

---

# Convergent Adaptive Gradient Methods in Decentralized Optimization

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

Adaptive gradient methods including Adam, AdaGrad, and their variants have been very successful for training deep learning models, such as neural networks, in the past few years. Meanwhile, given the need for distributed training procedures, distributed optimization algorithms are at the center of attention. With the growth of computing power and the need for using machine learning models on mobile devices, the communication cost of distributed training algorithms needs careful consideration. In that regard, more and more attention is shifted from the traditional parameter server training paradigm to the decentralized one, which usually requires lower communication costs. In this paper, we rigorously incorporate adaptive gradient methods into decentralized training procedures and introduce novel convergent decentralized adaptive gradient methods. Specifically, we propose a general algorithmic framework that can convert existing adaptive gradient methods to their decentralized counterparts. In addition, we thoroughly analyze the convergence behavior of the proposed algorithmic framework and show that if a given adaptive gradient method converges, under some specific conditions, then its decentralized counterpart is also convergent.

## 1 Introduction

Distributed training of machine learning models is drawing growing attention in the past few years due to its practical benefits and necessities. Given the evolution of computing capabilities of CPUs and GPUs, computation time in distributed settings is gradually dominated by the communication time in many circumstances [Chilimbi et al., 2014, McMahan et al., 2016]. As a result, a large amount of recent works has been focussing on reducing communication cost for distributed learning [Alistarh et al., 2017, Lin et al., 2017, Wangni et al., 2018, Stich et al., 2018, Wang et al., 2018, Tang et al., 2019]. In the traditional parameter (central) server setting, where a parameter server is employed to manage communication in the whole network, many effective communication reductions have been proposed based on gradient compression [Aji and Heafield, 2017] and quantization [Chen et al., 2010, Ge et al., 2013, Jegou et al., 2010] techniques. Despite these communication reduction techniques, its cost still, usually, scales linearly with the number of workers. Due to this limitation and with the sheer size of decentralized devices, the *decentralized training paradigm* [Duchi et al., 2011b], where the parameter server is removed and each node only communicates with its neighbors, is drawing attention. It has been shown in Lian et al. [2017] that decentralized training algorithms can outperform parameter server-based algorithms when the training bottleneck is the communication cost. The decentralized paradigm is also preferred when a central parameter server is not available.

In light of recent advances in nonconvex optimization, an effective way to accelerate training is by using adaptive gradient methods like AdaGrad [Duchi et al., 2011a], Adam [Kingma and Ba, 2014] or AMSGrad [Reddi et al., 2019]. Their popularity are due to their practical benefits in training neural networks, featured by faster convergence and ease of parameter tuning compared with Stochastic Gradient Descent (SGD) [Robbins and Monro, 1951]. Despite a large amount of studies

within the distributed optimization literature, few works have considered bringing adaptive gradient methods into distributed training, largely due to the lack of understanding of their convergence behaviors. Notably, Reddi et al. [2020] develop the first decentralized ADAM method for distributed optimization problems with a direct application to federated learning. An inner loop is employed to compute mini-batch gradients on each node and a global adaptive step is performed to update the global parameter at each outer iteration. Yet, in the settings of our paper, nodes can only communicate to their neighbors on a fixed communication graph while a server/worker communication is needed in [Reddi et al., 2020]. Designing adaptive methods in such settings is highly non-trivial due to the already complicated update rules and to the interaction between the effect of using adaptive learning rates and the decentralized communication protocols. This paper is an attempt at bridging the gap between both realms in nonconvex optimization. Our contributions are summarized as follows:

- In this paper, we investigate the possibility of using any adaptive gradient methods in the decentralized training paradigm, where nodes have only a local view of the whole communication graph. We develop a general technique that converts an adaptive gradient method from a centralized method to its decentralized variant.
- By using our proposed technique, we present a new decentralized optimization algorithm, called decentralized AMSGrad, as the decentralized counterpart of AMSGrad.
- We provide a theoretical verification interface, in Theorem 2, for analyzing the behavior of decentralized adaptive gradient methods obtained as a result of our technique. Thus, we characterize the convergence rate of decentralized AMSGrad, which is the first convergent decentralized adaptive gradient method, to the best of our knowledge.

A *novel technique* in our framework is a mechanism to enforce a consensus on adaptive learning rates at different nodes. We show the importance of consensus on adaptive learning rates by proving a divergent problem instance for a recently proposed decentralized adaptive gradient method, namely DADAM [Nazari et al., 2019], a decentralized version of ADAM. Though consensus is performed on the model parameter, DADAM lacks of consensus principles on the nodes adaptive learning rates.

After having presented existing related work and important concepts of decentralized adaptive methods in Section 2, we develop our general framework for converting any adaptive gradient algorithm in its decentralized counterpart along with their rigorous finite-time convergence analysis in Section 3 concluded by some illustrative examples of our framework’s behavior in practice.

**Notations:**  $x_{t,i}$  denotes variable  $x$  at node  $i$  and iteration  $t$ .  $\|\cdot\|_{abs}$  denotes the entry-wise  $L_1$  norm of a matrix, i.e.  $\|A\|_{abs} = \sum_{i,j} A_{i,j}$ . We introduce important notations used throughout the paper: for any  $t > 0$ ,  $G_t := [g_{t,N}]$  where  $[g_{t,N}]$  denotes the vector  $[g_{t,1}, g_{t,2}, \dots, g_{t,N}]$ ,  $M_t := [m_{t,N}]$ ,  $X_t := [x_{t,N}]$ ,  $\nabla f(X_t) := \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i})$ ,  $U_t := [u_{t,N}]$ ,  $\tilde{U}_t := [\tilde{u}_{t,N}]$ ,  $V_t := [v_{t,N}]$ ,  $\hat{V}_t := [\hat{v}_{t,N}]$ ,  $\bar{X}_t := \frac{1}{N} \sum_{i=1}^N x_{t,i}$ ,  $\bar{U}_t := \frac{1}{N} \sum_{i=1}^N u_{t,i}$  and  $\bar{\tilde{U}}_t := \frac{1}{N} \sum_{i=1}^N \tilde{u}_{t,i}$ .

## 2 Decentralized Adaptive Training and Divergence of DADAM

### 2.1 Related Work

**Decentralized optimization:** Traditional decentralized optimization methods include well-know algorithms such as ADMM [Boyd et al., 2011], Dual Averaging [Duchi et al., 2011b], Distributed Subgradient Descent [Nedic and Ozdaglar, 2009]. More recent algorithms include Extra [Shi et al., 2015], Next [Di Lorenzo and Scutari, 2016] and Prox-PDA [Hong et al., 2017]. While these algorithms are commonly used in applications other than deep learning, recent algorithmic advances in the machine learning community have shown that decentralized optimization can be useful for training deep models such as neural networks as well. Lian et al. [2017] demonstrate that a stochastic version of Decentralized Subgradient Descent can outperform parameter server-based algorithms when the communication cost is high. Tang et al. [2018] propose the  $D^2$  algorithm improving the convergence rate over Stochastic Subgradient Descent. Assran et al. [2018] propose the Stochastic Gradient Push that is more robust to network failures for training neural networks. The study of decentralized training algorithms in the machine learning community is only at its initial stage. No existing work, to our knowledge, has seriously considered integrating *adaptive gradient methods* in the setting of decentralized learning. One noteworthy work [Nazari et al., 2019] propose a decentralized version of AMSGrad [Reddi et al., 2019] and is proven to satisfy some non-standard regret.

**Adaptive gradient methods:** Adaptive gradient methods have been popular in recent years due to their superior performance in training neural networks. Most commonly used adaptive methods include AdaGrad [Duchi et al., 2011a] or Adam [Kingma and Ba, 2014] and their variants. Key features of such methods lie in the use of momentum and adaptive learning rates (which means that the learning rate is changing during the optimization and is anisotropic, i.e. depends on the dimension). The method of reference, called Adam, has been analyzed in [Reddi et al., 2019] where the authors point out an error in previous convergence analyses. Since then, a variety of papers have been focussing on analyzing the convergence behavior of the numerous existing adaptive gradient methods. Ward et al. [2018], Li and Orabona [2018] derive convergence guarantees for a variant of AdaGrad without coordinate-wise learning rates. Chen et al. [2018] analyze the convergence behavior of a broad class of algorithms including AMSGrad and AdaGrad. Zou and Shen [2018] provide a unified convergence analysis for AdaGrad with momentum. Noticeable recent works on adaptive gradient methods can be found in [Agarwal et al., 2018, Luo et al., 2019, Zaheer et al., 2018].

## 2.2 Decentralized Optimization

In distributed optimization (with  $N$  nodes), we aim at solving the following problem

$$\min_{x \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^N f_i(x), \quad (1)$$

where  $x$  is the vector of parameters and  $f_i$  is only accessible by the  $i$ th node. Through the prism of empirical risk minimization procedures,  $f_i$  can be viewed as the average loss of the data samples located at node  $i$ , for all  $i \in [N]$ . Throughout the paper, we make the following mild assumptions required for analyzing the convergence behavior of the different decentralized optimization algorithms:

**A1.** For all  $i \in [N]$ ,  $f_i$  is differentiable and the gradients is  $L$ -Lipschitz, i.e., for all  $(x, y) \in \mathbb{R}^d$ ,  $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$ .

**A2.** We assume that, at iteration  $t$ , node  $i$  accesses a stochastic gradient  $g_{t,i}$ . The stochastic gradients and the gradients of  $f_i$  have bounded  $L_\infty$  norms, i.e.  $\|g_{t,i}\| \leq G_\infty$ ,  $\|\nabla f_i(x)\|_\infty \leq G_\infty$ .

**A3.** The gradient estimators are unbiased and each coordinate have bounded variance, i.e.  $\mathbb{E}[g_{t,i}] = \nabla f_i(x_{t,i})$  and  $\mathbb{E}[(g_{t,i} - \nabla f_i(x_{t,i}))_j^2] \leq \sigma^2, \forall t, i, j$ .

Assumptions A1 and A3 are standard in distributed optimization literature. A2 is slightly stronger than the traditional assumption that the estimator has bounded variance, but is commonly used for the analysis of adaptive gradient methods [Chen et al., 2018, Ward et al., 2018]. Note that the bounded gradient estimator assumption in A2 implies the bounded variance assumption in A3. In decentralized optimization, the nodes are connected as a graph and each node only communicates to its neighbors. In such case, one usually constructs a  $N \times N$  matrix  $W$  for information sharing when designing new algorithms. We denote  $\lambda_i$  to be its  $i$ th largest eigenvalue and define  $\lambda \triangleq \max(|\lambda_2|, |\lambda_N|)$ . The matrix  $W$  cannot be arbitrary, its required key properties are listed in the following assumption:

**A4.** The matrix  $W$  satisfies: (I)  $\sum_{j=1}^N W_{i,j} = 1$ ,  $\sum_{i=1}^N W_{i,j} = 1$ ,  $W_{i,j} \geq 0$ , (II)  $\lambda_1 = 1$ ,  $|\lambda_2| < 1$ ,  $|\lambda_N| < 1$  and (III)  $W_{i,j} = 0$  if node  $i$  and node  $j$  are not neighbors.

We now present the failure to converge of current decentralized adaptive method before introducing our proposed framework for general decentralized adaptive gradient methods.

## 2.3 Divergence of DADAM

Recently, Nazari et al. [2019] initiated an attempt to bring adaptive gradient methods into decentralized optimization, the resulting algorithm is DADAM, which is shown in Algorithm 1. DADAM is essentially a decentralized version of ADAM and the key modification is the use of a consensus step on optimization variable  $x$  to transmit information across the network, encouraging convergence. The matrix  $W$  is a doubly stochastic matrix (which satisfies A4) for achieving average consensus of  $x$ . Introducing such mixing matrix is a standard approach for decentralizing an algorithm, such as distributed gradient descent [Nedic and Ozdaglar, 2009, Yuan et al., 2016]. It is proven in Nazari et al. [2019] that DADAM admits a non-standard regret bound in the online setting, however, whether the algorithm can converge to stationary points in standard offline settings such training neural networks is still unknown.

