

---

# On the Convergence of Decentralized Adaptive Gradient Methods

---

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. Meanwhile, given the need for distributed computing, distributed optimization algorithms are rapidly becoming a focal point. 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 this paper, we introduce novel convergent decentralized adaptive gradient methods and rigorously incorporate adaptive gradient methods into decentralized training procedures. 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. We illustrate the benefit of our generic decentralized framework on a prototype method, *i.e.* AMSGrad, both theoretically and numerically.

## 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 [10; 25]. As a result, a large amount of recent works has been focussing on reducing communication cost for distributed learning [3; 22; 36; 32; 35; 34]. 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 [2] and quantization [9; 14; 16] 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* [13], where the parameter server is removed and each node only communicates with its neighbors, is drawing attention. It has been shown in [21] 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 [12], Adam [17] or AMSGrad [29]. 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) [30]. 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. [28] 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 applied

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 required in [28]. Designing adaptive methods in such settings is highly non-trivial due to the already complex 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:

- We investigate the use of 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 and highlight the importance of adaptive learning rate consensus.
- 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.

The paper is organized as follows. In Section 2, we show the importance of adaptive learning rate consensus by proving a divergent example for a recently proposed decentralized adaptive gradient method, DADAM [26]. In Section 3, we develop our general framework for converting adaptive gradient methods into their decentralized counterparts along with convergence analysis and converted algorithms. Illustrative experiments are presented in Section 4. Section 5 concludes our work.

**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 matrix  $[g_{t,1}, g_{t,2}, \dots, g_{t,N}]$  (where  $g_{t,i}$  is a column vector),  $M_t := [m_{t,N}]$ ,  $X_t := [x_{t,N}]$ ,  $\overline{\nabla}f(X_t) := \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i})$ ,  $U_t := [u_{t,N}]$ ,  $\tilde{U}_t := [\tilde{u}_{t,N}]$ ,  $V_t := [v_{t,N}]$ ,  $\hat{V}_t := [\hat{v}_{t,N}]$ ,  $\bar{X}_t := \frac{1}{N} \sum_{i=1}^N x_{t,i}$ ,  $\bar{U}_t := \frac{1}{N} \sum_{i=1}^N u_{t,i}$  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-known algorithms such as ADMM [7], Dual Averaging [13], Distributed Subgradient Descent [27]. More recent algorithms include Extra [31], Next [11], Prox-PDA [15], GNSD [23], and Choco-SGD [18]. 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 also be useful for training deep models such as neural networks. Lian et al. [21] demonstrate that a stochastic version of Decentralized Subgradient Descent can outperform parameter server-based algorithms when the communication cost is high. Tang et al. [33] propose the D<sup>2</sup> algorithm improving the convergence rate over Stochastic Subgradient Descent. Assran et al. [4] 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 [26] proposes a decentralized version of AMSGrad [29] and it 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 [12] or Adam [17] 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 [29] where the authors point out an error in previous convergence analyses. Since then, a variety of papers have been focusing on analyzing the convergence behavior of the numerous existing adaptive gradient methods. Ward et al. [37], Li and Orabona [20] derive convergence guarantees for a variant of AdaGrad without coordinate-wise learning rates. Chen et al. [8] analyze the convergence behavior of a broad class of algorithms including AMSGrad and

92 AdaGrad. Zhou et al. [41] give a more refined analysis of AMSGrad with better convergence rate.  
 93 Zou and Shen [42] provide a unified convergence analysis for AdaGrad with momentum. Noticeable  
 94 recent works on adaptive gradient methods can be found in [1; 24; 40].

## 95 2.2 Decentralized Optimization

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

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

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

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

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

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

107 Assumptions A1 and A3 are standard in distributed optimization literature. A2 is slightly stronger  
 108 than the traditional assumption that the estimator has bounded variance, but is commonly used for the  
 109 analysis of adaptive gradient methods [8; 37]. Note that the bounded gradient estimator assumption  
 110 in A2 implies the bounded variance assumption in A3. In decentralized optimization, the nodes are  
 111 connected as a graph and each node only communicates to its neighbors. In such case, one usually  
 112 constructs a  $N \times N$  matrix  $W$  for information sharing when designing new algorithms. We denote  
 113  $\lambda_i$  to be its  $i$ th largest eigenvalue and define  $\lambda \triangleq \max(|\lambda_2|, |\lambda_N|)$ . The matrix  $W$  cannot be arbitrary,  
 114 its required key properties are listed in the following assumption:

115 **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)  $\lambda_1 = 1$ ,  $|\lambda_2| < 1$ ,  
 116  $|\lambda_N| < 1$  and (III)  $W_{i,j} = 0$  if node  $i$  and node  $j$  are not neighbors.

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

## 119 2.3 Divergence of DADAM

120 Recently, Nazari et al. [26] initiated an attempt  
 121 to bring adaptive gradient methods into decen-  
 122 tralized optimization with Decentralized ADAM  
 123 (DADAM), shown in Algorithm 1. DADAM is  
 124 essentially a decentralized version of ADAM  
 125 and the key modification is the use of a con-  
 126 sensus step on the optimization variable  $x$  to  
 127 transmit information across the network, encour-  
 128 aging its convergence. The matrix  $W$  is a dou-  
 129 bly stochastic matrix (which satisfies A4) for  
 130 achieving average consensus of  $x$ . Introducing  
 131 such mixing matrix is standard for decentraliz-  
 132 ing an algorithm, such as distributed gradient de-  
 133 scent [27; 39]. It is proven in [26] that DADAM  
 134 admits a non-standard regret bound in the online  
 135 setting. Nevertheless, whether the algorithm can  
 136 converge to stationary points in standard offline settings such training neural networks is still unknown.  
 137 The next theorem shows that DADAM may fail to converge in the offline settings.

138 **Theorem 1.** There exists a problem satisfying A1-A4 where DADAM fails to converge to a stationary  
 139 points with  $\nabla f(\bar{X}_t) = 0$ .

