Convergent Adaptive Gradient Methods in Decentralized Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Adaptive gradient methods including Adam, AdaGrad, and their variants are proven to be very successful for training machine learning models such as neural nets in the past a few years. At the same time, distributed optimization is becoming increasingly popular, partly due to its success in training neural nets. With the growth of computing power and the need for using machine learning models on mobile devices, the communication cost of distributed training algorithms becomes unignorable. In response to this, more and more attention is shifted from the traditional parameter server training paradigm to decentralized training paradigms, which usually require lower communication costs. In this paper, we try to rigorously incorporate adaptive gradient methods into decentralized training, coming up with convergent decentralized adaptive gradient methods. Specifically, we propose an algorithmic framework that can convert existing adaptive gradient methods to their decentralized counterparts. In addition, we rigorously analyze the convergence behavior of the proposed algorithmic framework and show that if an adaptive gradient method satisfy some specific conditions, its converted counterpart is also convergent. Finally, using the framework, we proposed the first convergent decentralized adaptive gradient method.

1 Introduction

Distributed training of machine learning models is drawing increasing attention in the past few years due to its practical benefits and necessities. Due to the evolution of computing capabilities of CPUs and GPUs, computation time in distributed training is gradually dominated by the communication time in many circumstances [6, 16]. In response to this fact, a large amount of recent works has been focusing on reducing communication cost for distributed training [2, 14, 25, 21, 24, 23]. In the traditional parameter 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 and quantization. Despite these communication reduction techniques, the amount of data flow of the parameter server usually scales linearly with the number of workers. Due to this limitation, there is a rising interest in the research community in the decentralized training paradigm [9], where the parameter server is removed and every node only communicates with its neighbors. It has been shown in Lian et al. [13] that decentralized training algorithms can outperform parameter server-based algorithms when the training bottleneck is the communication cost. The decentralized training paradigm is also preferred when a parameter server is not available.

In parallel to distributed training, another effective way to accelerate training is by using adaptive gradient methods like AdaGrad [8], Adam [11] and AMSGrad [19]. Their practical benefits are proven by their popularity in training neural nets, featured by faster convergence and ease of parameter tuning compared with SGD.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Despite a large amount of literature in distributed optimization, there have been few works seriously considering bringing adaptive gradient methods into distributed training, largely due to the lack of understanding in convergence behavior of adaptive gradient methods.

In this paper, we investigate the possibility of using adaptive gradient methods in the decentralized training paradigm. Designing adaptive methods in such settings is highly non-trivial due to the already complicated update rules and the interaction between the effect of using adaptive learning rates and decentralized communication protocols.

The key result of this work is a general technique that can convert an adaptive gradient method from a centralized method to a decentralized method. More importantly, we provide a theoretical verification interface for analyzing the behavior of decentralized adaptive gradient methods converted by our technique.

By using our proposed technique, we also present a new decentralized optimization algorithm, called decentralized AMSGrad, converted by our technique from AMSGrad. Build on our proposed framework for analyzing the type of algorithms, we can characterize the convergence rate of decentralized AMSGrad, which is the first convergent decentralized adaptive gradient method.

A novel technique in our framework is a mechanism to enforce a consensus on adaptive learning rates at different nodes. We show the importance of consensus on adaptive learning rates by proving a divergent problem instance for a recently proposed decentralized adaptive gradient method DADAM, which lacks consensus mechanisms on adaptive learning rates.

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. $\|A\|_{abs} = \sum_{i,j} A_{i,j}$. For the ease of presentation, here we also introduce some notations that will be used later in the paper.

- $G_t = [g_{t,1}, g_{t,2}, ..., g_{t,N}]$
- $M_t = [m_{t,1}, m_{t,2}, ..., m_{t,N}]$
- $X_t = [x_{t,1}, x_{t,2}, ..., x_{t,N}]$
- $\overline{\nabla} f(X_t) = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_{t,i})$
- $U_t = [u_{t,1}, u_{t,2}, ..., u_{t,N}]$
- $\tilde{U}_t = [\tilde{u}_{t,1}, \tilde{u}_{t,2}, ..., \tilde{u}_{t,N}]$
- $V_t = [v_{t,1}, v_{t,2}, ..., v_{t,N}]$
- $\hat{V}_t = [\hat{v}_{t,1}, \hat{v}_{t,2}, ..., \hat{v}_{t,N}]$
- $\bar{X}_t = \frac{1}{N}\sum_{i=1}^{N} x_{t,i}$
- $\bar{U}_t = \frac{1}{N}\sum_{i=1}^{N} u_{t,i}$
- $\bar{\tilde{U}}_t = \frac{1}{N}\sum_{i=1}^{N} \tilde{u}_{t,i}$

Also, we will introduce a $N \times N$ matrix W later in the paper, we denote λ_i to be its ith largest eigenvalue and define $\lambda \triangleq \max(|\lambda_2|, |\lambda_N|)$.