138 In the following, we show the DADAM may fail to  
139 converge in the offline optimization settings.

140 **Theorem 1.** *There exist a problem satisfying*  
141 *A1-A4 where DADAM fail to converge.*

142 *Proof.* Consider a 1 dimensional optimiza-  
143 tion problem distributed on two nodes  
144  $\min_x \frac{1}{2} \sum_{i=1}^2 f_i(x)$  where  $f_i(x) = \frac{1}{2}(x - a_i)^2$   
145 and  $a_1 = 0, a_2 = 1$ . The network contains  
146 only two nodes and the matrix  $W$  satisfies  
147  $W_{ij} = \frac{1}{2}$  for all indices  $i, j$ . For simplicity,  
148 we consider running DADAM algorithm with  
149  $\beta_1 = \beta_2 = \beta_3 = 0$  and  $\epsilon = 0.6$ . Suppose we  
150 initialize DADAM at  $x_{1,i} = 0$  for all  $i \in [N]$   
151 and use the following learning rate  $\alpha = 0.001$ .

152 We observe that at  $x_{1,i} = 0, \nabla f_1(x_{1,1}) = 0$ , we have  $\nabla f_2(x_{1,2}) = 1$ , leading to  $\hat{v}_{1,1} = 0.6$  and  
153  $\hat{v}_{1,2} = 1$ . Thus, from step 1, we will obtain  $\hat{v}_{1,2} \geq 1$ . In addition, it is can be easily proved that, with  
154 the above selected stepsize, we always have  $\hat{v}_{1,1} < 1$ , in fact, it will never reach the value of 0.6.  
155 Thus, in the next iterations, the gradient of losses on node 1 and 2 will be scaled differently. This  
156 scaling is equivalent to running gradient descent on a objective where the losses of the two nodes  
157 are scaled by different factors. In such case, the algorithm will converge to a stationary point of a  
158 weighted average of the loss on node 1. Recall that the problem we tackle to illustrate Theorem 1 is a  
159 quadratic problem with only one minimizer. Then, since the weight of the losses on the two nodes  
160 are different and that the unbalanced weights on the two functions yields a different minimizer, the  
161 algorithm will not converge to the unique stationary point of the original loss (which is  $x = 0.5$ ).  $\square$

162 Theorem 1 claims that even though DADAM is proven to satisfy some regret bounds, see [Nazari  
163 et al., 2019], it can fail to converge to stationary points in the nonconvex offline setting, which is  
164 a common setting for training neural networks. We conjecture that this inconsistency is due to the  
165 definition of the regret in [Nazari et al., 2019]. In the next section, we design decentralized adaptive  
166 gradient methods that are guaranteed to converge to stationary points of some defined objective and  
167 provide a characterization of that convergence in finite-time and independently of the initialization.

### 168 3 Decentralized Adaptive Gradient Methods and their Convergence

169 In this section, we discuss the difficulties of designing adaptive gradient methods in decentralized  
170 optimization and introduce an algorithmic framework that converts existing convergent adaptive gra-  
171 dient methods to their decentralized counterparts. We also develop the first convergent decentralized  
172 adaptive gradient method, converted from AMSGrad, as an instance of this proposed framework.

#### 173 3.1 Importance and Difficulties of Consensus on Adaptive Learning Rates

174 The divergent example in the previous section implies that we should synchronize the adaptive  
175 learning rates on different nodes. This can be easily achieved in the parameter server setting where  
176 all the nodes are sending their gradients to a central server at each iteration. The parameter server can  
177 then exploit the received gradients to maintain a sequence of synchronized adaptive learning rates  
178 when updating the parameters, see [Reddi et al., 2020]. However, in our decentralized setting, every  
179 node can only communicate with its neighbors and such central server does not exist. Under that  
180 setting, the information for updating the adaptive learning rates can only be shared locally instead  
181 of broadcasted over the whole network. This makes it impossible to obtain, in a single iteration, a  
182 synchronized adaptive learning rate update using all the information in the network.

183 *Systemic Approach:* On a systemic level, one way to alleviate this bottleneck is to design communi-  
184 cation protocols in order to give each node access to the same aggregated gradients over the whole  
185 network, at least periodically if not at every iteration. Therefore, the nodes can update their individual  
186 adaptive learning rates based on the same shared information. However, such solution may introduce  
187 an extra communication cost since it involves broadcasting the information over the whole network.

188 *Algorithmic Approach:* Our contributions being on an algorithmic level, another way to solve the  
189 aforementioned problem is by letting the sequences of adaptive learning rates, present on different

---

#### Algorithm 1 DADAM (with N nodes)

---

```

1: Input:  $\alpha$ , current point  $X_t, u_{\frac{1}{2},i} = \hat{v}_{0,i} = \epsilon \mathbf{1}$ ,  
    $m_0 = 0$  and mixing matrix  $W$ 
2: for  $t = 1, 2, \dots, T$  do
3:   for all  $i \in [N]$  do in parallel
4:      $g_{t,i} \leftarrow \nabla f_i(x_{t,i}) + \xi_{t,i}$ 
5:      $m_{t,i} = \beta_1 m_{t-1,i} + (1 - \beta_1) g_{t,i}$ 
6:      $v_{t,i} = \beta_2 v_{t-1,i} + (1 - \beta_2) g_{t,i}^2$ 
7:      $\hat{v}_{t,i} = \beta_3 \hat{v}_{t-1,i} + (1 - \beta_3) \max(\hat{v}_{t-1,i}, v_{t,i})$ 
8:      $x_{t+\frac{1}{2},i} = \sum_{j=1}^N W_{ij} x_{t,j}$ 
9:      $x_{t+1,i} = x_{t+\frac{1}{2},i} - \alpha \frac{m_{t,i}}{\sqrt{\hat{v}_{t,i}}}$ 
10:  end for

```

---

nodes, to gradually *consent*, through the iterations. Intuitively, if the adaptive learning rates can consent fast enough, the difference among the adaptive learning rates on different nodes will not affect the convergence behavior of the algorithm. Consequently, no extra communication costs need to be introduced. We now develop this exact idea within the existing adaptive methods stressing on the need for a relatively low-cost and easy-to-implement consensus of adaptive learning rates.

### 3.2 Decentralized Adaptive Gradient Unifying Framework

As mentioned before, we need to choose a method to implement consensus of adaptive learning rates. While each node can have different  $\hat{v}_{t,i}$  in DADAM, one can keep track of the min/max/average of these adaptive learning rates and use this quantity to update the adaptive learning rates. The predefinition of some convergent lower and upper bounds may also lead to a gradual synchronization of the adaptive learning rates on different nodes as developed for AdaBound in [Luo et al., 2019]. In this paper, we opt for the average consensus on  $\hat{v}_{t,i}$ , see consensus update in line 8 and 11 of Algorithm 2. Since for adaptive gradient methods such as AdaGrad or Adam,  $\hat{v}_{t,i}$  approximates the second moment of the gradient estimator, the average of the estimations of those second moments from different nodes is an estimation of second moment on the whole network. Also, this design will not introduce any extra hyperparameters that can potentially complicate the tuning process ( $\epsilon$  in line 9 is important for numerical stability as in vanilla Adam). Our method is presented Algorithm 2. We now present the main convergence result for our class of methods:

**Theorem 2.** Assume A1-A4. Set  $\alpha = 1/\sqrt{Td}$ . When  $\alpha \leq \frac{\epsilon^{0.5}}{16L}$ , Algorithm 2 yields the following regret bound

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] &\leq C_1 \frac{\sqrt{d}}{\sqrt{T}} \left( \mathbb{E}[f(Z_1)] - \min_z f(z) + \frac{\sigma^2}{N} \right) + \frac{C_2}{T} + \frac{C_3}{T^{1.5}d^{0.5}} \\ &\quad + \left( \frac{C_4}{TN^{0.5}} + \frac{C_5}{T^{1.5}d^{0.5}N^{0.5}} \right) \mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right], \end{aligned} \quad (2)$$

where  $\|\cdot\|_{abs}$  denotes the entry-wise  $L_1$  norm of a matrix (i.e.  $\|A\|_{abs} = \sum_{i,j} |A_{i,j}|$ ). The constants  $C_1 = \max(4, 4L/\epsilon)$ ,  $C_2 = 6((\beta_1/(1-\beta_1))^2 + 1/(1-\lambda)^2)LG_\infty^2/\epsilon^{1.5}$ ,  $C_3 = 16L^2(1-\lambda)G_\infty^2/\epsilon^2$ ,  $C_4 = 2/(\epsilon^{1.5}(1-\lambda))(\lambda + \beta_1/(1-\beta_1))G_\infty^2$ ,  $C_5 = 2/(\epsilon^2(1-\lambda))L(\lambda + \beta_1/(1-\beta_1))G_\infty^2 + 4/(\epsilon^2(1-\lambda))LG_\infty^2$  are independent of  $d$ ,  $T$  and  $N$ . In addition,  $\frac{1}{N} \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 \leq \alpha^2 \left( \frac{1}{1-\lambda} \right)^2 dG_\infty^2 \frac{1}{\epsilon}$  which quantifies the consensus error.

Theorem 2 shows that if  $\mathbb{E}[\sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}] = o(T)$  and  $\bar{U}_t$  is upper bounded, then Algorithm 2 is guaranteed to converge to stationary points of the regret function. Intuitively, this means that if the adaptive learning rates on different nodes do not change too fast, the algorithm can converge. This assertion is backed in [Chen et al., 2018] where it is shown that if such condition is violated, the algorithm may diverge. Furthermore, Theorem 2 conveys the benefits of using more nodes in the communication graph. Indeed, as  $N$  becomes larger, the term  $\sigma^2/N$  will be small. This is also strengthened by the fact that with a larger  $N$ , the training process tends to be more stable.

We now present, in Algorithm 3, a notable special case of our algorithmic framework, namely Decentralized AMSGrad, which is a decentralized variant of AMSGrad. Compared with DADAM, the above algorithm exhibits a dynamic average consensus mechanism to keep track of the average of  $\{\hat{v}_{t,i}\}_{i=1}^N$ , stored as  $\tilde{u}_{t,i}$  on  $i$ th node, and uses  $u_{t,i} = \max(\tilde{u}_{t,i}, \epsilon)$  for updating the adaptive learning rate for  $i$ th node. As the number of iteration grows, even though  $\hat{v}_{t,i}$  on different nodes can converge to different constants, all the  $u_{t,i}$  will converge to the same number  $\lim_{t \rightarrow \infty} 1/N \sum_{i=1}^N \hat{v}_{t,i}$  if the limit exists.