---

### 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:      $\hat{v}_{t,i} = \beta_3 \hat{v}_{t,i} + (1 - \beta_3) \max(\hat{v}_{t-1,i}, v_{t,i})$ 
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

```

---

140 *Proof.* Consider a two-node setting with objective function  $f(x) = 1/2 \sum_{i=1}^2 f_i(x)$  and  $f_1(x) =$   
141  $\mathbb{1}[|x| \leq 1]2x^2 + \mathbb{1}[|x| > 1](4|x| - 2)$ ,  $f_2(x) = \mathbb{1}[|x-1| \leq 1](x-1)^2 + \mathbb{1}[|x-1| > 1](2|x-1| - 1)$ .  
142 We set the mixing matrix  $W = [0.5, 0.5; 0.5, 0.5]$ . The optimal solution is  $x^* = 1/3$ . Both  $f_1$  and  
143  $f_2$  are smooth and convex with bounded gradient norm 4 and 2, respectively. We also have  $L = 4$   
144 (defined in A1). If we initialize with  $x_{1,1} = x_{1,2} = -1$  and run DADAM with  $\beta_1 = \beta_2 = \beta_3 = 0$   
145 and  $\epsilon \leq 1$ , we will get  $\hat{v}_{1,1} = 16$  and  $\hat{v}_{1,2} = 4$ . Since  $|g_{t,1}| \leq 4$ ,  $|g_{t,2}| \leq 2$  due to bounded gradient,  
146 and  $(\hat{v}_{t,1}, \hat{v}_{t,2})$  are non-decreasing, we have  $\hat{v}_{t,1} = 16$ ,  $\hat{v}_{t,2} = 4$ ,  $\forall t \geq 1$ . Thus, after  $t = 1$ , DADAM  
147 is equivalent to running decentralized gradient descent (D-PSGD) [39] with a re-scaled  $f_1$  and  $f_2$ , i.e.  
148 running D-PSGD on  $f'(x) = \sum_{i=1}^2 f'_i(x)$  with  $f'_1(x) = 0.25f_1(x)$  and  $f'_2(x) = 0.5f_2(x)$ , which  
149 unique optimal  $x' = 0.5$ . Define  $\bar{x}_t = (x_{t,1} + x_{t,2})/2$ , then by Theorem 2 in [39], we have when  
150  $\alpha < 1/4$ ,  $f'(\bar{x}_t) - f(x') = O(1/(\alpha t))$ . Since  $f'$  has a unique optima  $x'$ , the above bound implies  
151  $\bar{x}_t$  is converging to  $x' = 0.5$  which has non-zero gradient on function  $\nabla f(0.5) = 0.5$ .  $\square$

152 Theorem 1 shows that, even though DADAM is proven to satisfy some regret bounds [26], it can  
153 fail to converge to stationary points in the nonconvex offline setting (common for training neural  
154 networks). We conjecture that this inconsistency in the convergence behavior of DADAM is due to  
155 the definition of the regret in [26]. The next section presents decentralized adaptive gradient methods  
156 that are guaranteed to converge to stationary points under assumptions and provide a characterization  
157 of that convergence in finite-time and independently of the initialization.

### 158 3 On the Convergence of Decentralized Adaptive Gradient Methods

159 In this section, we discuss the difficulties of designing adaptive gradient methods in decentralized  
160 optimization and introduce an algorithmic framework that can turn existing convergent adaptive gra-  
161 dient methods to their decentralized counterparts. We also develop the first convergent decentralized  
162 adaptive gradient method, converted from AMSGrad, as an instance of this framework.

#### 163 3.1 Importance and Difficulties of Consensus on Adaptive Learning Rates

164 The divergent example in the previous section implies that we should synchronize the adaptive  
165 learning rates on different nodes. This can be easily achieved in the parameter server setting where  
166 all the nodes are sending their gradients to a central server at each iteration. The parameter server  
167 can then exploit the received gradients to maintain a sequence of synchronized adaptive learning  
168 rates when updating the parameters, see [28]. However, in our decentralized setting, every node can  
169 only communicate with its neighbors and such central server does not exist. Under that setting, the  
170 information for updating the adaptive learning rates can only be shared locally instead of broadcasted  
171 over the whole network. This makes it impossible to obtain, in a single iteration, a synchronized  
172 adaptive learning rate update using all the information in the network.

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

178 *Algorithmic Approach:* Our contributions being on an algorithmic level, another way to solve the  
179 aforementioned problem is by letting the sequences of adaptive learning rates, present on different  
180 nodes, to gradually *consent*, through the iterations. Intuitively, if the adaptive learning rates can  
181 consent fast enough, the difference among the adaptive learning rates on different nodes will not  
182 affect the convergence behavior of the algorithm. Consequently, no extra communication costs need  
183 to be introduced. We now develop this exact idea within the existing adaptive methods stressing on  
184 the need for a relatively low-cost and easy-to-implement consensus of adaptive learning rates.

185 Below is main archetype of the adaptive rates consensus mechanism within a decentralized framework.

#### 186 3.2 Unifying Decentralized Adaptive Gradient Framework

187 While each node can have different  $\hat{v}_{t,i}$  in DADAM (Algorithm 1), one can keep track of the  
188 min/max/average of these adaptive learning rates and use that quantity as the new adaptive learning  
189 rate. The predefinition of some convergent lower and upper bounds may also lead to a gradual  
190 synchronization of the adaptive learning rates on different nodes as developed for AdaBound in [24].

In this paper, we present an algorithm framework for decentralized adaptive gradient methods as Algorithm 2, which uses average consensus of  $\hat{v}_{t,i}$  (see consensus update in line 8 and 11) to help convergence. Algorithm 2 can become different adaptive gradient methods by specifying  $r_t$  as different functions. E.g., when we choose  $\hat{v}_{t,i} = \frac{1}{t} \sum_{k=1}^t g_{k,i}^2$ , Algorithm 2 becomes a decentralized version of AdaGrad. When one chooses  $\hat{v}_{t,i}$  to be the adaptive learning rate for AMSGrad, we get decentralized AMSGrad (Algorithm 3). The intuition of using average consensus is that 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). The following result gives a finite-time convergence rate for our framework described in Algorithm 2.

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

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] &\leq C_1 \left( \frac{1}{T\alpha} (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \alpha \frac{d\sigma^2}{N} \right) + C_2 \alpha^2 d \\ &\quad + C_3 \alpha^3 d + \frac{1}{T\sqrt{N}} (C_4 + C_5 \alpha) \mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right] \end{aligned} \quad (2)$$

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.

In addition, one can specify  $\alpha$  to show convergence in terms of  $T$ ,  $d$ , and  $N$ . An immediate result, shown in Corollary 2.1, is by setting  $\alpha = \sqrt{N}/\sqrt{Td}$ :

**Corollary 2.1.** Assume A1-A4. Set  $\alpha = \sqrt{N}/\sqrt{Td}$ . When  $\alpha \leq \frac{\epsilon^{0.5}}{16L}$ , Algorithm 2 yields:

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] &\leq C_1 \frac{\sqrt{d}}{\sqrt{TN}} \left( (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \sigma^2 \right) + C_2 \frac{N}{T} \\ &\quad + C_3 \frac{N^{1.5}}{T^{1.5}d^{0.5}} + \left( C_4 \frac{1}{T\sqrt{N}} + C_5 \frac{1}{T^{1.5}d^{0.5}} \right) \mathbb{E}[\mathcal{V}_T] \end{aligned} \quad (3)$$

where  $\mathcal{V}_T := \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}$  and  $C_1, C_2, C_3, C_4, C_5$  are defined in Theorem 2.

Corollary 2.1 indicates that if  $\mathbb{E}[\mathcal{V}_T] = o(T)$  and  $\bar{U}_t$  is upper bounded, then Algorithm 2 is guaranteed to converge to stationary points of the loss function. Intuitively, this means that if the adaptive learning rates on different nodes do not change too fast, the algorithm can converge. In convergence analysis, the term  $\mathbb{E}[\mathcal{V}_T]$  upper bounds the total bias in update direction caused by the correlation between  $m_{t,i}$  and  $\hat{v}_{t,i}$ . It is shown in [8] that when  $N = 1$ ,  $\mathbb{E}[\mathcal{V}_T] = \tilde{O}(d)$  for AdaGrad and AMSGrad. Besides,  $\mathbb{E}[\mathcal{V}_T] = \tilde{O}(Td)$  for Adam which do not converge. Later, we will show convergence of decentralized versions of AMSGrad and AdaGrad by bounding this term as  $O(Nd)$  and  $O(Nd \log(T))$ , respectively. Corollary 2.1 also conveys the benefits of using more nodes in the graph employed. When  $T$  is large enough such that the term  $O(\sqrt{d}/\sqrt{TN})$  dominates the right hand side of (3), then linear speedup can be achieved by increasing the number of nodes  $N$ .



