\sum_{l=1}^{L}\sum_{i=1}^{n}\|\frac{\phi(\|\theta_{t,i}^l\|)}{\hat{v}^{1/4}\|\psi_{t,i}^l\|}g_{t,i}^l\|^2\big] \\
&\geq \phi_m^2(1-\beta_2)\mathbb{E}\left[\|\frac{1}{n}\sum_{i=1}^{n}\frac{\nabla f(\theta_{t,i})}{\hat{v}_r^{1/4}}\|^2\right] \\
&= \phi_m^2(1-\beta_2)\mathbb{E}\left[\|\frac{\overline{\nabla}f(\theta_t)}{\hat{v}_r^{1/4}}\|^2\right].
\end{aligned}\tag{16}$$

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

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

where the last line is due to (15). Therefore, we have obtained

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

Substitute back into (12), and leave other derivations unchanged. Assuming $M \leq 1$, we have the following by taking the telescope sum

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

And if we set the learning rate to be of order $\mathcal{O}(\frac{1}{\sqrt{\mathsf{h}R}})$ then:

$$\frac{1}{R}\sum_{t=1}^R \mathbb{E}\left[\left\|\frac{\nabla f(\overline{\theta_r})}{\hat{v}_r^{1/4}}\right\|^2\right] \leq \mathcal{O}\left(\sqrt{\frac{M^2 p}{n}}\frac{1}{\sqrt{\mathsf{h}R}} + \frac{G^2(T-1)^2 p}{R\sqrt{L}} + \frac{\sigma^2}{Rn\sqrt{p}}\right).$$

This concludes the proof.

□