---

#### Algorithm 2 Decentralized Adaptive Gradient Method (with N nodes)

---

```

1: Input:  $\alpha$ , initial point  $x_{1,i} = x_{init}, u_{\frac{1}{2},i} = \hat{v}_{0,i}, m_{0,i} = 0$ , mixing matrix  $W$ 
2: for  $t = 1, 2, \dots, T$  do
3:   for all  $i \in [N]$  do in parallel
4:      $g_{t,i} \leftarrow \nabla f_i(x_{t,i}) + \xi_{t,i}$ 
5:      $m_{t,i} = \beta_1 m_{t-1,i} + (1 - \beta_1)g_{t,i}$ 
6:      $\hat{v}_{t,i} = r_t(g_{1,i}, \dots, g_{t,i})$ 
7:      $x_{t+\frac{1}{2},i} = \sum_{j=1}^N W_{ij}x_{t,j}$ 
8:      $\tilde{u}_{t,i} = \sum_{j=1}^N W_{ij}\tilde{u}_{t-\frac{1}{2},j}$ 
9:      $u_{t,i} = \max(\tilde{u}_{t,i}, \epsilon)$ 
10:     $x_{t+1,i} = x_{t+\frac{1}{2},i} - \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}}$ 
11:     $\tilde{u}_{t+\frac{1}{2},i} = \tilde{u}_{t,i} - \hat{v}_{t-1,i} + \hat{v}_{t,i}$ 
12: end for

```

---



Using this average consensus mechanism enables the consensus of adaptive learning rates on different nodes, which accordingly guarantees the convergence of the method to stationary points of a given suboptimality condition. The consensus of adaptive learning rates is the key difference between decentralized AMSGrad and DADAM and is the reason why decentralized AMSGrad is a convergent algorithm while DADAM is not. One may notice that decentralized AMSGrad does not deduce to AMSGrad since the quantity  $u_{t,i}$  in line 10 is calculated based on  $v_{t-1,i}$  instead of  $v_{t,i}$ . This encourages the execution of gradient computation and communication in a parallel manner. Specifically, line 4-7 in Algorithm 3 and Algorithm 2 can be executed in parallel with line 8-9 to overlap communication and computation time. If  $u_{t,i}$  depends on  $v_{t,i}$  which in turn depends on  $g_{t,i}$ , the gradient computation must finish before the consensus step of the adaptive learning rate in line 9. This can slow down the running time per-iteration of the algorithm. To avoid such delayed adaptive learning, adding  $\tilde{u}_{t-\frac{1}{2},i} = \tilde{u}_{t,i} - \hat{v}_{t-1,i} + \hat{v}_{t,i}$  before line 9 and get rid of line 12 in Algorithm 2 is an option. Similar convergence guarantees will hold since one can easily modify our proof of Theorem 2 for such update rule. As stated above, Algorithm 3 converges, with the following rate:

**Theorem 3.** Assume A1-A4. Set  $\alpha = 1/\sqrt{Td}$ . When  $\alpha \leq \frac{\epsilon^{0.5}}{16L}$ , Algorithm 3 yields the following regret bound

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \leq C'_1 \frac{\sqrt{d}}{\sqrt{T}} \left( \mathbb{E}[f(Z_1)] - \min_z f(z) + \frac{\sigma^2}{N} \right) + \frac{C'_2}{T} + \frac{d}{T} \sqrt{N} C'_4 + \frac{\sqrt{d}}{T^{1.5}} \sqrt{N} C'_5,$$

where  $C'_1 = C_1$ ,  $C'_2 = C_2$ ,  $C'_3 = C_3$ ,  $C'_4 = C_4 G_\infty^2$  and  $C'_5 = C_5 G_\infty^2$ .  $C_1, C_2, C_3, C_4, C_5$  are constants independent of  $d, T$  and  $N$  defined in Theorem 2. In addition, the consensus of variables at different nodes is given by  $\frac{1}{N} \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 \leq \frac{1}{T} \left( \frac{1}{1-\lambda} \right)^2 G_\infty^2 \frac{1}{\epsilon}$ .

Theorem 3 shows that Algorithm 3 converges with a rate of  $\mathcal{O}(\sqrt{d}/\sqrt{T})$  when  $T$  is large, which is the best known convergence rate under the given assumptions. Note that in some related works, SGD admits a convergence rate of  $\mathcal{O}(1/\sqrt{T})$  without any dependence on the dimension of the problem. Such improved convergence rate is derived under the assumption that the gradient estimator have a bounded  $L_2$  norm, which can thus hide a dependency of  $\sqrt{d}$  in the final convergence rate.

### 3.3 Convergence Analysis

The detailed proofs of this section are reported in the supplementary material.

**Proof of Theorem 2:** We now present a proof sketch for our main convergence result of Algorithm 2. *Step 1: Reparameterization.* Similarly to [Yan et al., 2018, Chen et al., 2018] with SGD (with momentum) and centralized adaptive gradient methods, define the following auxiliary sequence:

$$Z_t = \bar{X}_t + \frac{\beta_1}{1-\beta_1} (\bar{X}_t - \bar{X}_{t-1}), \quad (3)$$

with  $\bar{X}_0 \triangleq \bar{X}_1$ . Such an auxiliary sequence can help us deal with the bias brought by the momentum and simplifies the convergence analysis. An intermediary result needed to conduct our proof reads:

**Lemma 1.** For the sequence defined in (3), we have

$$Z_{t+1} - Z_t = \alpha \frac{\beta_1}{1-\beta_1} \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) - \alpha \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}}.$$

---

#### Algorithm 3 Decentralized AMSGrad (with $N$ nodes)

---

```

1: Input: learning rate  $\alpha$ , initial point  $x_{1,i} = x_{init}$ ,  $u_{\frac{1}{2},i} = \hat{v}_{0,i} = \epsilon \mathbf{1}$  (with  $\epsilon \geq 0$ ),  $m_{0,i} = 0$ , mixing matrix  $W$ 
2: for  $t = 1, 2, \dots, T$  do
3:   for all  $i \in [N]$  do in parallel
4:      $g_{t,i} \leftarrow \nabla f_i(x_{t,i}) + \xi_{t,i}$ 
5:      $m_{t,i} = \beta_1 m_{t-1,i} + (1 - \beta_1) g_{t,i}$ 
6:      $v_{t,i} = \beta_2 v_{t-1,i} + (1 - \beta_2) g_{t,i}^2$ 
7:      $\hat{v}_{t,i} = \max(\hat{v}_{t-1,i}, v_{t,i})$ 
8:      $x_{t+\frac{1}{2},i} = \sum_{j=1}^N W_{ij} x_{t,j}$ 
9:      $\tilde{u}_{t,i} = \sum_{j=1}^N W_{ij} \tilde{u}_{t-\frac{1}{2},j}$ 
10:     $u_{t,i} = \max(\tilde{u}_{t,i}, \epsilon)$ 
11:     $x_{t+1,i} = x_{t+\frac{1}{2},i} - \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}}$ 
12:     $\tilde{u}_{t+\frac{1}{2},i} = \tilde{u}_{t,i} - \hat{v}_{t-1,i} + \hat{v}_{t,i}$ 
13:  end for
```

---

Lemma 1 does not display any momentum term in  $\frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}}$ . This simplification is convenient since it is directly related to the current gradients instead of the exponential average of past gradients. *Step 2: Smoothness.* Using smoothness assumption A1 involves the following scalar product term:  $\kappa_t := \langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) / \sqrt{\bar{U}_t} \rangle$  which can be lower bounded by:

$$\kappa_t \geq \frac{1}{2} \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{3}{2} \left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{3}{2} \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2.$$

The above inequality substituted in the smoothness condition  $f(Z_{t+1}) \leq f(Z_t) + \langle \nabla f(Z_t), Z_{t+1} - Z_t \rangle + \frac{L}{2} \|Z_{t+1} - Z_t\|^2$  yields:

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \leq \frac{2}{T\alpha} (\mathbb{E}[\Delta_f]) + \frac{L}{T\alpha} \sum_{t=1}^T \mathbb{E} [\|Z_{t+1} - Z_t\|^2] + \frac{2}{T} \frac{\beta_1}{1 - \beta_1} D_1 + \frac{2}{T} D_2 + \frac{3}{T} D_3, \quad (4)$$

where  $\Delta_f := \mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]$ .  $D_1, D_2$  and  $D_3$  are three terms, defined in the supplementary material, and which can be tightly bounded from above. We first bound  $D_3$  using the following quantities of interest:

$$\sum_{t=1}^T \|Z_t - \bar{X}_t\|^2 \leq T \left( \frac{\beta_1}{1 - \beta_1} \right)^2 \alpha^2 d \frac{G_\infty^2}{\epsilon} \quad \text{and} \quad \sum_{t=1}^T \frac{1}{N} \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 \leq T \alpha^2 \left( \frac{1}{1 - \lambda} \right)^2 d G_\infty^2 \frac{1}{\epsilon}.$$

where  $\lambda = \max(|\lambda_2|, |\lambda_N|)$  and recall that  $\lambda_i$  is  $i$ th largest eigenvalue of  $W$ .

Then, concerning the term  $D_2$ , few derivations, not detailed here for simplicity, yields:

$$D_2 \leq \frac{G_\infty^2}{N} \mathbb{E} \left[ \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \left\| -\sum_{l=2}^N \tilde{U}_t q_l q_l^T \right\|_{abs} \right],$$

where  $q_l$  is the eigenvector corresponding to  $l$ th largest eigenvalue of  $W$  and  $\|\cdot\|_{abs}$  is the entry-wise  $L_1$  norm of matrices. We can also show that

$$\sum_{t=1}^T \left\| -\sum_{l=2}^N \tilde{U}_t q_l q_l^T \right\|_{abs} \leq \sqrt{N} \sum_{o=0}^{T-1} \frac{\lambda}{1 - \lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs},$$

resulting in an upper bound for  $D_2$  proportional to  $\sum_{o=0}^{T-1} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs}$ . Similarly:

$$D_1 \leq G_\infty^2 \frac{1}{2\epsilon^{1.5}} \frac{1}{\sqrt{N}} \mathbb{E} \left[ \frac{1}{1 - \lambda} \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right].$$

*Step 3: Bounding the drift term variance.* An important term that needs upper bounding in our proof is the variance of the gradients multiplied (element-wise) by the adaptive learning rate:

$$\mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] \leq \mathbb{E}[\|\Gamma_u^f\|^2] + \frac{d}{N} \frac{\sigma^2}{\epsilon},$$

where  $\Gamma_u^f := 1/N \sum_{i=1}^N \nabla f_i(x_{t,i}) / \sqrt{u_{t,i}}$ . Two consecutive and simple bounding of the above yields:

$$\sum_{t=1}^T \mathbb{E}[\|\Gamma_u^f\|^2] \leq 2 \sum_{t=1}^T \mathbb{E}[\|\Gamma_{\bar{U}}^f\|^2] + 2 \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N G_\infty^2 \frac{1}{\sqrt{\epsilon}} \left\| \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right\| \right]$$

and

$$\sum_{t=1}^T \mathbb{E}[\|\Gamma_{\bar{U}}^f\|^2] \leq 2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(\bar{X}_t) - \nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right]. \quad (5)$$

Then, by plugging the LHS of (5) in (4), and further bounding as operated for  $D_2, D_3$  (see supplement), we obtain the desired bound in Theorem 2.

**Proof of Theorem 3:** Recall the bound in (2) of Theorem 2. Since Algorithm 3 is a special case of Algorithm 2, the remaining of the proof consists in characterizing the growth rate of  $\mathbb{E}[\sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}]$ . By construction,  $\hat{V}_t$  is non decreasing, then it can be shown that  $\mathbb{E}[\sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}] = \mathbb{E}[\sum_{i=1}^N \sum_{j=1}^d (-[\hat{v}_{0,i}]_j + [\hat{v}_{T-1,i}]_j)]$ . Besides, since for all  $t, i, \|g_{t,i}\|_\infty \leq G_\infty$  and  $v_{t,i}$  is an exponential moving average of  $g_{k,i}^2, k = 1, 2, \dots, t$ , we have  $|\hat{v}_{t,i}|_j \leq G_\infty^2$  for all  $t, i, j$ . By construction of  $\hat{V}_t$ , we also observe that each element of  $\hat{V}_t$  cannot be greater than  $G_\infty^2$ , i.e.  $|\hat{v}_{t,i}|_j \leq G_\infty^2$  for all  $t, i, j$ . Given that  $[\hat{v}_{0,i}]_j \geq 0$ , we have

$$\mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right] = \mathbb{E} \left[ \sum_{i=1}^N \sum_{j=1}^d (-[\hat{v}_{0,i}]_j + [\hat{v}_{T-1,i}]_j) \right] \leq \sum_{i=1}^N \sum_{j=1}^d \mathbb{E}[G_\infty^2] = NdG_\infty^2.$$

Substituting into (2) yields the desired convergence bound for Algorithm 3.