Another point worth discussion is the choice of  $W$  since the convergence rate depends on  $\lambda$  which is dependent on  $W$ . A common way to set  $W$  for undirected graph is the maximum-degree method (MDM) in [5]. Denote  $d_i$  as degree of vertex  $i$  and  $d_{\max} = \max_i d_i$ , MDM sets  $W_{i,i} = 1 - d_i/d_{\max}$ ,  $W_{i,j} = 1/d_{\max}$  if  $i \neq j$  and  $(i, j)$  is an edge, and  $W_{i,j} = 0$  otherwise. This  $W$  ensures Assumption A4 for many common connected graph types, so does the variant  $\gamma I + (1 - \gamma)W$  for any  $\gamma \in [0, 1)$ . A more refined choice of  $W$  coupled with a comprehensive discussion on  $\lambda$  in our Theorem 2 can be found in [6], e.g.,  $1 - \lambda = O(1/N^2)$  for cycle graphs,  $1 - \lambda = O(1/\log(N))$  for hypercube graphs,  $\lambda = 0$  for fully connected graph. Intuitively,  $\lambda$  can be close to 1 for sparse graphs and to 0 for dense graphs. This is consistent (2), whose RHS is large for  $\lambda$  close to 1 and small for  $\lambda$  close to 0 since average consensus on sparser graphs is expected to take longer time.

### 3.3 Application to AMSGrad algorithm

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, the  $u_{t,i}$  will converge to the same number  $\lim_{t \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \hat{v}_{t,i}$  if the limit exists.

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. The consensus of adaptive learning rates is the key difference between decentralized AMSGrad and DADAM and is the reason why decentralized AMSGrad is convergent while DADAM is not.

One may notice that decentralized AMSGrad does not reduce to AMSGrad for  $N = 1$  since the quantity  $u_{t,i}$  in line 10 is calculated based on  $v_{t-1,i}$  instead of  $v_{t,i}$ . This design encourages the execution of gradient computation and communication in a parallel manner. Specifically, line 4-7 (line 4-6) in Algorithm 3 (Algorithm 2) can be executed in parallel with line 8-9 (line 7-8) 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 getting 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  $\alpha \leq \frac{\epsilon^{0.5}}{16L}$ , then Algorithm 3 satisfies:

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \leq C'_1 \frac{\sqrt{d}}{\sqrt{TN}} (D_f + \sigma^2) + C'_2 \frac{N}{T} + C'_3 \frac{N^{1.5}}{T^{1.5}d^{0.5}} + C'_4 \frac{\sqrt{Nd}}{T} + C'_5 \frac{Nd^{0.5}}{T^{1.5}},$$

where  $D_f := \mathbb{E}[f(Z_1)] - \min_x f(x)$ ,  $C'_1 = C_1$ ,  $C'_2 = C_2$ ,  $C'_3 = C_3$ ,  $C'_4 = C_4 G_\infty^2$  and  $C'_5 = C_5 G_\infty^2$ .  $C_1, C_2, C_3, C_4, C_5$  are independent of  $d, T$  and  $N$  defined in Theorem 2. In addition, the consensus of variables at different nodes is given by  $\frac{1}{N} \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 \leq \frac{N}{T} \left( \frac{1}{1-\lambda} \right)^2 G_\infty^2 \frac{1}{\epsilon}$ .

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. Another remark is the convergence measure can be converted to  $\frac{1}{T} \sum_{t=1}^T \mathbb{E} [\|\nabla f(\bar{X}_t)\|^2]$  using the fact that  $\|\bar{U}_t\|_\infty \leq G_\infty^2$  (by update rule of Algorithm 3), for the ease of comparison with existing literature.

**Proof Sketch of Theorem 2:** The detailed proofs are reported in the supplementary material.

*Step 1: Reparameterization.* Similarly to [38; 8] with SGD (with momentum) and centralized adaptive gradient methods, define the following auxiliary sequence:  $Z_t = \bar{X}_t + \frac{\beta_1}{1-\beta_1}(\bar{X}_t - \bar{X}_{t-1})$ , 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.

*Step 2: Bounding gradient.* With the help of  $Z_t$ , we can remove the complicated update dependence on  $m_t$ , and perform convergence analysis to bound gradient of  $Z_t$ . Then bound gradient of  $\bar{X}_t$  by smoothness of gradient, which yields:

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

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, bounding  $D_1$  and  $D_2$  give rise to the terms related to  $\mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right]$ .

*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,

$$\mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] \leq \mathbb{E}[\|\Gamma_u^f\|^2] + \frac{d}{N} \frac{\sigma^2}{\epsilon}, \quad \text{where } \Gamma_u^f := \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) / \sqrt{u_{t,i}}. \quad \text{We can}$$

then transform  $\mathbb{E}[\|\Gamma_u^f\|^2]$  into  $\mathbb{E}[\|\Gamma_{\bar{U}}^f\|^2]$  by splitting out two error terms, then bounding the error terms as operated for  $D_2$  and  $D_3$ . Then, by plugging it into (4), we obtain the desired bound in Theorem 2.

**Proof of Theorem 3:** Recall the bound in (3) of Theorem 2. Since Algorithm 3 is a special case of Algorithm 2, the remaining of the proof consists of 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, so that  $\mathbb{E}[\sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}] = \mathbb{E}[\sum_{i=1}^N \sum_{j=1}^d (-\hat{v}_{0,i,j} + \hat{v}_{T-1,i,j})]$ . We can also prove  $|v_{t,i,j}| \leq G_\infty^2$  using  $\|g_{t,i}\|_\infty \leq G_\infty$ . Then we have  $\mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right] \leq \sum_{i=1}^N \sum_{j=1}^d \mathbb{E}[G_\infty^2] = NdG_\infty^2$ . Substituting into (3) yields the desired convergence bound for Algorithm 3.

### 3.4 Application to AdaGrad algorithm

In this section, we provide a decentralized version of AdaGrad [12] (optionally with momentum) converted by Algorithm 2, further supporting the usefulness of our decentralization framework. The required modification for decentralized AdaGrad is to specify line 4 of Algorithm 2 as follows:  $\hat{v}_{t,i} = \frac{t-1}{t} \hat{v}_{t-1,i} + \frac{1}{t} g_{t,i}^2$ , which is equivalent to  $\hat{v}_{t,i} = \frac{1}{t} \sum_{k=1}^t g_{k,i}^2$ . In this section, we call this algorithm decentralized AdaGrad.

The pseudo code of the algorithm is shown in Algorithm 4. There are two details in Algorithm 4 worth mentioning. The first one is that the introduced framework leverages momentum  $m_{t,i}$  in updates, while original AdaGrad does not use momentum. The momentum can be turned off by setting  $\beta_1 = 0$  and the convergence results will still hold. The other one is that in Decentralized AdaGrad, we use the average instead of the sum in the term  $\hat{v}_{t,i}$ . In other words, we write  $\hat{v}_{t,i} = \frac{1}{t} \sum_{k=1}^t g_{k,i}^2$ . This latter point is different from the original AdaGrad which actually uses  $\hat{v}_{t,i} = \sum_{k=1}^t g_{k,i}^2$ .

The reason is that in the original AdaGrad, a constant stepsize ( $\alpha$  independent of  $t$  or  $T$ ) is used with  $\hat{v}_{t,i} = \sum_{k=1}^t g_{k,i}^2$ . This is equivalent to using a well-known decreasing stepsize sequence  $\alpha_t = \frac{1}{\sqrt{t}}$  with  $\hat{v}_{t,i} = \frac{1}{t} \sum_{k=1}^t g_{k,i}^2$ . In our convergence analysis, which can be found below, we use a constant stepsize  $\alpha = O(\frac{1}{\sqrt{T}})$  to replace the decreasing stepsize sequence  $\alpha_t = O(\frac{1}{\sqrt{t}})$ . Such a replacement is popularly used in Stochastic Gradient Descent analysis for the sake of simplicity and to achieve a better convergence rate. In addition, it is easy to modify our theoretical framework to include decreasing stepsize sequences such as  $\alpha_t = O(\frac{1}{\sqrt{t}})$ . The convergence analysis for decentralized AdaGrad is shown in Theorem 4.