### 3.4 Illustrative Numerical Experiments

In this section, we conduct some experiments to test the performance of Decentralized AMSGrad, developed in Algorithm 3, on both *homogeneous* data and *heterogeneous* data distribution (i.e. the data generating distribution on different nodes are assumed to be different). Comparison with DADAM and the decentralized stochastic gradient descent (DGD) developed in [Lian et al., 2017] are conducted. We train a Convolutional Neural Network (CNN) with 3 convolution layers followed by a fully connected layer on MNIST [LeCun, 1998]. We set  $\epsilon = 10^{-6}$  for both Decentralized AMSGrad and DADAM. The learning rate is chosen from the grid  $[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$  based on validation accuracy for all algorithms. In the following experiments, the graph contains 5 nodes and each node can only communicate with its two adjacent neighbors forming a cycle. Regarding the mixing matrix  $W$ , we set  $W_{ij} = 1/3$  if nodes  $i$  and  $j$  are neighbors and  $W_{ij} = 0$  otherwise. More details and experiments can be found in the supplementary material of our paper.

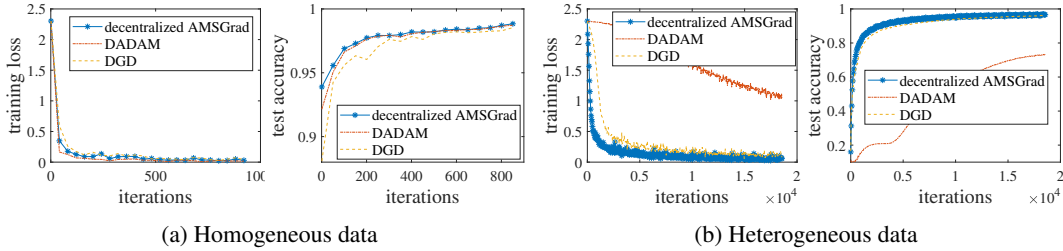


Figure 1: Training loss and Testing accuracy for homogeneous and heterogeneous data

**Homogeneous data:** The whole dataset is shuffled and evenly split into different nodes. We see, Figure 1(a), that decentralized AMSGrad and DADAM perform quite similarly while DGD is much slower both in terms of training loss and test accuracy. Though the (possible) non convergence of DADAM, mentioned in this paper, its performance are empirically good on homogeneous data. The reason is that the adaptive learning rates tend to be similar on different nodes in presence of homogeneous data distribution. We thus compare these algorithms under the heterogeneous regime.

**Heterogeneous data:** Here, each node only contains training data with two labels out of ten. We can see that each algorithm converges significantly slower than with homogeneous data. Especially, the performance of DADAM deteriorates significantly. Decentralized AMSGrad achieves the best training and testing performance in that setting as observed Figure 1(b).

## 4 Conclusion

This paper studies the problem of designing adaptive gradient methods for decentralized training. We propose a unifying algorithmic framework that can convert existing adaptive gradient methods to decentralized settings. With rigorous convergence analysis, we show that if the original algorithm satisfies converges under some minor conditions, the converted algorithm obtained using our proposed framework is guaranteed to converge to stationary points of the regret function. By applying our framework to AMSGrad, we propose the first convergent adaptive gradient methods, namely Decentralized AMSGrad. Experiments show that the proposed algorithm achieves better performance than the baselines.



## 5 Broader Impact

We hope that efforts towards developing decentralized optimization methods can be put to good use for practical applications where data can not be shared in a central server for privacy reasons. Indeed, when the data is sensible and captured on several devices (nodes), we must come up with efficient and low-cost optimization methods for fitting complex models. We believe our work is a step forward leveraging current state-of-the-art optimization methods for decentralized optimization.

## References

- Naman Agarwal, Brian Bullins, Xinyi Chen, Elad Hazan, Karan Singh, Cyril Zhang, and Yi Zhang. The case for full-matrix adaptive regularization. *arXiv preprint arXiv:1806.02958*, 2018.
- Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017.
- Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.
- Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael Rabbat. Stochastic gradient push for distributed deep learning. *arXiv preprint arXiv:1811.10792*, 2018.
- Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
- Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv preprint arXiv:1808.02941*, 2018.
- Yongjian Chen, Tao Guan, and Cheng Wang. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 10(12):11259–11273, 2010.
- Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pages 571–582, 2014.
- Paolo Di Lorenzo and Gesualdo Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011a.
- John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011b.
- Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2946–2953, 2013.
- Mingyi Hong, Davood Hajinezhad, and Ming-Min Zhao. Prox-pda: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1529–1538. JMLR.org, 2017.
- Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.

385 Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive  
386 stepsizes. *arXiv preprint arXiv:1805.08114*, 2018.

387 Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized  
388 algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic  
389 gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.

390 Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression:  
391 Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*,  
392 2017.

393 Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic  
394 bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

395 H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient  
396 learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

397 Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based  
398 distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*,  
399 2019.

400 Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization.  
401 *IEEE Transactions on Automatic Control*, 54(1):48, 2009.

402 Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný,  
403 Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint*  
404 *arXiv:2003.00295*, 2020.

405 Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv*  
406 *preprint arXiv:1904.09237*, 2019.

407 Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical*  
408 *statistics*, pages 400–407, 1951.

409 Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized  
410 consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

411 Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In  
412 *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.

413 Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D<sup>2</sup>: Decentralized training over  
414 decentralized data. *arXiv preprint arXiv:1803.07068*, 2018.

415 Hanlin Tang, Xiangru Lian, Tong Zhang, and Ji Liu. Doublesqueeze: Parallel stochastic gradient  
416 descent with double-pass error-compensated compression. *arXiv preprint arXiv:1905.05957*, 2019.

417 Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen  
418 Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in*  
419 *Neural Information Processing Systems*, pages 9850–9861, 2018.

420 Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-  
421 efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pages  
422 1299–1309, 2018.

423 Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex  
424 landscapes, from any initialization. *arXiv preprint arXiv:1806.01811*, 2018.

425 Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic momentum  
426 methods for deep learning. *arXiv preprint arXiv:1808.10396*, 2018.

427 Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM*  
428 *Journal on Optimization*, 26(3):1835–1854, 2016.

- 429 Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods  
430 for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages  
431 9793–9803, 2018.
- 432 Fangyu Zou and Li Shen. On the convergence of weighted adagrad with momentum for training deep  
433 neural networks. *arXiv preprint arXiv:1808.03408*, 2018.

## 434 A Proof of Auxiliary Lemmas

435 **Lemma 1.** *For the sequence defined in (9), we have*

$$Z_{t+1} - Z_t = \alpha \frac{\beta_1}{1 - \beta_1} \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) - \alpha \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}}. \quad (6)$$

436 **Proof:** By update rule of Algorithm 2, we first have

$$\begin{aligned} \bar{X}_{t+1} &= \frac{1}{N} \sum_{i=1}^N x_{t+1,i} \\ &= \frac{1}{N} \sum_{i=1}^N \left( x_{t+0.5,i} - \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}} \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \sum_{j=1}^N W_{ij} x_{t,j} - \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}} \right) \\ &\stackrel{(i)}{=} \left( \frac{1}{N} \sum_{j=1}^N x_{t,j} \right) - \frac{1}{N} \sum_{i=1}^N \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}} \\ &= \bar{X}_t - \frac{1}{N} \sum_{i=1}^N \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}}, \end{aligned}$$

437 where (i) is due to an interchange of summation and  $\sum_{i=1}^N W_{ij} = 1$ . Then, we have

$$\begin{aligned} Z_{t+1} - Z_t &= \bar{X}_{t+1} - \bar{X}_t + \frac{\beta_1}{1 - \beta_1} (\bar{X}_{t+1} - \bar{X}_t) - \frac{\beta_1}{1 - \beta_1} (\bar{X}_{t+1} - \bar{X}_t) \\ &= \frac{1}{1 - \beta_1} (\bar{X}_{t+1} - \bar{X}_t) - \frac{\beta_1}{1 - \beta_1} (\bar{X}_{t+1} - \bar{X}_t) \\ &= \frac{1}{1 - \beta_1} \left( -\frac{1}{N} \sum_{i=1}^N \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}} \right) - \frac{\beta_1}{1 - \beta_1} \left( -\frac{1}{N} \sum_{i=1}^N \alpha \frac{m_{t-1,i}}{\sqrt{u_{t-1,i}}} \right) \\ &= \frac{1}{1 - \beta_1} \left( -\frac{1}{N} \sum_{i=1}^N \alpha \frac{\beta_1 m_{t-1,i} + (1 - \beta_1) g_{t,i}}{\sqrt{u_{t,i}}} \right) - \frac{\beta_1}{1 - \beta_1} \left( -\frac{1}{N} \sum_{i=1}^N \alpha \frac{m_{t-1,i}}{\sqrt{u_{t-1,i}}} \right) \\ &= \alpha \frac{\beta_1}{1 - \beta_1} \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) - \alpha \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}}, \end{aligned}$$

438 which is the desired result.  $\square$

439 **Lemma 2.** *Given a set of numbers  $a_1, \dots, a_n$  and denote their mean to be  $\bar{a} = \frac{1}{n} \sum_{i=1}^n a_i$ . In*  
 440 *addition, define  $b_i(r) \triangleq \max(a_i, r)$  and  $\bar{b}(r) = \frac{1}{n} \sum_{i=1}^n b_i(r)$ . For any  $r$  and  $r'$  with  $r' \geq r$  we*  
 441 *have*

$$\sum_{i=1}^n |b_i(r) - \bar{b}(r)| \geq \sum_{i=1}^n |b_i(r') - \bar{b}(r')| \quad (7)$$

442 and when  $r \leq \min_{i \in [n]} a_i$ , we have

$$\sum_{i=1}^n |b_i(r) - \bar{b}(r)| = \sum_{i=1}^n |a_i - \bar{a}|. \quad (8)$$

443 **Proof:** Without loss of generality, assume  $a_i \leq a_j$  when  $i < j$ , i.e.  $a_i$  is a non-decreasing sequence.  
 444 Define

$$h(r) = \sum_{i=1}^n |b_i(r) - \bar{b}(r)| = \sum_{i=1}^n \left| \max(a_i, r) - \frac{1}{n} \sum_{j=1}^n \max(a_j, r) \right|.$$

445 We need to prove that  $h$  is a non-increasing function of  $r$ . First, it is easy to see that  $h$  is a continuous  
 446 function of  $r$  with non-differentiable points  $r = a_i, i \in [n]$ , thus  $h$  is a piece-wise linear function.

447 Next, we will prove that  $h(r)$  is non-increasing in each piece. Define  $l(r)$  to be the largest index with  
 448  $a(l(r)) < r$ , and  $s(r)$  to be the largest index with  $a_{s(r)} < \bar{b}(r)$ . Note that we have  $b_i(r) = r, \forall i \leq l(r)$   
 449 and  $b_i(r) - \bar{b}(r) \leq 0, \forall i \leq s(r)$  because  $a_i$  is a non-decreasing sequence. Therefore, we have

$$h(r) = \sum_{i=1}^{l(r)} (\bar{b}(r) - r) + \sum_{i=l(r)+1}^{s(r)} (\bar{b}(r) - a_i) + \sum_{i=s(r)+1}^n (a_i - \bar{b}(r))$$

450 and

$$\bar{b}(r) = \frac{1}{n} \left( l(r)r + \sum_{i=l(r)+1}^n a_i \right).$$

451 Taking derivative of the above form, we know the derivative of  $h(r)$  at differentiable points is

$$\begin{aligned} h'(r) &= l(r) \left( \frac{l(r)}{n} - 1 \right) + (s(r) - l(r)) \frac{l(r)}{n} - (n - s(r)) \frac{l(r)}{n} \\ &= \frac{l(r)}{n} ((l(r) - n) + (s(r) - l(r)) - (n - s(r))). \end{aligned}$$

452 Since we have  $s(r) \leq n$  we know  $(l(r) - n) + (s(r) - l(r)) - (n - s(r)) \leq 0$  and thus

$$h'(r) \leq 0,$$

453 which means  $h(r)$  is non-increasing in each piece. Combining with the fact that  $h(r)$  is continuous,  
 454 (7) is proven. When  $r \leq a(i)$ , we have  $b(i) = \max(a_i, r) = r, \forall r \in [n]$  and  $\bar{b}(r) = \frac{1}{n} \sum_{i=1}^n a_i = \bar{a}$   
 455 which proves (8).  $\square$

## 456 B Proof of Theorem 2

457 To prove convergence of the algorithm, we first define an auxiliary sequence

$$Z_t = \bar{X}_t + \frac{\beta_1}{1 - \beta_1} (\bar{X}_t - \bar{X}_{t-1}), \quad (9)$$

458 with  $\bar{X}_0 \triangleq \bar{X}_1$ . Since  $\mathbb{E}[g_{t,i}] = \nabla f(x_{t,i})$  and  $u_{t,i}$  is a function of  $G_{1:t-1}$  (which denotes  
 459  $G_1, G_2, \dots, G_{t-1}$ ), we have

$$\mathbb{E}_{G_t|G_{1:t-1}} \left[ \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right] = \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}}.$$

460 Assuming smoothness (A1) we have

$$f(Z_{t+1}) \leq f(Z_t) + \langle \nabla f(Z_t), Z_{t+1} - Z_t \rangle + \frac{L}{2} \|Z_{t+1} - Z_t\|^2.$$

461 Using Lemma 1 into the above inequality and take expectation over  $G_t$  given  $G_{1:t-1}$ , we have

$$\begin{aligned} &\mathbb{E}_{G_t|G_{1:t-1}} [f(Z_{t+1})] \\ &\leq f(Z_t) - \alpha \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\rangle + \frac{L}{2} \mathbb{E}_{G_t|G_{1:t-1}} [\|Z_{t+1} - Z_t\|^2] \\ &\quad + \alpha \frac{\beta_1}{1 - \beta_1} \mathbb{E}_{G_t|G_{1:t-1}} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right]. \end{aligned}$$



462 Then take expectation over  $G_{1:t-1}$  and rearrange, we have

$$\alpha \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\rangle \right] \quad (10)$$

$$\leq \mathbb{E}[f(Z_t)] - \mathbb{E}[f(Z_{t+1})] + \frac{L}{2} \mathbb{E}[\|Z_{t+1} - Z_t\|^2] \\ + \alpha \frac{\beta_1}{1 - \beta_1} \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right]. \quad (11)$$

463 In addition, we have

$$\left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\rangle \\ = \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\rangle + \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right) \right\rangle \quad (12)$$

464 and the first term on RHS of the equality can be lower bounded as

$$\left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\rangle \\ = \frac{1}{2} \left\| \frac{\nabla f(Z_t)}{\bar{U}_t^{1/4}} \right\|^2 + \frac{1}{2} \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i})}{\bar{U}_t^{1/4}} \right\|^2 - \frac{1}{2} \left\| \frac{\nabla f(Z_t) - \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i})}{\bar{U}_t^{1/4}} \right\|^2 \\ \geq \frac{1}{4} \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 + \frac{1}{4} \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{1}{2} \left\| \frac{\nabla f(Z_t) - \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i})}{\bar{U}_t^{1/4}} \right\|^2 \\ - \frac{1}{2} \left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{1}{2} \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \\ \geq \frac{1}{2} \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{3}{2} \left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{3}{2} \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2, \quad (13)$$

465 where the inequalities are all due to Cauchy-Schwartz. Substituting (13) and (12) into (10), we get

$$\frac{1}{2} \alpha \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \leq \mathbb{E}[f(Z_t)] - \mathbb{E}[f(Z_{t+1})] + \frac{L}{2} \mathbb{E}[\|Z_{t+1} - Z_t\|^2] \\ + \alpha \frac{\beta_1}{1 - \beta_1} \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right] \\ - \alpha \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right) \right\rangle \right] \\ + \frac{3}{2} \alpha \mathbb{E} \left[ \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 + \left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right].$$

466 Then sum over the above inequality from  $t = 1$  to  $T$  and divide both sides by  $T\alpha/2$ , we have

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \\
& \leq \frac{2}{T\alpha} (\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + \frac{L}{T\alpha} \sum_{t=1}^T \mathbb{E} [\|Z_{t+1} - Z_t\|^2] \\
& \quad + \underbrace{\frac{2}{T} \frac{\beta_1}{1-\beta_1} \sum_{t=1}^T \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right]}_{D_1} \\
& \quad + \underbrace{\frac{2}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{\bar{U}_t}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right]}_{D_2} \\
& \quad + \underbrace{\frac{3}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 + \left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right]}_{D_3}. \tag{14}
\end{aligned}$$

467 Now we need to upper bound all the terms on RHS of the above inequality to get the convergence  
 468 rate. For the terms composing  $D_3$  in (14), we can upper bound them by

$$\begin{aligned}
\left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 & \leq \frac{1}{\min_{j \in [d]} [\bar{U}_t^{1/2}]_j} \|\nabla f(Z_t) - \nabla f(\bar{X}_t)\|^2 \\
& \leq L \frac{1}{\min_{j \in [d]} [\bar{U}_t^{1/2}]_j} \underbrace{\|Z_t - \bar{X}_t\|^2}_{D_4} \tag{15}
\end{aligned}$$

469 and

$$\begin{aligned}
\left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 & \leq \frac{1}{\min_{j \in [d]} [\bar{U}_t^{1/2}]_j} \frac{1}{N} \sum_{i=1}^N \|\nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)\|^2 \\
& \leq L \frac{1}{\min_{j \in [d]} [\bar{U}_t^{1/2}]_j} \frac{1}{N} \underbrace{\sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2}_{D_5}, \tag{16}
\end{aligned}$$

470 using Jensen's inequality, Lipschitz continuity of  $f_i$ , and the fact that  $f = \frac{1}{N} \sum_{i=1}^N f_i$ . Next we need  
 471 to bound  $D_4$  and  $D_5$ . Recall the update rule of  $X_t$ , we have

$$X_t = X_{t-1}W - \alpha \frac{M_{t-1}}{\sqrt{U_{t-1}}} = X_1 W^{t-1} - \alpha \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} W^k, \tag{17}$$

472 where we define  $W^0 = \mathbf{I}$ . Since  $W$  is a symmetric matrix, we can decompose it as  $W = Q\Lambda Q^T$   
 473 where  $Q$  is a orthonormal matrix and  $\Lambda$  is a diagonal matrix whose diagonal elements correspond  
 474 to eigenvalues of  $W$  in an descending order, i.e.  $\Lambda_{ii} = \lambda_i$  with  $\lambda_i$  being  $i$ th largest eigenvalue of  
 475  $W$ . In addition, because  $W$  is a doubly stochastic matrix, we know  $\lambda_1 = 1$  and  $q_1 = \frac{1}{\sqrt{N}}$ . With  
 476 eigen-decomposition of  $W$ , we can rewrite  $D_5$  as

$$\sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 = \|X_t - \bar{X}_t \mathbf{1}_N^T\|_F^2 = \|X_t Q Q^T - X_t \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T\|_F^2 = \sum_{l=2}^N \|X_t q_l\|^2. \tag{18}$$

477 In addition, we can rewrite (17) as

$$X_t = X_1 W^{t-1} - \alpha \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} W^k = X_1 - \alpha \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} Q \Lambda^k Q^T, \quad (19)$$

478 where the last equality is because  $x_{1,i} = x_{1,j}, \forall i, j$  and thus  $X_1 W = X_1$ . Then we have when  $l > 1$ ,

$$X_t q_l = (X_1 - \alpha \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} Q \Lambda^k Q^T) q_l = -\alpha \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} q_l \lambda_l^k, \quad (20)$$

479 since  $Q$  is orthonormal and  $X_1 q_l = x_{1,1} \mathbf{1}_N^T q_l = x_{1,1} \sqrt{N} q_1^T q_l = 0, \forall l \neq 1$ .

480 Combining (18) and (20), we have

$$\begin{aligned} D_5 &= \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 = \sum_{l=2}^N \|X_t q_l\|^2 \\ &= \sum_{l=2}^N \alpha^2 \left\| \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} \lambda_l^k q_l \right\|^2 \\ &\leq \alpha^2 \left( \frac{1}{1-\lambda} \right)^2 N d G_\infty^2 \frac{1}{\epsilon}, \end{aligned} \quad (21)$$

481 where the last inequality follows from the fact that  $g_{t,i} \leq G_\infty, \|q_l\| = 1$ , and  $|\lambda_l| \leq \lambda < 1$ . Now let  
482 us turn to  $D_4$ , it can be rewritten as

$$\begin{aligned} \|Z_t - \bar{X}_t\|^2 &= \left\| \frac{\beta_1}{1-\beta_1} (\bar{X}_t - \bar{X}_{t-1}) \right\|^2 = \left( \frac{\beta_1}{1-\beta_1} \right)^2 \alpha^2 \left\| \frac{1}{N} \sum_{i=1}^N \frac{m_{t-1,i}}{\sqrt{u_{t-1,i}}} \right\|^2 \\ &\leq \left( \frac{\beta_1}{1-\beta_1} \right)^2 \alpha^2 d \frac{G_\infty^2}{\epsilon}. \end{aligned} \quad (22)$$

483 Now we know both  $D_4$  and  $D_5$  are in the order of  $\mathcal{O}(\alpha^2)$  and thus  $D_3$  is in the order of  
484  $\mathcal{O}(\alpha^2)$ . Next we will bound  $D_2$  and  $D_1$ . Define  $G_1 \triangleq \max_{t \in [T]} \max_{i \in [N]} \|\nabla f_i(x_{t,i})\|_\infty$ ,  
485  $G_2 \triangleq \max_{t \in [T]} \|\nabla f(Z_t)\|_\infty$ ,  $G_3 \triangleq \max_{t \in [T]} \max_{i \in [N]} \|g_{t,i}\|_\infty$  and  $G_\infty = \max(G_1, G_2, G_3)$ .  
486 Then we have

$$\begin{aligned} D_2 &= \sum_{t=1}^T \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{U_t}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right] \\ &\leq \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{1}{\sqrt{[U_t]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right| \right] \\ &= \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{1}{\sqrt{[U_t]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right| \frac{\sqrt{[U_t]_j} + \sqrt{[u_{t,i}]_j}}{\sqrt{[U_t]_j} + \sqrt{[u_{t,i}]_j}} \right] \\ &= \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{[U_t]_j - [u_{t,i}]_j}{[U_t]_j \sqrt{[u_{t,i}]_j} + \sqrt{[U_t]_j} [u_{t,i}]_j} \right| \right] \\ &\leq \underbrace{\mathbb{E} \left[ \sum_{t=1}^T G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{[U_t]_j - [u_{t,i}]_j}{2\epsilon^{1.5}} \right| \right]}_{T_6}, \end{aligned} \quad (23)$$

487 where the last inequality is due to  $[u_{t,i}]_j \geq \epsilon, \forall t, i, j$ .