**Theorem 4.** Assume A1-A4. Set  $\alpha = \sqrt{N}/\sqrt{Td}$ . When  $\alpha \leq \frac{\epsilon_{0.5}}{16L}$ , decentralized AdaGrad yields the following regret bound

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \leq \frac{C'_1 \sqrt{d}}{\sqrt{TN}} D'_f + \frac{C'_2}{T} + \frac{C'_3 N^{1.5}}{T^{1.5} d^{0.5}} + \frac{\sqrt{N}(1 + \log(T))}{T} (dC'_4 + \frac{\sqrt{d}}{T^{0.5}} C'_5),$$

where  $D'_f := \mathbb{E}[f(Z_1)] - \min_z f(z) + \sigma^2$ ,  $C'_1 = C_1$ ,  $C'_2 = C_2$ ,  $C'_3 = C_3$ ,  $C'_4 = C_4 G_\infty^2$  and  $C'_5 = C_5 G_\infty^2$ .  $C_1, C_2, C_3, C_4, C_5$  are defined in Theorem 2 independent of  $d, T$  and  $N$ . In addition, the consensus of variables at different nodes is given by  $\frac{1}{N} \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 \leq \frac{N}{T} \left( \frac{1}{1-\lambda} \right)^2 G_\infty^2 \frac{1}{\epsilon}$ .

## 4 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 parallel stochastic gradient descent (D-PSGD) developed in [21] are conducted. We train a Convolutional Neural Network (CNN) with 3 convolution layers followed by a fully connected layer on MNIST [19]. 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 on experiments can be found in the supplementary material of our paper.

### 4.1 Effect of heterogeneity

**Homogeneous data:** The whole dataset is shuffled and evenly split into different nodes. Such a setting is possible when the nodes are in a computer cluster. We see, Figure 1(a), that decentralized AMSGrad and DADAM perform quite similarly while D-PSGD (labelled as DGD) is much slower

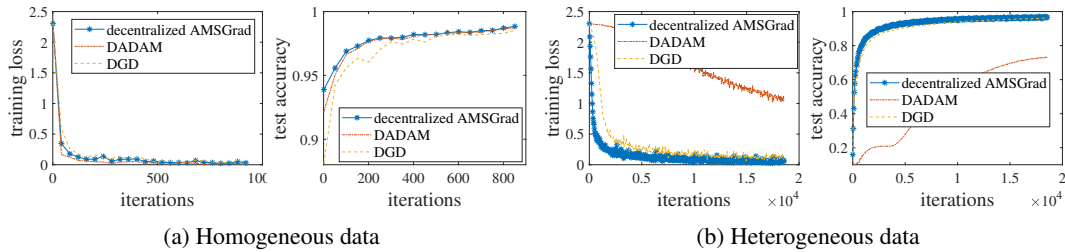


Figure 1: Training loss and Testing accuracy for homogeneous and heterogeneous data



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. Such a setting is common when data shuffling is prohibited, such as in federated learning. 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 in Figure 1(b).

## 4.2 Sensitivity to the Learning Rate

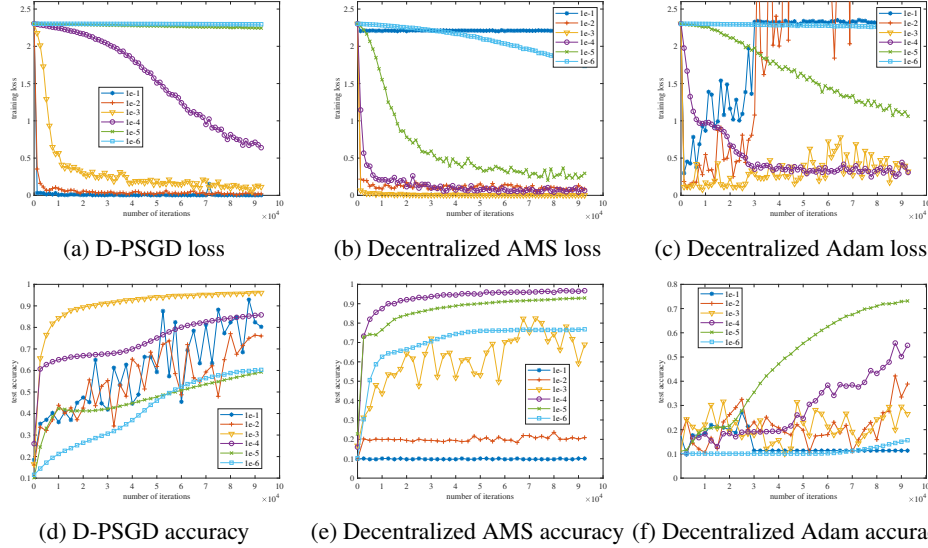


Figure 2: Training loss and testing accuracy comparison of different stepsizes for various methods

We compare the training loss and testing accuracies of different D-PSGD, DADAM, and our proposed Decentralized AMSGrad, with different stepsizes on *heterogeneous* data distribution. We use 5 nodes and the heterogeneous data distribution is created by assigning each node with data of only two labels. Note that there are no overlapping labels between different nodes. We observe Figure 2(a) and (d) that the stepsize  $10^{-3}$  works best for D-PSGD in terms of test accuracy and  $10^{-1}$  works best in terms of training loss. This difference is caused by the inconsistency among the model parameters on different nodes when the stepsize is large.

Figure 2(b) and (e) shows the performance of decentralized AMSGrad with different stepsizes. We see that its best performance is better than the one of D-PSGD and the performance is more stable (the test performance is less sensitive to stepsize tuning). As expected, the performance of DADAM is not as good as D-PSGD or decentralized AMSGrad, see Figure 2(c) and (f). Its divergence characteristic, highlighted Section 2.3, coupled with the heterogeneity in the data amplify its non-convergence issue in our experiments. 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 any newly proposed methods in the presented setting.

## 5 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 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. We also give an extension to a decentralized variant of AdaGrad for completeness of our converting scheme. Experiments show that the proposed algorithm achieves better performance than the baselines.

## References

- [1] Naman Agarwal, Brian Bullins, Xinyi Chen, Elad Hazan, Karan Singh, Cyril Zhang, and Yi Zhang. Efficient full-matrix adaptive regularization. In *International Conference on Machine Learning*, pages 102–110, 2019.
- [2] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Empirical Methods in Natural Language Processing*, pages 440–445, 2017.
- [3] 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.
- [4] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353, 2019.
- [5] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing markov chain on a graph. *SIAM review*, 46(4):667–689, 2004.
- [6] Stephen Boyd, Persi Diaconis, Pablo Parrilo, and Lin Xiao. Fastest mixing markov chain on graphs with symmetries. *SIAM Journal on Optimization*, 20(2):792–819, 2009.
- [7] 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.
- [8] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference for Learning Representations*, 2019.
- [9] Yongjian Chen, Tao Guan, and Cheng Wang. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 10(12):11259–11273, 2010.
- [10] Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *Symposium on Operating Systems Design and Implementation*, pages 571–582, 2014.
- [11] 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.
- [12] 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, 2011.
- [13] 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, 2011.
- [14] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2946–2953, 2013.
- [15] 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 *International Conference on Machine Learning*, pages 1529–1538, 2017.
- [16] 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.
- [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

- [18] Anastasia Koloskova, Sebastian U Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, pages 3478–3487, 2019.
- [19] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [20] Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *International Conference on Artificial Intelligence and Statistics*, pages 983–992, 2019.
- [21] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.
- [22] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *International Conference on Learning Representations*, 2018.
- [23] Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. Gnsd: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 315–321, 2019.
- [24] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference for Learning Representations*, 2019.
- [25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.
- [26] Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.
- [27] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48, 2009.
- [28] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.
- [29] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.
- [30] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- [31] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.
- [32] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.
- [33] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D<sup>2</sup>: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856, 2018.
- [34] Hanlin Tang, Chen Yu, Xiangru Lian, Tong Zhang, and Ji Liu. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *International Conference on Machine Learning*, pages 6155–6165, 2019.

- 484 [35] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and  
 485 Stephen Wright. Atomo: Communication-efficient learning via atomic sparsification. In  
 486 *Advances in Neural Information Processing Systems*, pages 9850–9861, 2018.
- 487 [36] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for  
 488 communication-efficient distributed optimization. In *Advances in Neural Information Process-*  
 489 *ing Systems*, pages 1299–1309, 2018.
- 490 [37] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over  
 491 nonconvex landscapes. In *International Conference on Machine Learning*, pages 6677–6686,  
 492 2019.
- 493 [38] Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic mo-  
 494 mentum methods for deep learning. In *International Joint Conference on Artificial Intelligence*,  
 495 pages 2955–2961, 2018.
- 496 [39] Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent.  
 497 *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.
- 498 [40] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive  
 499 methods for nonconvex optimization. In *Advances in Neural Information Processing Systems*,  
 500 pages 9793–9803, 2018.