488 To simplify notations, define  $\|A\|_{abs} = \sum_{i,j} |A_{ij}|$  to be the entry-wise  $L_1$  norm of a matrix  $A$ , then  
 489 we obtain

$$\begin{aligned}
 T_6 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \|\bar{U}_t \mathbf{1}^T - U_t\|_{abs} \\
 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \|\bar{\tilde{U}}_t \mathbf{1}^T - \tilde{U}_t\|_{abs} \\
 &= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \|\tilde{U}_t \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T - \tilde{U}_t Q Q^T\|_{abs} \\
 &= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \left\| -\tilde{U}_t \sum_{l=2}^N q_l q_l^T \right\|_{abs} \\
 &= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \left\| -\sum_{l=2}^N \tilde{U}_t q_l q_l^T \right\|_{abs},
 \end{aligned}$$

490 where the second inequality is due to Lemma 2, introduced Section A, and the fact that  $U_t =$   
 491  $\max(\tilde{U}_t, \epsilon)$  (element-wise max operator). Recall from update rule of  $U_t$ , by defining  $\hat{V}_{-1} \triangleq \hat{V}_0$  and  
 492  $U_0 \triangleq U_{1/2}$ , we have  $\forall t \geq 0$

$$\tilde{U}_{t+1} = (\tilde{U}_t - \hat{V}_{t-1} + \hat{V}_t)W$$

493 and thus

$$\tilde{U}_t = \tilde{U}_0 W^t + \sum_{k=1}^t (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) W^k = \tilde{U}_0 + \sum_{k=1}^t (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) Q \Lambda^k Q^T.$$

494 Then we further obtain when  $l \neq 1$ ,

$$\tilde{U}_t q_l = (\tilde{U}_0 + \sum_{k=1}^t (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) Q \Lambda^k Q^T) q_l = \sum_{k=1}^t (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) q_l \lambda_l^k,$$

495 where the last equality is due to the definition  $\tilde{U}_0 \triangleq U_{1/2} = \epsilon \mathbf{1}_d \mathbf{1}_N^T = \sqrt{N} \epsilon \mathbf{1}_d \mathbf{1}_N^T$  (recall that  
 496  $q_1 = \frac{1}{\sqrt{N}} \mathbf{1}_N^T$ ) and  $q_i^T q_j = 0$  when  $i \neq j$ . Note that by definition of  $\|\cdot\|_{abs}$ , we have  $\forall A, B, \|A +$   
 497  $B\|_{abs} \leq \|A\|_{abs} + \|B\|_{abs}$ , then

$$\begin{aligned}
 T_6 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \left\| -\sum_{l=2}^N \tilde{U}_t q_l q_l^T \right\|_{abs} \\
 &= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \left\| -\sum_{k=1}^t (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) \sum_{l=2}^N q_l \lambda_l^k q_l^T \right\|_{abs} \\
 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \left\| (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) \sum_{l=2}^N q_l \lambda_l^k q_l^T \right\|_{abs} \\
 &= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \sum_{j=1}^d \left\| \sum_{l=2}^N q_l \lambda_l^k q_l^T (-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j \right\|_1 \\
 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \sum_{j=1}^d \left\| \sum_{l=2}^N q_l \lambda_l^k q_l^T \right\|_1 \left\| (-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j \right\|_1 \\
 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \sum_{j=1}^d \sqrt{N} \left\| \sum_{l=2}^N q_l \lambda_l^k q_l^T \right\|_2 \left\| (-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j \right\|_1 \\
 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \sum_{j=1}^d \left\| (-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j \right\|_1 \sqrt{N} \lambda^k
 \end{aligned}$$

$$\begin{aligned}
&= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \|(-\hat{V}_{t-1-k} + \hat{V}_{t-k})\|_{abs} \sqrt{N} \lambda^k \\
&= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{o=0}^{t-1} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \sqrt{N} \lambda^{t-o} \\
&= \frac{G_\infty^2}{N} \frac{1}{2\epsilon^{1.5}} \sum_{o=0}^{T-1} \sum_{t=o+1}^T \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \sqrt{N} \lambda^{t-o} \\
&\leq \frac{G_\infty^2}{\sqrt{N}} \frac{1}{2\epsilon^{1.5}} \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs}, \tag{24}
\end{aligned}$$

498 where  $\lambda = \max(|\lambda_2|, |\lambda_N|)$ . Combining (23) and (24), we have

$$D_2 \leq \frac{G_\infty^2}{\sqrt{N}} \frac{1}{2\epsilon^{1.5}} \frac{\lambda}{1-\lambda} \mathbb{E} \left[ \sum_{o=0}^{T-1} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \right].$$

499 Now we need to bound  $D_1$ , we have

$$\begin{aligned}
D_1 &= \sum_{t=1}^T \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right] \\
&\leq \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{1}{\sqrt{[u_{t-1,i}]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right| \right] \\
&= \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \left( \frac{1}{\sqrt{[u_{t-1,i}]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right) \frac{\sqrt{[u_{t,i}]_j} + \sqrt{[u_{t-1,i}]_j}}{\sqrt{[u_{t,i}]_j} + \sqrt{[u_{t-1,i}]_j}} \right| \right] \\
&\leq \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{1}{2\epsilon^{1.5}} ([u_{t-1,i}]_j - [u_{t,i}]_j) \right| \right] \\
&\stackrel{(a)}{\leq} \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \frac{1}{2\epsilon^{1.5}} |([\tilde{u}_{t-1,i}]_j - [\tilde{u}_{t,i}]_j)| \right] \\
&= G_\infty^2 \frac{1}{2\epsilon^{1.5}} \frac{1}{N} \mathbb{E} \left[ \sum_{t=1}^T \|\tilde{U}_{t-1} - \tilde{U}_t\|_{abs} \right], \tag{25}
\end{aligned}$$



500 where (a) is due to  $[\tilde{u}_{t-1,i}]_j = \max([u_{t-1,i}]_j, \epsilon)$  and the function  $\max(\cdot, \epsilon)$  is 1-Lipschitz. In  
 501 addition, by update rule of  $U_t$ , we have

$$\begin{aligned}
& \sum_{t=1}^T \|\tilde{U}_{t-1} - \tilde{U}_t\|_{abs} \\
&= \sum_{t=1}^T \|\tilde{U}_{t-1} - (\tilde{U}_{t-1} - \hat{V}_{t-2} + \hat{V}_{t-1})W\|_{abs} \\
&= \sum_{t=1}^T \|\tilde{U}_{t-1}(QQ^T - Q\Lambda Q^T) + (-\hat{V}_{t-2} + \hat{V}_{t-1})W\|_{abs} \\
&= \sum_{t=1}^T \|\tilde{U}_{t-1}(\sum_{l=2}^N q_l(1 - \lambda_l)q_l^T) + (-\hat{V}_{t-2} + \hat{V}_{t-1})W\|_{abs} \\
&\leq \sum_{t=1}^T \left\| \sum_{k=1}^{t-1} (-\hat{V}_{t-2-k} + \hat{V}_{t-1-k}) \sum_{l=2}^N q_l \lambda_l^k (1 - \lambda_l) q_l^T \right\|_{abs} + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})W\|_{abs} \\
&\leq \sum_{t=1}^T \left( \sum_{k=1}^{t-1} \|\hat{V}_{t-2-k} - \hat{V}_{t-1-k}\|_{abs} \sqrt{N} \lambda^k \right) + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \\
&= \sum_{t=1}^T \left( \sum_{o=1}^{t-1} \|\hat{V}_{o-2} - \hat{V}_{o-1}\|_{abs} \sqrt{N} \lambda^{t-o} \right) + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \\
&= \sum_{o=1}^{T-1} \sum_{t=o+1}^T \left( \|\hat{V}_{o-2} - \hat{V}_{o-1}\|_{abs} \sqrt{N} \lambda^{t-o} \right) + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \\
&\leq \sum_{o=1}^{T-1} \frac{\lambda}{1-\lambda} \left( \|\hat{V}_{o-2} - \hat{V}_{o-1}\|_{abs} \sqrt{N} \right) + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \\
&\leq \frac{1}{1-\lambda} \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \sqrt{N}.
\end{aligned} \tag{26}$$

502 Combining (25) and (26), we have

$$D_1 \leq G_\infty^2 \frac{1}{2\epsilon^{1.5}} \frac{1}{N} \mathbb{E} \left[ \frac{1}{1-\lambda} \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \sqrt{N} \right]. \tag{27}$$

503 What remains is to bound  $\sum_{t=1}^T \mathbb{E} [\|Z_{t+1} - Z_t\|^2]$ . By update rule of  $Z_t$ , we have

$$\begin{aligned}
& \|Z_{t+1} - Z_t\|^2 \\
&= \left\| \alpha \frac{\beta_1}{1-\beta_1} \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) - \alpha \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \\
&\leq 2\alpha^2 \left\| \frac{\beta_1}{1-\beta_1} \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\|^2 + 2\alpha^2 \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \\
&\leq 2\alpha^2 \left( \frac{\beta_1}{1-\beta_1} \right)^2 G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \frac{1}{\sqrt{\epsilon}} \left| \frac{1}{\sqrt{[u_{t-1,i}]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right| + 2\alpha^2 \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \\
&\leq 2\alpha^2 \left( \frac{\beta_1}{1-\beta_1} \right)^2 G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \frac{1}{\sqrt{\epsilon}} \left| \frac{[u_{t,i}]_j - [u_{t-1,i}]_j}{2\epsilon^{1.5}} \right| + 2\alpha^2 \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \\
&\leq 2\alpha^2 \left( \frac{\beta_1}{1-\beta_1} \right)^2 G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \frac{1}{2\epsilon^2} |\tilde{u}_{t,i}]_j - [\tilde{u}_{t-1,i}]_j| + 2\alpha^2 \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2
\end{aligned}$$

$$= 2\alpha^2 \left( \frac{\beta_1}{1-\beta_1} \right)^2 G_\infty^2 \frac{1}{N} \frac{1}{2\epsilon^2} \|\tilde{U}_t - \tilde{U}_{t-1}\|_{abs} + 2\alpha^2 \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2, \quad (28)$$

504 where the last inequality is again due to the definition that  $[\tilde{u}_{t,i}]_j = \max([u_{t,i}]_j, \epsilon)$  and the fact that  
 505  $\max(\cdot, \epsilon)$  is 1-Lipschitz.

506 Then, we have

$$\begin{aligned} & \sum_{t=1}^T \mathbb{E}[\|Z_{t+1} - Z_t\|^2] \\ & \leq 2\alpha^2 \left( \frac{\beta_1}{1-\beta_1} \right)^2 G_\infty^2 \frac{1}{N} \frac{1}{2\epsilon^2} \mathbb{E} \left[ \sum_{t=1}^T \|\tilde{U}_t - \tilde{U}_{t-1}\|_{abs} \right] + 2\alpha^2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] \\ & \leq \alpha^2 \left( \frac{\beta_1}{1-\beta_1} \right)^2 \frac{G_\infty^2}{\sqrt{N}} \frac{1}{\epsilon^2} \frac{1}{1-\lambda} \mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right] + 2\alpha^2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right], \end{aligned}$$

507 where the last inequality is due to (26).