- 501 [41] Dongruo Zhou, Jinghui Chen, Yuan Cao, Yiqi Tang, Ziyang Yang, and Quanquan Gu. On  
 502 the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint*  
 503 *arXiv:1808.05671*, 2018.
- 504 [42] Fangyu Zou and Li Shen. On the convergence of weighted adagrad with momentum for training  
 505 deep neural networks. *arXiv preprint arXiv:1808.03408*, 2018.

## Checklist

### 1. For all authors...

- (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
- (b) Did you describe the limitations of your work? [Yes]
- (c) Did you discuss any potential negative societal impacts of your work? [N/A]
- (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

### 2. If you are including theoretical results...

- (a) Did you state the full set of assumptions of all theoretical results? [Yes]
- (b) Did you include complete proofs of all theoretical results? [Yes]

### 3. If you ran experiments...

- (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Available upon demand
- (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
- (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
- (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

### 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

- (a) If your work uses existing assets, did you cite the creators? [Yes]
- (b) Did you mention the license of the assets? [No]
- (c) Did you include any new assets either in the supplemental material or as a URL? [No]
- (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Open source code and datasets.
- (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

### 5. If you used crowdsourcing or conducted research with human subjects...

- (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
- (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
- (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]



---

## On the Convergence of Decentralized Adaptive Gradient Methods (Appendix)

---

The main purpose of this appendix is to give thorough and detailed proofs for our convergence analysis described in the main paper. After having established several important Lemmas in Section A, we provide a proof for our main Theorem, namely Theorem 2, in Section B. Section C and Section D correspond to the proofs for the extension and application of Theorem 2 to the AMSGrad and AdaGrad algorithms used as prototypes of our general class of decentralized adaptive gradient methods. Section E contains additional numerical runs for more empirical insights on our scheme.

### A Proof of Auxiliary Lemmas

Similarly to [38; 8] with SGD (with momentum) and centralized adaptive gradient methods, define the following auxiliary sequence:

$$Z_t = \bar{X}_t + \frac{\beta_1}{1 - \beta_1} (\bar{X}_t - \bar{X}_{t-1}), \quad (5)$$

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.

**Lemma A.1.** *For the sequence defined in (5), we have*

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

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

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

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

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

558 which is the desired result.  $\square$

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

$$\sum_{i=1}^n |b_i(r) - \bar{b}(r)| \geq \sum_{i=1}^n |b_i(r') - \bar{b}(r')| \quad (6)$$

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

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

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

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

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

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

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

570 and

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

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

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

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

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

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

## 576 B Proof of Theorem 2

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

$$Z_t = \bar{X}_t + \frac{\beta_1}{1 - \beta_1} (\bar{X}_t - \bar{X}_{t-1}), \quad (8)$$

578 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  
579  $G_1, G_2, \dots, G_{t-1}$ ), we have

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

580 Assuming smoothness (A1) we have

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

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

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

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

$$\alpha \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\rangle \right] \quad (9)$$

$$\begin{aligned} & \leq \mathbb{E}[f(Z_t)] - \mathbb{E}[f(Z_{t+1})] + \frac{L}{2} \mathbb{E} [\|Z_{t+1} - Z_t\|^2] \\ & \quad + \alpha \frac{\beta_1}{1 - \beta_1} \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right]. \end{aligned} \quad (10)$$

583 In addition, we have

$$\begin{aligned} & \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\rangle \\ & = \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\rangle + \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right) \right\rangle \end{aligned} \quad (11)$$

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

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

$$\begin{aligned}
& -\frac{1}{2} \left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{1}{2} \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \\
& \geq \frac{1}{2} \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{3}{2} \left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 - \frac{3}{2} \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2, \quad (12)
\end{aligned}$$

585 where the inequalities are all due to Cauchy-Schwartz. Substituting (12) and (11) into (9), we get

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

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

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \\
& \leq \frac{2}{T\alpha} (\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + \frac{L}{T\alpha} \sum_{t=1}^T \mathbb{E}[\|Z_{t+1} - Z_t\|^2] \\
& + \frac{2}{T} \frac{\beta_1}{1 - \beta_1} \sum_{t=1}^T \underbrace{\mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right]}_{D_1} \\
& + \frac{2}{T} \sum_{t=1}^T \underbrace{\mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{\bar{U}_t}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right]}_{D_2} \\
& + \frac{3}{T} \sum_{t=1}^T \underbrace{\mathbb{E} \left[ \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 + \left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right]}_{D_3}. \quad (13)
\end{aligned}$$

587 Now we need to upper bound all the terms on RHS of the above inequality to get the convergence  
588 rate. For the terms composing  $D_3$  in (13), we can upper bound them by

$$\begin{aligned}
\left\| \frac{\nabla f(Z_t) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 & \leq \frac{1}{\min_{j \in [d]} [\bar{U}_t^{1/2}]_j} \|\nabla f(Z_t) - \nabla f(\bar{X}_t)\|^2 \\
& \leq L \frac{1}{\min_{j \in [d]} [\bar{U}_t^{1/2}]_j} \underbrace{\|Z_t - \bar{X}_t\|^2}_{D_4} \quad (14)
\end{aligned}$$

589 and

$$\begin{aligned} \left\| \frac{\frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 &\leq \frac{1}{\min_{j \in [d]} [\bar{U}_t^{1/2}]_j} \frac{1}{N} \sum_{i=1}^N \|\nabla f_i(x_{t,i}) - \nabla f(\bar{X}_t)\|^2 \\ &\leq L \underbrace{\frac{1}{\min_{j \in [d]} [\bar{U}_t^{1/2}]_j} \frac{1}{N} \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2}_{D_5}, \end{aligned} \quad (15)$$

590 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  
591 to bound  $D_4$  and  $D_5$ . Recall the update rule of  $X_t$ , we have

$$X_t = X_{t-1}W - \alpha \frac{M_{t-1}}{\sqrt{U_{t-1}}} = X_1 W^{t-1} - \alpha \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} W^k, \quad (16)$$

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

$$\sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 = \|X_t - \bar{X}_t \mathbf{1}_N^T\|_F^2 = \|X_t Q Q^T - X_t \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T\|_F^2 = \sum_{l=2}^N \|X_t q_l\|^2. \quad (17)$$