508 We now bound the last term on RHS of the above inequality. A trivial bound can be

$$\sum_{t=1}^T \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \leq \sum_{t=1}^T d G_\infty^2 \frac{1}{\epsilon},$$

509 due to  $\|g_{t,i}\| \leq G_\infty$  and  $[u_{t,i}]_j \geq \epsilon, \forall j$  (this is easy to verify from update rule of  $u_{t,i}$  and the  
 510 assumption that  $[v_{t,i}]_j \geq \epsilon, \forall i$ ). However, the above bound is independent of  $N$ , to get a better bound,  
 511 we need a more involved analysis to show its dependency on  $N$ . To do this, we first notice that

$$\begin{aligned} & \mathbb{E}_{G_t|G_{1:t-1}} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] \\ & = \mathbb{E}_{G_t|G_{1:t-1}} \left[ \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \left\langle \frac{\nabla f_i(x_{t,i}) + \xi_{t,i}}{\sqrt{u_{t,i}}}, \frac{\nabla f_j(x_{t,j}) + \xi_{t,j}}{\sqrt{u_{t,j}}} \right\rangle \right] \\ & \stackrel{(a)}{=} \mathbb{E}_{G_t|G_{1:t-1}} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 \right] + \mathbb{E}_{G_t|G_{1:t-1}} \left[ \frac{1}{N^2} \sum_{i=1}^N \left\| \frac{\xi_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] \\ & \stackrel{(b)}{=} \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 + \frac{1}{N^2} \sum_{i=1}^N \sum_{l=1}^d \frac{\mathbb{E}_{G_t|G_{1:t-1}}[[\xi_{t,i}]_l^2]}{[u_{t,i}]_l} \\ & \stackrel{(c)}{\leq} \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 + \frac{d}{N} \frac{\sigma^2}{\epsilon}, \end{aligned}$$

512 where (a) is due to  $\mathbb{E}_{G_t|G_{1:t-1}}[\xi_{t,i}] = 0$  and  $\xi_{t,i}$  is independent of  $x_{t,j}, \forall j, u_{t,j}, \forall j$ , and  $\xi_j, \forall j \neq i$ ,  
 513 (b) comes from the fact that  $x_{t,i}, u_{t,i}$  are fixed given  $G_{1:t}$ , (c) is due to  $\mathbb{E}_{G_t|G_{1:t-1}}[[\xi_{t,i}]_l^2] \leq \sigma^2$  and  
 514  $[u_{t,i}]_l \geq \epsilon$  by definition.

515 Then we have

$$\begin{aligned} \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] & = \mathbb{E}_{G_{1:t-1}} \left[ \mathbb{E}_{G_t|G_{1:t-1}} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] \right] \\ & \leq \mathbb{E}_{G_{1:t-1}} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 + \frac{d}{N} \frac{\sigma^2}{\epsilon} \right] \\ & = \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 \right] + \frac{d}{N} \frac{\sigma^2}{\epsilon}. \end{aligned} \quad (29)$$

516 In traditional analysis of SGD-like distributed algorithms, the term corresponding to  
 517  $\mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 \right]$  will be merged with the first order descent when the stepsize is cho-  
 518 sen to be small enough. However, in our case, the term cannot be merged because it is different from  
 519 the first order descent in our algorithm. A brute-force upper bound is possible but this will lead to a  
 520 worse convergence rate in terms of  $N$ . Thus, we need a more detailed analysis for the term in the  
 521 following.

$$\begin{aligned}
 & \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 \right] \\
 &= \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} + \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right) \right\|^2 \right] \\
 &\leq 2\mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 2\mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right) \right\|^2 \right] \\
 &\leq 2\mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 2\mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N \left\| \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right) \right\|^2 \right] \\
 &\leq 2\mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 2\mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N G_\infty^2 \frac{1}{\sqrt{\epsilon}} \left\| \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right\|_1 \right].
 \end{aligned}$$

522 Summing over  $T$ , we have

$$\begin{aligned}
 & \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 \right] \\
 &\leq 2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 2 \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N G_\infty^2 \frac{1}{\sqrt{\epsilon}} \left\| \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right\|_1 \right]. \quad (30)
 \end{aligned}$$

523 For the last term on RHS of (30), we can bound it similarly as what we did for  $D_2$  from (23) to (24),  
 524 which yields

$$\begin{aligned}
 & \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N G_\infty^2 \frac{1}{\sqrt{\epsilon}} \left\| \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right\|_1 \right] \leq \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N G_\infty^2 \frac{1}{\sqrt{\epsilon}} \frac{1}{2\epsilon^{1.5}} \|u_{t,i} - \bar{U}_t\|_1 \right] \\
 &= \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} G_\infty^2 \frac{1}{2\epsilon^2} \|\bar{U}_t \mathbf{1}^T - U_t\|_{abs} \right] \\
 &\leq \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} G_\infty^2 \frac{1}{2\epsilon^2} \left\| - \sum_{l=2}^N \tilde{U}_t q_l q_l^T \right\|_{abs} \right] \\
 &\leq \frac{1}{\sqrt{N}} G_\infty^2 \frac{1}{2\epsilon^2} \mathbb{E} \left[ \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \right]. \quad (31)
 \end{aligned}$$

525 Further, we have

$$\begin{aligned}
& \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] \\
& \leq 2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(\bar{X}_t)}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(\bar{X}_t) - \nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] \\
& = 2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(\bar{X}_t) - \nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right]
\end{aligned}$$

526 and the last term on RHS of the above inequality can be bounded following similar procedures from  
527 (16) to (21), as what we did for  $D_3$ . Completing the procedures yields

$$\begin{aligned}
& \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(\bar{X}_t) - \nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] \\
& \leq \sum_{t=1}^T \mathbb{E} \left[ L \frac{1}{\epsilon} \frac{1}{N} \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 \right] \\
& \leq \sum_{t=1}^T \mathbb{E} \left[ L \frac{1}{\epsilon} \frac{1}{N} \alpha^2 \left( \frac{1}{1-\lambda} \right) N d G_\infty^2 \frac{1}{\epsilon} \right] \\
& = T L \frac{1}{\epsilon^2} \alpha^2 \left( \frac{1}{1-\lambda} \right) d G_\infty^2.
\end{aligned} \tag{32}$$

528 Finally, combining (29) to (32), we get

$$\begin{aligned}
& \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] \\
& \leq 4 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 4 T L \frac{1}{\epsilon^2} \alpha^2 \left( \frac{1}{1-\lambda} \right) d G_\infty^2 \\
& \quad + 2 \frac{1}{\sqrt{N}} G_\infty^2 \frac{1}{2\epsilon^2} \mathbb{E} \left[ \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \right] + T \frac{d}{N} \frac{\sigma^2}{\epsilon} \\
& \leq 4 \frac{1}{\sqrt{\epsilon}} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] + 4 T L \frac{1}{\epsilon^2} \alpha^2 \left( \frac{1}{1-\lambda} \right) d G_\infty^2 \\
& \quad + 2 \frac{1}{\sqrt{N}} G_\infty^2 \frac{1}{2\epsilon^2} \mathbb{E} \left[ \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \right] + T \frac{d}{N} \frac{\sigma^2}{\epsilon}.
\end{aligned}$$

529 where the last inequality is due to each element of  $\bar{U}_t$  is lower bounded by  $\epsilon$  by definition.

530 Combining all above, we can have

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \\
& \leq \frac{2}{T\alpha} (\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) \\
& \quad + \frac{L}{T} \alpha \left( \frac{\beta_1}{1-\beta_1} \right)^2 \frac{G_\infty^2}{\sqrt{N}} \frac{1}{\epsilon^2} \frac{1}{1-\lambda} \mathbb{E}[\mathcal{V}_T] \\
& \quad + \frac{8L}{T} \alpha \frac{1}{\sqrt{\epsilon}} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] + 8L^2 \alpha \frac{1}{\epsilon^2} \alpha^2 \left( \frac{1}{1-\lambda} \right) d G_\infty^2 \\
& \quad + \frac{4L}{T} \alpha \frac{1}{\sqrt{N}} G_\infty^2 \frac{1}{2\epsilon^2} \mathbb{E} \left[ \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \right] + 2L\alpha \frac{d}{N} \frac{\sigma^2}{\epsilon} \\
& \quad + \frac{2}{T} \frac{\beta_1}{1-\beta_1} G_\infty^2 \frac{1}{2\epsilon^{1.5}} \frac{1}{\sqrt{N}} \mathbb{E} \left[ \frac{1}{1-\lambda} \mathcal{V}_T \right] \\
& \quad + \frac{2}{T} \frac{G_\infty^2}{\sqrt{N}} \frac{1}{2\epsilon^{1.5}} \frac{\lambda}{1-\lambda} \mathbb{E}[\mathcal{V}_T] \\
& \quad + \frac{3}{T} \left( \sum_{t=1}^T L \left( \frac{1}{1-\lambda} \right)^2 \alpha^2 d G_\infty^2 \frac{1}{\epsilon^{1.5}} + \sum_{t=1}^T L \left( \frac{\beta_1}{1-\beta_1} \right)^2 \alpha^2 d \frac{G_\infty^2}{\epsilon^{1.5}} \right) \\
& = \frac{2}{T\alpha} (\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + 2L\alpha \frac{d}{N} \frac{\sigma^2}{\epsilon} + 8L\alpha \frac{1}{\sqrt{\epsilon}} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \\
& \quad + 3\alpha^2 d \left( \left( \frac{\beta_1}{1-\beta_1} \right)^2 + \left( \frac{1}{1-\lambda} \right)^2 \right) L \frac{G_\infty^2}{\epsilon^{1.5}} + 8\alpha^3 L^2 \left( \frac{1}{1-\lambda} \right) d \frac{G_\infty^2}{\epsilon^2} \\
& \quad + \frac{1}{T\epsilon^{1.5}} \frac{G_\infty^2}{\sqrt{N}} \frac{1}{1-\lambda} \left( L\alpha \left( \frac{\beta_1}{1-\beta_1} \right)^2 \frac{1}{\epsilon^{0.5}} + \lambda + \frac{\beta_1}{1-\beta_1} + 2L\alpha \frac{1}{\epsilon^{0.5}} \lambda \right) \mathbb{E}[\mathcal{V}_T]. \tag{33}
\end{aligned}$$

531 where  $\mathcal{V}_T := \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}$ . Set  $\alpha = \frac{1}{\sqrt{dT}}$  and when  $\alpha \leq \frac{\epsilon^{0.5}}{16L}$ , we further have

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \\
& \leq \frac{4}{T\alpha} (\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + 4L\alpha \frac{d}{N} \frac{\sigma^2}{\epsilon} \\
& \quad + 6\alpha^2 d \left( \left( \frac{\beta_1}{1-\beta_1} \right)^2 + \left( \frac{1}{1-\lambda} \right)^2 \right) L \frac{G_\infty^2}{\epsilon^{1.5}} + 16\alpha^3 L^2 \left( \frac{1}{1-\lambda} \right) d \frac{G_\infty^2}{\epsilon^2} \\
& \quad + \frac{2}{T\epsilon^{1.5}} \frac{G_\infty^2}{\sqrt{N}} \frac{1}{1-\lambda} \left( L\alpha \left( \frac{\beta_1}{1-\beta_1} \right)^2 \frac{1}{\epsilon^{0.5}} + \lambda + \frac{\beta_1}{1-\beta_1} + 2L\alpha \frac{1}{\epsilon^{0.5}} \lambda \right) \mathbb{E}[\mathcal{V}_T] \\
& = \frac{4\sqrt{d}}{\sqrt{T}} (\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + 4L \frac{\sqrt{d}}{\sqrt{T}} \frac{1}{N} \frac{\sigma^2}{\epsilon} \\
& \quad + 6 \frac{1}{T} \left( \left( \frac{\beta_1}{1-\beta_1} \right)^2 + \left( \frac{1}{1-\lambda} \right)^2 \right) L \frac{G_\infty^2}{\epsilon^{1.5}} + 16 \frac{1}{T^{1.5} d^{0.5}} L^2 \left( \frac{1}{1-\lambda} \right) \frac{G_\infty^2}{\epsilon^2} \\
& \quad + \frac{2}{T\epsilon^{1.5}} \frac{G_\infty^2}{\sqrt{N}} \frac{1}{1-\lambda} \left( \frac{L}{\sqrt{Td}} \left( \frac{\beta_1}{1-\beta_1} \right)^2 \frac{1}{\epsilon^{0.5}} + \lambda + \frac{\beta_1}{1-\beta_1} + 2 \frac{L}{\sqrt{Td}} \frac{1}{\epsilon^{0.5}} \lambda \right) \mathbb{E}[\mathcal{V}_T]
\end{aligned}$$



$$\leq C_1 \frac{\sqrt{d}}{\sqrt{T}} \left( \mathbb{E}[f(Z_1)] - \min_z f(z) + \frac{\sigma^2}{N} \right) + \frac{1}{T} C_2 + \frac{1}{T^{1.5} d^{0.5}} C_3 \\ + \left( \frac{1}{TN^{0.5}} C_4 + \frac{1}{T^{1.5} d^{0.5} N^{0.5}} C_5 \right) \mathbb{E}[\mathcal{V}_T],$$

532 where the first inequality is obtained by moving the term  $8L\alpha \frac{1}{\sqrt{\epsilon}} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right]$  on the  
 533 RHS of (33) to the LHS to cancel it using the assumption  $8L\alpha \frac{1}{\sqrt{\epsilon}} \leq \frac{1}{2}$  followed by multiplying both  
 534 sides by 2, and the constants introduced in the last step are defined as following

$$C_1 = \max(4, 4L/\epsilon), \\ C_2 = 6 \left( \left( \frac{\beta_1}{1-\beta_1} \right)^2 + \left( \frac{1}{1-\lambda} \right)^2 \right) L \frac{G_\infty^2}{\epsilon^{1.5}}, \\ C_3 = 16L^2 \left( \frac{1}{1-\lambda} \right) \frac{G_\infty^2}{\epsilon^2}, \\ C_4 = \frac{2}{\epsilon^{1.5}} \frac{1}{1-\lambda} \left( \lambda + \frac{\beta_1}{1-\beta_1} \right) G_\infty^2, \\ C_5 = \frac{2}{\epsilon^2} \frac{1}{1-\lambda} L \left( \frac{\beta_1}{1-\beta_1} \right)^2 G_\infty^2 + \frac{4}{\epsilon^2} \frac{\lambda}{1-\lambda} L G_\infty^2.$$

535 Substituting into  $Z_1 = \bar{X}_1$  completes the proof.  $\square$

### 536 C Proof of Theorem 3

537 By Theorem 2, we know under the assumptions of the theorem, we have

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \leq C_1 \frac{\sqrt{d}}{\sqrt{T}} \left( \mathbb{E}[f(\bar{X}_1)] - \min_z f(z) + \frac{\sigma^2}{N} \right) + \frac{1}{T} C_2 + \frac{1}{T^{1.5} d^{0.5}} C_3 \\ + \left( \frac{1}{TN^{0.5}} C_4 + \frac{1}{T^{1.5} d^{0.5} N^{0.5}} C_5 \right) \mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right], \quad (34)$$

538 where  $\|\cdot\|_{abs}$  denotes the entry-wise  $L_1$  norm of a matrix (i.e.  $\|A\|_{abs} = \sum_{i,j} |A_{ij}|$ ) and  
 539  $C_1, C_2, C_3, C_4, C_5$  are defined in Theorem 2.

540 Since Algorithm 3 is a special case of 2, building on result of Theorem 2, we just need to characterize  
 541 the growth speed of  $\mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right]$  to prove convergence of Algorithm 3. By the  
 542 update rule of Algorithm 3, we know  $\hat{V}_t$  is non decreasing and thus

$$\mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right] = \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^d |-\hat{v}_{t-2,i,j} + \hat{v}_{t-1,i,j}| \right] \\ = \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^d (-\hat{v}_{t-2,i,j} + \hat{v}_{t-1,i,j}) \right] \\ = \mathbb{E} \left[ \sum_{i=1}^N \sum_{j=1}^d (-\hat{v}_{-1,i,j} + \hat{v}_{T-1,i,j}) \right] \\ = \mathbb{E} \left[ \sum_{i=1}^N \sum_{j=1}^d (-\hat{v}_{0,i,j} + \hat{v}_{T-1,i,j}) \right],$$

543 where the last equality is because we defined  $\hat{V}_{-1} \triangleq \hat{V}_0$  previously.

544 Further, because  $\|g_{t,i}\|_\infty \leq G_\infty, \forall t, i$  and  $v_{t,i}$  is a exponential moving average of  $g_{k,i}, k =$   
545  $1, 2, \dots, t$ , we know  $|\hat{v}_{t,i}| \leq G_\infty, \forall t, i, j$ . In addition, by update rule of  $\hat{V}_t$ , we also know  
546 each element of  $\hat{V}_t$  also cannot be greater than  $G_\infty^2$ , i.e.  $|\hat{v}_{t,i}| \leq G_\infty^2, \forall t, i, j$ . Given the fact that  
547  $[\hat{v}_{0,i}]_j \geq 0$ , we have

$$\mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right] = \mathbb{E} \left[ \sum_{i=1}^N \sum_{j=1}^d (-[\hat{v}_{0,i}]_j + [\hat{v}_{T-1,i}]_j) \right] \leq \mathbb{E} \left[ \sum_{i=1}^N \sum_{j=1}^d G_\infty^2 \right] = NdG_\infty^2.$$

548 Substituting the above into (34), we have

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] &\leq C_1 \frac{\sqrt{d}}{\sqrt{T}} \left( \mathbb{E}[f(\bar{X}_1)] - \min_z f(z) + \frac{\sigma^2}{N} \right) + \frac{1}{T} C_2 + \frac{1}{T^{1.5} d^{0.5}} C_3 \\ &\quad + \frac{d}{T} C_4 \sqrt{N} G_\infty^2 + \frac{\sqrt{d}}{T^{1.5}} C_5 \sqrt{N} G_\infty^2 \\ &= C'_1 \frac{\sqrt{d}}{\sqrt{T}} \left( \mathbb{E}[f(\bar{X}_1)] - \min_z f(z) + \frac{\sigma^2}{N} \right) + \frac{1}{T} C'_2 + \frac{1}{T^{1.5} d^{0.5}} C'_3 \\ &\quad + \frac{d}{T} \sqrt{N} C'_4 + \frac{\sqrt{d}}{T^{1.5}} \sqrt{N} C'_5, \end{aligned} \quad (35)$$

549 where we have

$$C'_1 = C_1 \quad C'_2 = C_2 \quad C'_3 = C_3 \quad C'_4 = C_4 G_\infty^2 \quad C'_5 = C_5 G_\infty^2. \quad (36)$$

550 and concluding our proof.  $\square$

## 551 D Additional Experiments and Details

552 In this section, we compare the learning curves of different algorithms with different stepsizes on  
553 heterogeneous data distribution. We use 5 nodes and the heterogeneous data distribution is created  
554 by assigning each node with data of only two labels and there are no overlapping labels between  
555 different nodes. For all algorithms, we compare stepsizes in the set  $[1e-1, 1e-2, 1e-3, 1e-4, 1e-5,$   
556  $1e-6]$ .

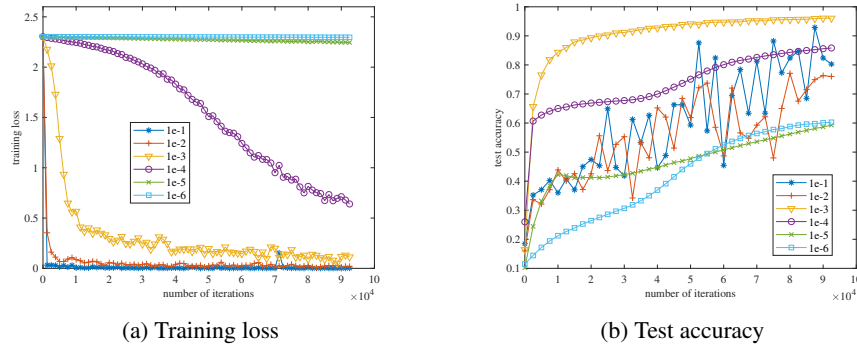


Figure 2: Performance comparison of different stepsizes for DGD

557 Figure 2 shows the training loss and test accuracy of DGD, it can be seen that the stepsize 1e-3 works  
558 best for DGD in terms of test accuracy and 1e-1 works best in terms of training loss. The difference  
559 is caused by the inconsistency among the value of parameters on different nodes when the stepsize  
560 is large. The training loss is calculated as the average of the loss value of different local models  
561 evaluated on their local training batch. Thus, though the training loss is small evaluated at a particular

node, the test accuracy will be low when evaluating data with labels not seen by the node (recall that each node contains data with different labels).

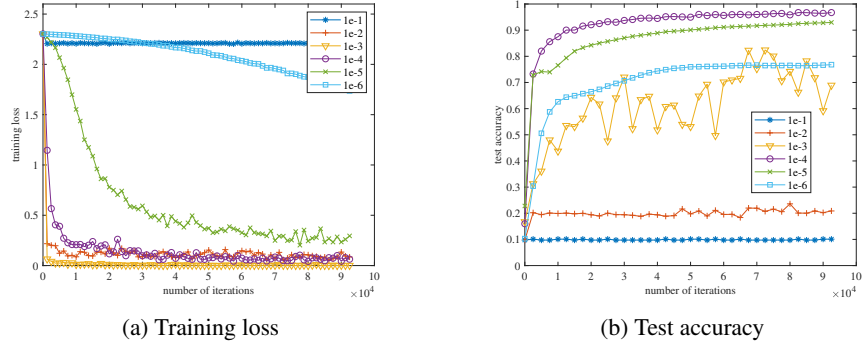


Figure 3: Performance comparison of different stepsizes for decentralized AMSGrad

Figure 3 shows the performance of decentralized AMSGrad with different stepsizes, we can see its best performance is better than DGD and the performance is stabler (the test performance is less sensitive to stepsize choice).

Figure 4 shows the performance of DADM, as it can be expected, the performance of DADAM is not as good as DGD and decentralized AMSGrad since it is not a convergent algorithm and the heterogeneity in data amplified the non-convergence issue of DADAM.

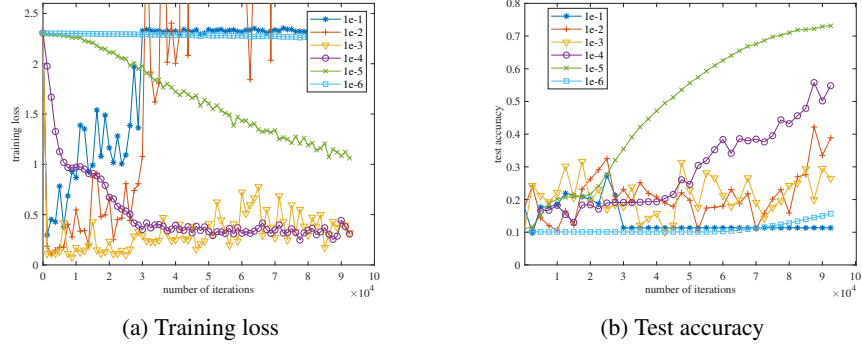


Figure 4: Performance comparison of different stepsizes for DADAM

From the experiments above, we can see the advantages of decentralized AMSGrad in terms of both performance and ease of parameter tuning, and the importance of ensuring the theoretical convergence of algorithms.