597 In addition, we can rewrite (16) as

$$X_t = X_1 W^{t-1} - \alpha \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} W^k = X_1 - \alpha \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} Q \Lambda^k Q^T, \quad (18)$$

598 where the last equality is because  $x_{1,i} = x_{1,j}$ , for all  $i, j$  and thus  $X_1 W = X_1$ . Then we have when  
599  $l > 1$ ,

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

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

601 Combining (17) and (19), we have

$$\begin{aligned} D_5 &= \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 = \sum_{l=2}^N \|X_t q_l\|^2 \\ &= \sum_{l=2}^N \alpha^2 \left\| \sum_{k=0}^{t-2} \frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}} \lambda_l^k q_l \right\|^2 \\ &\leq \alpha^2 \left( \frac{1}{1-\lambda} \right)^2 N d G_\infty^2 \frac{1}{\epsilon}, \end{aligned} \quad (20)$$

602 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  
603 us turn to  $D_4$ , it can be rewritten as

$$\begin{aligned} \|Z_t - \bar{X}_t\|^2 &= \left\| \frac{\beta_1}{1-\beta_1} (\bar{X}_t - \bar{X}_{t-1}) \right\|^2 = \left( \frac{\beta_1}{1-\beta_1} \right)^2 \alpha^2 \left\| \frac{1}{N} \sum_{i=1}^N \frac{m_{t-1,i}}{\sqrt{u_{t-1,i}}} \right\|^2 \\ &\leq \left( \frac{\beta_1}{1-\beta_1} \right)^2 \alpha^2 d \frac{G_\infty^2}{\epsilon}. \end{aligned} \quad (21)$$

604 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  
605  $\mathcal{O}(\alpha^2)$ . Next we will bound  $D_2$  and  $D_1$ . Define  $G_1 \triangleq \max_{t \in [T]} \max_{i \in [N]} \|\nabla f_i(x_{t,i})\|_\infty$ ,



606  $G_2 \triangleq \max_{t \in [T]} \|\nabla f(Z_t)\|_\infty$ ,  $G_3 \triangleq \max_{t \in [T]} \max_{i \in [N]} \|g_{t,i}\|_\infty$  and  $G_\infty = \max(G_1, G_2, G_3)$ .  
 607 Then we have

$$\begin{aligned}
 D_2 &= \sum_{t=1}^T \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N \nabla f_i(x_{t,i}) \odot \left( \frac{1}{\sqrt{\bar{U}_t}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right] \\
 &\leq \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{1}{\sqrt{[\bar{U}_t]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right| \right] \\
 &= \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{1}{\sqrt{[\bar{U}_t]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right| \frac{\sqrt{[\bar{U}_t]_j} + \sqrt{[u_{t,i}]_j}}{\sqrt{[\bar{U}_t]_j} + \sqrt{[u_{t,i}]_j}} \right] \\
 &= \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{[\bar{U}_t]_j - [u_{t,i}]_j}{[\bar{U}_t]_j \sqrt{[u_{t,i}]_j} + \sqrt{[\bar{U}_t]_j} [u_{t,i}]_j} \right| \right] \\
 &\leq \underbrace{\mathbb{E} \left[ \sum_{t=1}^T G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{[\bar{U}_t]_j - [u_{t,i}]_j}{2\epsilon^{1.5}} \right| \right]}_{D_6},
 \end{aligned} \tag{22}$$

608 where the last inequality is due to  $[u_{t,i}]_j \geq \epsilon$ , for all  $t, i, j$ . To simplify notations, define  $\|A\|_{abs} =$   
 609  $\sum_{i,j} |A_{ij}|$  to be the entry-wise  $L_1$  norm of a matrix  $A$ , then we obtain

$$\begin{aligned}
 D_6 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \|\bar{U}_t \mathbf{1}^T - U_t\|_{abs} \leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \|\bar{\tilde{U}}_t \mathbf{1}^T - \tilde{U}_t\|_{abs} \\
 &= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \|\tilde{U}_t \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T - \tilde{U}_t Q Q^T\|_{abs} \\
 &= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \left\| - \sum_{l=2}^N \tilde{U}_t q_l q_l^T \right\|_{abs},
 \end{aligned}$$

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

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

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

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

614 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  
 615  $q_1 = \frac{1}{\sqrt{N}} \mathbf{1}_N^T$ ) and  $q_i^T q_j = 0$  when  $i \neq j$ . Note that by definition of  $\|\cdot\|_{abs}$ , we have for all

616  $A, B, \|A + B\|_{abs} \leq \|A\|_{abs} + \|B\|_{abs}$ , then

$$\begin{aligned}
D_6 &\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \left\| - \sum_{l=2}^N \tilde{U}_t q_l q_l^T \right\|_{abs} \\
&= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \left\| - \sum_{k=1}^t (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) \sum_{l=2}^N q_l \lambda_l^k q_l^T \right\|_{abs} \\
&\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \sum_{j=1}^d \left\| \sum_{l=2}^N q_l \lambda_l^k q_l^T \right\|_1 \|(-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j\|_1 \\
&\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \sum_{j=1}^d \sqrt{N} \left\| \sum_{l=2}^N q_l \lambda_l^k q_l^T \right\|_2 \|(-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j\|_1 \\
&\leq \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \sum_{j=1}^d \|(-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j\|_1 \sqrt{N} \lambda^k \\
&= \frac{G_\infty^2}{N} \sum_{t=1}^T \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^t \|(-\hat{V}_{t-1-k} + \hat{V}_{t-k})\|_{abs} \sqrt{N} \lambda^k \\
&= \frac{G_\infty^2}{N} \frac{1}{2\epsilon^{1.5}} \sum_{o=0}^{T-1} \sum_{t=o+1}^T \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \sqrt{N} \lambda^{t-o} \\
&\leq \frac{G_\infty^2}{\sqrt{N}} \frac{1}{2\epsilon^{1.5}} \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs},
\end{aligned} \tag{23}$$

617 where  $\lambda = \max(|\lambda_2|, |\lambda_N|)$ . Combining (22) and (23), we have

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

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

$$\begin{aligned}
D_1 &= \sum_{t=1}^T \mathbb{E} \left[ \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^N m_{t-1,i} \odot \left( \frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}} \right) \right\rangle \right] \\
&\leq \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{1}{\sqrt{[u_{t-1,i}]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right| \right] \\
&= \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \left( \frac{1}{\sqrt{[u_{t-1,i}]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right) \frac{\sqrt{[u_{t,i}]_j} + \sqrt{[u_{t-1,i}]_j}}{\sqrt{[u_{t,i}]_j} + \sqrt{[u_{t-1,i}]_j}} \right| \right] \\
&\leq \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \left| \frac{1}{2\epsilon^{1.5}} ([u_{t-1,i}]_j - [u_{t,i}]_j) \right| \right] \\
&\stackrel{(a)}{\leq} \sum_{t=1}^T \mathbb{E} \left[ G_\infty^2 \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d \frac{1}{2\epsilon^{1.5}} |([\tilde{u}_{t-1,i}]_j - [\tilde{u}_{t,i}]_j)| \right] \\
&= G_\infty^2 \frac{1}{2\epsilon^{1.5}} \frac{1}{N} \mathbb{E} \left[ \sum_{t=1}^T \|\tilde{U}_{t-1} - \tilde{U}_t\|_{abs} \right],
\end{aligned} \tag{24}$$

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 addition, by update rule of  $U_t$ , we have

$$\begin{aligned}
& \sum_{t=1}^T \|\tilde{U}_{t-1} - \tilde{U}_t\|_{abs} \\
&= \sum_{t=1}^T \|\tilde{U}_{t-1} - (\tilde{U}_{t-1} - \hat{V}_{t-2} + \hat{V}_{t-1})W\|_{abs} \\
&= \sum_{t=1}^T \|\tilde{U}_{t-1}(QQ^T - Q\Lambda Q^T) + (-\hat{V}_{t-2} + \hat{V}_{t-1})W\|_{abs} \\
&= \sum_{t=1}^T \|\tilde{U}_{t-1}(\sum_{l=2}^N q_l(1 - \lambda_l)q_l^T) + (-\hat{V}_{t-2} + \hat{V}_{t-1})W\|_{abs} \\
&\leq \sum_{t=1}^T \left\| \sum_{k=1}^{t-1} (-\hat{V}_{t-2-k} + \hat{V}_{t-1-k}) \sum_{l=2}^N q_l \lambda_l^k (1 - \lambda_l) q_l^T \right\|_{abs} + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})W\|_{abs} \\
&\leq \sum_{t=1}^T \left( \sum_{k=1}^{t-1} \|\hat{V}_{t-2-k} - \hat{V}_{t-1-k}\|_{abs} \sqrt{N} \lambda^k \right) + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \\
&= \sum_{t=1}^T \left( \sum_{o=1}^{t-1} \|\hat{V}_{t-2-o} - \hat{V}_{t-1-o}\|_{abs} \sqrt{N} \lambda^{t-o} \right) + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \\
&= \sum_{o=1}^{T-1} \sum_{t=o+1}^T \left( \|\hat{V}_{t-2-o} - \hat{V}_{t-1-o}\|_{abs} \sqrt{N} \lambda^{t-o} \right) + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \\
&\leq \sum_{o=1}^{T-1} \frac{\lambda}{1-\lambda} \left( \|\hat{V}_{t-2-o} - \hat{V}_{t-1-o}\|_{abs} \sqrt{N} \right) + \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \\
&\leq \frac{1}{1-\lambda} \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \sqrt{N}.
\end{aligned} \tag{25}$$

Combining (24) and (25), we have

$$D_1 \leq G_\infty^2 \frac{1}{2\epsilon^{1.5}} \frac{1}{N} \mathbb{E} \left[ \frac{1}{1-\lambda} \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \sqrt{N} \right]. \tag{26}$$

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

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

$$= 2\alpha^2 \left( \frac{\beta_1}{1-\beta_1} \right)^2 G_\infty^2 \frac{1}{N} \frac{1}{2\epsilon^2} \|\tilde{U}_t - \tilde{U}_{t-1}\|_{abs} + 2\alpha^2 \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2, \quad (27)$$

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  $\max(\cdot, \epsilon)$  is 1-Lipschitz. Then, we have

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

where the last inequality is due to (25).

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

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

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

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

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

$$\begin{aligned} \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] & = \mathbb{E}_{G_{1:t-1}} \left[ \mathbb{E}_{G_t|G_{1:t-1}} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right\|^2 \right] \right] \\ & \leq \mathbb{E}_{G_{1:t-1}} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 + \frac{d}{N} \frac{\sigma^2}{\epsilon} \right] \\ & = \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 \right] + \frac{d}{N} \frac{\sigma^2}{\epsilon}. \end{aligned} \quad (28)$$

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

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

639 Summing over  $T$ , we have

$$\begin{aligned}
& \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\|^2 \right] \\
&\leq 2 \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] + 2 \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N G_\infty^2 \frac{1}{\sqrt{\epsilon}} \left\| \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right\|_1 \right]. \quad (29)
\end{aligned}$$

640 For the last term on RHS of (29), we can bound it similarly as what we did for  $D_2$  from (22) to (23),  
641 which yields

$$\begin{aligned}
& \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N G_\infty^2 \frac{1}{\sqrt{\epsilon}} \left\| \frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right\|_1 \right] \leq \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N G_\infty^2 \frac{1}{\sqrt{\epsilon}} \frac{1}{2\epsilon^{1.5}} \|u_{t,i} - \bar{U}_t\|_1 \right] \\
&= \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} G_\infty^2 \frac{1}{2\epsilon^2} \|\bar{U}_t \mathbf{1}^T - U_t\|_{abs} \right] \\
&\leq \sum_{t=1}^T \mathbb{E} \left[ \frac{1}{N} G_\infty^2 \frac{1}{2\epsilon^2} \left\| - \sum_{l=2}^N \tilde{U}_t q_l q_l^T \right\|_{abs} \right] \\
&\leq \frac{1}{\sqrt{N}} G_\infty^2 \frac{1}{2\epsilon^2} \mathbb{E} \left[ \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \right]. \quad (30)
\end{aligned}$$



642 Further, we have

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

643 and the last term on RHS of the above inequality can be bounded following similar procedures from  
644 (15) to (20), as what we did for  $D_3$ . Completing the procedures yields

$$\begin{aligned}
\sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{1}{N} \sum_{i=1}^N \frac{\nabla f_i(\bar{X}_t) - \nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\|^2 \right] & \leq \sum_{t=1}^T \mathbb{E} \left[ L \frac{1}{\epsilon} \frac{1}{N} \sum_{i=1}^N \|x_{t,i} - \bar{X}_t\|^2 \right] \\
& \leq \sum_{t=1}^T \mathbb{E} \left[ L \frac{1}{\epsilon} \frac{1}{N} \alpha^2 \left( \frac{1}{1-\lambda} \right) N d G_\infty^2 \frac{1}{\epsilon} \right] \quad (31) \\
& = T L \frac{1}{\epsilon^2} \alpha^2 \left( \frac{1}{1-\lambda} \right) d G_\infty^2.
\end{aligned}$$

645 Finally, combining (28) to (31), we get

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

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

647 Combining all above, we obtain

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \\
& \leq \frac{2}{T\alpha} (\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) \\
& \quad + \frac{L}{T} \alpha \left( \frac{\beta_1}{1-\beta_1} \right)^2 \frac{G_\infty^2}{\sqrt{N}} \frac{1}{\epsilon^2} \frac{1}{1-\lambda} \mathbb{E}[\mathcal{V}_T] \\
& \quad + \frac{8L}{T} \alpha \frac{1}{\sqrt{\epsilon}} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] + 8 L^2 \alpha \frac{1}{\epsilon^2} \alpha^2 \left( \frac{1}{1-\lambda} \right) d G_\infty^2 \quad (32) \\
& \quad + \frac{4L}{T} \alpha \frac{1}{\sqrt{N}} G_\infty^2 \frac{1}{2\epsilon^2} \mathbb{E} \left[ \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs} \right] + 2 L \alpha \frac{d}{N} \frac{\sigma^2}{\epsilon}
\end{aligned}$$

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

648 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

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] \\
& \leq \frac{4}{T\alpha} (\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + 4L\alpha \frac{d}{N} \frac{\sigma^2}{\epsilon} \\
& \quad + 6\alpha^2 d \left( \left( \frac{\beta_1}{1-\beta_1} \right)^2 + \left( \frac{1}{1-\lambda} \right)^2 \right) L \frac{G_\infty^2}{\epsilon^{1.5}} + 16\alpha^3 L^2 \left( \frac{1}{1-\lambda} \right) d \frac{G_\infty^2}{\epsilon^2} \\
& \quad + \frac{2}{T\epsilon^{1.5}} \frac{G_\infty^2}{\sqrt{N}} \frac{1}{1-\lambda} \left( L\alpha \left( \frac{\beta_1}{1-\beta_1} \right)^2 \frac{1}{\epsilon^{0.5}} + \lambda + \frac{\beta_1}{1-\beta_1} + 2L\alpha \frac{1}{\epsilon^{0.5}} \lambda \right) \mathbb{E} [\mathcal{V}_T] \\
& \leq \frac{4}{T\alpha} (\mathbb{E}[f(Z_1)] - \min_x f(x)) + 4L\alpha \frac{d}{N} \frac{\sigma^2}{\epsilon} \\
& \quad + 6\alpha^2 d \left( \left( \frac{\beta_1}{1-\beta_1} \right)^2 + \left( \frac{1}{1-\lambda} \right)^2 \right) L \frac{G_\infty^2}{\epsilon^{1.5}} + 16\alpha^3 d L^2 \left( \frac{1}{1-\lambda} \right) \frac{G_\infty^2}{\epsilon^2} \\
& \quad + \frac{2}{T\epsilon^{1.5}} \frac{G_\infty^2}{\sqrt{N}} \frac{1}{1-\lambda} \left( L\alpha \left( \frac{\beta_1}{1-\beta_1} \right)^2 \frac{1}{\epsilon^{0.5}} + \lambda + \frac{\beta_1}{1-\beta_1} + 2L\alpha \frac{1}{\epsilon^{0.5}} \lambda \right) \mathbb{E} [\mathcal{V}_T] \\
& \leq C_1 \left( \frac{1}{T\alpha} (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \alpha \frac{d\sigma^2}{N} \right) + C_2 \alpha^2 d + C_3 \alpha^3 d + \frac{1}{T\sqrt{N}} (C_4 + C_5 \alpha) \mathbb{E} [\mathcal{V}_T]
\end{aligned} \tag{33}$$

649 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

650 RHS of (32) to the LHS to cancel it using the assumption  $8L\alpha \frac{1}{\sqrt{\epsilon}} \leq \frac{1}{2}$  followed by multiplying both  
651 sides by 2. The constants introduced in the last step are defined as following

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

$$C_5 = \frac{2}{\epsilon^2} \frac{1}{1-\lambda} L \left( \frac{\beta_1}{1-\beta_1} \right)^2 G_\infty^2 + \frac{4}{\epsilon^2} \frac{\lambda}{1-\lambda} L G_\infty^2.$$

652 Substituting into  $Z_1 = \overline{X}_1$  completes the proof.

□

### 653 C Proof of Theorem 3

654 Under some assumptions stated in Corollary 2.1, we have that

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] &\leq C_1 \frac{\sqrt{d}}{\sqrt{TN}} \left( (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \sigma^2 \right) + C_2 \frac{N}{T} + C_3 \frac{N^{1.5}}{T^{1.5}d^{0.5}} \\ &\quad + \left( C_4 \frac{1}{T\sqrt{N}} + C_5 \frac{1}{T^{1.5}d^{0.5}} \right) \mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right] \end{aligned} \quad (34)$$

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

657 Since Algorithm 3 is a special case of 2, building on result of Theorem 2, we just need to characterize  
658 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  
659 update rule of Algorithm 3, we know  $\hat{V}_t$  is non decreasing and thus

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

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

661 Further, because  $\|g_{t,i}\|_\infty \leq G_\infty$  for all  $t, i$  and  $v_{t,i}$  is a exponential moving average of  $g_{k,i}^2, k =$   
662  $1, 2, \dots, t$ , we know  $|[v_{t,i}]_j| \leq G_\infty^2$ , for all  $t, i, j$ . In addition, by update rule of  $\hat{V}_t$ , we also know  
663 each element of  $\hat{V}_t$  also cannot be greater than  $G_\infty^2$ , i.e.  $|\hat{v}_{t,i,j}| \leq G_\infty^2$ , for all  $t, i, j$ . Given the fact  
664 that  $[\hat{v}_{0,i}]_j \geq 0$ , we have

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

665 Substituting the above into (34), we have

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] &\leq C_1 \frac{\sqrt{d}}{\sqrt{TN}} \left( (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \sigma^2 \right) + C_2 \frac{N}{T} + C_3 \frac{N^{1.5}}{T^{1.5}d^{0.5}} \\ &\quad + \left( C_4 \frac{1}{T\sqrt{N}} + C_5 \frac{1}{T^{1.5}d^{0.5}} \right) NdG_\infty^2 \\ &= C'_1 \frac{\sqrt{d}}{\sqrt{TN}} \left( (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \sigma^2 \right) + C'_2 \frac{N}{T} + C'_3 \frac{N^{1.5}}{T^{1.5}d^{0.5}} \\ &\quad + C'_4 \frac{\sqrt{Nd}}{T} + C'_5 \frac{Nd^{0.5}}{T^{1.5}}, \end{aligned} \quad (35)$$

666 where we have

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

667 and we conclude the proof.  $\square$

## 668 D Proof of Theorem 4

669 The proof follows the same flow as that of Theorem 3. Under assumptions stated in Corollary 2.1, set  
 670  $\alpha = \sqrt{N}/\sqrt{Td}$ , we have that

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] &\leq C_1 \frac{\sqrt{d}}{\sqrt{TN}} \left( (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \sigma^2 \right) + C_2 \frac{N}{T} + C_3 \frac{N^{1.5}}{T^{1.5}d^{0.5}} \\ &\quad + \left( C_4 \frac{1}{T\sqrt{N}} + C_5 \frac{1}{T^{1.5}d^{0.5}} \right) \mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right], \end{aligned} \quad (37)$$

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

673 Again, Since decentralized AdaGrad is a special case of 2, we can apply Corollary 2.1 and what we  
 674 need is to upper bound  $\mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right]$  derive convergence rate. By the update rule  
 675 of decentralized AdaGrad, we have  $\hat{v}_{t,i} = \frac{1}{t} (\sum_{k=1}^t g_{k,i}^2)$  for  $t \geq 1$  and  $\hat{v}_{0,i} = \epsilon \mathbf{1}$ . Then we have for  
 676  $t \geq 3$ ,

$$\begin{aligned} &\mathbb{E} \left[ \sum_{t=1}^T \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs} \right] \\ &= \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^d |-\hat{v}_{t-2,i} + \hat{v}_{t-1,i}| \right] \\ &\leq \mathbb{E} \left[ \sum_{t=3}^T \sum_{i=1}^N \sum_{j=1}^d \left| -\frac{1}{t-2} \left( \sum_{k=1}^{t-2} g_{k,i}^2 \right) + \frac{1}{t-1} \left( \sum_{k=1}^{t-1} g_{k,i}^2 \right) \right| \right] + Nd(G_\infty^2 - \epsilon) \\ &\leq \mathbb{E} \left[ \sum_{t=3}^T \sum_{i=1}^N \sum_{j=1}^d \left| \left( \frac{1}{t-1} - \frac{1}{t-2} \right) \left( \sum_{k=1}^{t-2} g_{k,i}^2 \right) + \frac{1}{t-1} g_{t-1,i}^2 \right| \right] + NdG_\infty^2 \\ &= \mathbb{E} \left[ \sum_{t=3}^T \sum_{i=1}^N \sum_{j=1}^d \left| \left( -\frac{1}{(t-1)(t-2)} \right) \left( \sum_{k=1}^{t-2} g_{k,i}^2 \right) + \frac{1}{t-1} g_{t-1,i}^2 \right| \right] + NdG_\infty^2 \\ &\leq \mathbb{E} \left[ \sum_{t=3}^T \sum_{i=1}^N \sum_{j=1}^d \max \left( \frac{1}{(t-1)(t-2)} \left( \sum_{k=1}^{t-2} g_{k,i}^2 \right), \frac{1}{t-1} g_{t-1,i}^2 \right) \right] + NdG_\infty^2 \\ &\leq \mathbb{E} \left[ Nd \sum_{t=3}^T \frac{G_\infty^2}{t-1} \right] + NdG_\infty^2 \\ &\leq NdG_\infty^2 \log(T) + NdG_\infty^2 \\ &= NdG_\infty^2 (\log(T) + 1) \end{aligned}$$

677 where the first equality is because we defined  $\hat{V}_{-1} \triangleq \hat{V}_0$  previously and  $\|g_{k,i}\|_\infty \leq G_\infty$  by assump-  
 678 tion.



679 Substituting the above into (37), we have

$$\begin{aligned}
\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \left\| \frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}} \right\|^2 \right] &\leq C_1 \frac{\sqrt{d}}{\sqrt{TN}} \left( (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \sigma^2 \right) + C_2 \frac{N}{T} + C_3 \frac{N^{1.5}}{T^{1.5} d^{0.5}} \\
&\quad + \left( C_4 \frac{1}{T\sqrt{N}} + C_5 \frac{1}{T^{1.5} d^{0.5}} \right) N d G_\infty^2 (\log(T) + 1) \\
&= C'_1 \frac{\sqrt{d}}{\sqrt{TN}} \left( (\mathbb{E}[f(Z_1)] - \min_x f(x)) + \sigma^2 \right) + C'_2 \frac{N}{T} + C'_3 \frac{N^{1.5}}{T^{1.5} d^{0.5}} \\
&\quad + C'_4 \frac{d\sqrt{N}(\log(T) + 1)}{T} + C'_5 \frac{(\log(T) + 1)N\sqrt{d}}{T^{1.5}},
\end{aligned}$$

680 where we have

$$C'_1 = C_1 \quad C'_2 = C_2 \quad C'_3 = C_3 \quad C'_4 = C_4 G_\infty^2 \quad C'_5 = C_5 G_\infty^2. \quad (38)$$

681 and we conclude the proof.  $\square$

## 682 E Additional Experiments and Details

683 In this section, we compare the training loss and testing accuracy of different algorithms, namely  
 684 Decentralized Stochastic Gradient Descent (D-PSGD), Decentralized Adam (DADAM) and our  
 685 proposed Decentralized AMSGrad, with different stepsizes on heterogeneous data distribution. We  
 686 use 5 nodes and the heterogeneous data distribution is created by assigning each node with data of  
 687 only two labels. Note that there are no overlapping labels between different nodes. For all algorithms,  
 688 we compare stepsizes in the grid  $[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$ .

689 Figure 3 shows the training loss and test accuracy for D-PSGD algorithm. We observe that the  
 690 stepsize  $10^{-3}$  works best for D-PSGD in terms of test accuracy and  $10^{-1}$  works best in terms of  
 691 training loss. This difference is caused by the inconsistency among the value of parameters on  
 692 different nodes when the stepsize is large. The training loss is calculated as the average of the loss  
 693 value of different local models evaluated on their local training batch. Thus, while the training loss is  
 694 small at a particular node, the test accuracy will be low when evaluating data with labels not seen by  
 695 the node (recall that each node contains data with different labels since we are in the heterogeneous  
 696 setting).

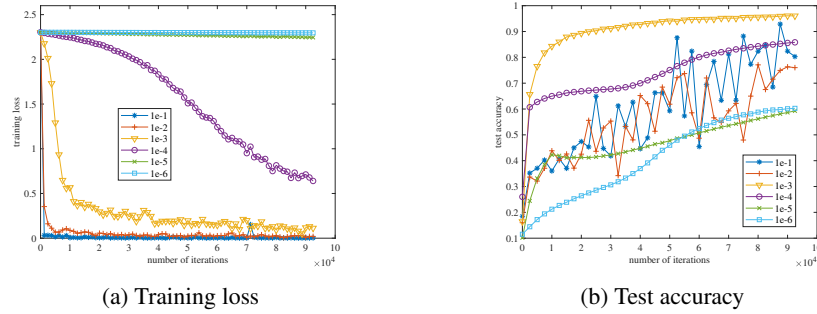


Figure 3: Performance comparison of different stepsizes for D-PSGD

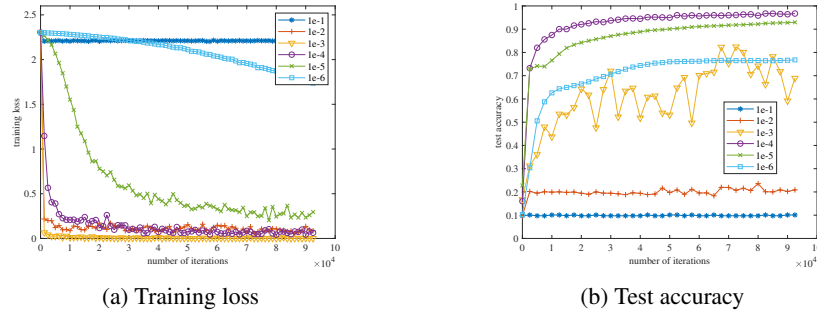


Figure 4: Performance comparison of different stepsizes for decentralized AMSGrad

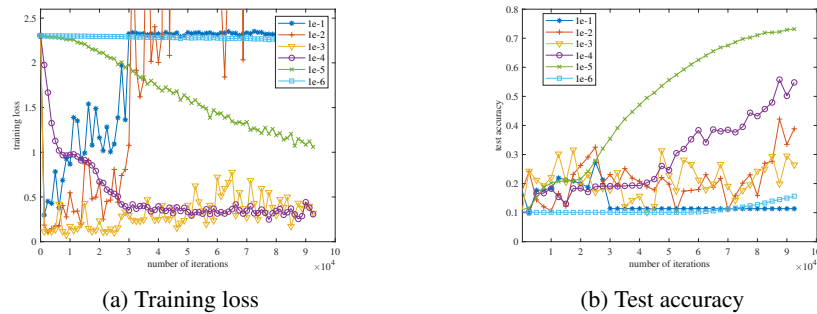


Figure 5: Performance comparison of different stepsizes for DADAM

697 Figure 4 shows the performance of decentralized AMSGrad with different stepsizes. We see that its  
698 best performance is better than the one of D-PSGD. Its performance is also more stable in the sense  
699 that the test performance is less sensitive to stepsize tuning according to our experiments.

700 Figure 5 displays the performance of Decentralized Adam algorithm. As expected, the performance  
701 of DADAM is not as good as D-PSGD or decentralized AMSGrad. Its divergence characteristic,  
702 highlighted Section 2.3, coupled with the heterogeneity in the data amplify its non-convergence issue  
703 in our experiments. From the experiments above, we can see the benefits of decentralized AMSGrad  
704 both in terms of performance and ease of parameter tuning, and the importance of ensuring the  
705 theoretical convergence of any newly proposed methods in the presented setting.

## 706 **F Improvements Made Over Last Submission**

707 We have added discussion on how to set mixing matrix  $W$  and how it affect  $\lambda$  defined in the paper.  
708 The proof sketch section is shortened to save more space. The comparison of different algorithms  
709 with different stepsizes are moved from Appendix to the main paper. Some other changes throughout  
710 the paper are also made to address comments from the previous round of review.