2 Related work

Decentralized optimization: Decentralized optimization has a long history, traditional decentralized optimization methods include well-know algorithms such as ADMM [4], dual averaging [9], distributed subgradient descent [18]. More recent algorithms include Extra [20], Next [7] and Prox-PDA [10]. While these algorithms were commonly used in applications other than deep learning, recent algorithmic advances in the machine learning community have shown that decentralized optimization can be useful for training neural nets. Lian et al. [13] showed that a stochastic version of decentralized subgradient descent can outperform parameter server-based algorithms when the communication cost is high. [22] proposed D^2 that improves the convergence rate over stochastic subgradient descent. [3] proposed the Stochastic Gradient Push that is more robust to network failures for training neural nets. The study of decentralized training in the machine learning community is only at its initial stage. No one has seriously considered designing adaptive gradient methods in the setting of decentralized training until the recent work Nazari et al. [17], a decentralized version of AMSGrad [19] is proposed in Nazari et al. [17] and it is proven to satisfy some non-standard regret.

Adaptive gradient methods: Adaptive gradient methods are popularized in recent years due to their superior performance in training neural nets. The type of methods usually refers to AdaGrad [8], Adam [11], and their variants. Key features of such methods include the use of momentum and adaptive learning rates (which means the learning rate is changing during optimization and the learning rates on different coordinates might be different). The most adaptive gradient is Adam, which is believed to converge until the recent work Reddi et al. [19] pointed out an error in the convergence analysis of Adam. Since then, many research efforts in the community are investigated into analyzing the convergence behavior of adaptive gradient methods. Ward et al. [26], Li and Orabona [12] analyzed convergence of a variant of AdaGrad without coordinate-wise learning rates. Chen et al. [5] analyzed the convergence behavior of a broad class of algorithms including AMSGrad [19] and AdaGrad. Zou and Shen [29] provided a unified convergence analysis for AdaGrad with momentum. A few recent adaptive gradient methods can be found in Agarwal et al. [1], Luo et al. [15], Zaheer et al. [28].

vspace-0.1in

3 Decentralized training and divergence of DADAM

3.1 Decentralized optimization

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

$$\min_x \frac{1}{N}\sum_{i=1}^{N} f_i(x) \tag{1}$$

where f_i is only accessible by the ith node. For neural net training, f_i can be viewed as the average loss of data located at node i.

Throughout the paper, we make the following assumptions for analyzing the convergence behavior of different algorithms.

Assumptions

A1: f_i's are differentiable and the gradients is L-Lipschitz, i.e. $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \forall x, y$.

A2: We assume at iteration t, node i can access a stochastic gradient $g_{t,i}$. In addition, the stochastic gradients have bounded L_∞ norm and the gradients of f_i are also bounded, i.e. $\|g_{t,i}\| \leq G_\infty$, $\|\nabla f_i(x)\|_\infty \leq G_\infty$.

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

The assumptions A1 and A3 are standard in distributed optimization. A2 is a little stronger than the traditional assumption that the estimator has bounded variance, it is commonly used in analyses for adaptive gradient methods [5, 26]. One thing that should be noted is that the bounded gradient estimator assumption in A2 implies the bounded variance assumption in A3, we denote the variance bound and the estimator bound differently to avoid confusion when we use them for different purposes.

In decentralized optimization, the nodes are connected as a graph and each node only communicates to its neighbors. In such cases, one usually construct a matrix W for information sharing when designing algorithms. As can be expected, W cannot be arbitrary, the key properties required for W are listed in A4.

A4: Assumptions on matrix W.

1). $\sum_{j=1}^N W_{i,j} = 1$, $\sum_{i=1}^N W_{i,j} = 1$, $W_{i,j} \geq 0$.

2). Denote λ_i to be ith largest eigenvalue of W, we have $\lambda_1 = 1$, $|\lambda_2| < 1$, $|\lambda_N| < 1$.

3). $W_{i,j} = 0$ if node i and node j are not neighbors.

Throughout this paper, we will assume A1-A4 hold.

3.2 Divergence of DADAM

Recently, Nazari et al. [17] initiated a trial to bring adaptive gradient methods into decentralized optimization, the resulting algorithm is DADAM, which is shown in Algorithm 1.

Algorithm 1 DADAM(with N nodes)

```
**Input:** learning rate $\alpha$, current point $X_t$, $u_{\frac{1}{2},i} = \hat{v}_{0,i} = \epsilon\mathbf{1}, \forall i$, $m_0 = 0$ mixing matrix $W$
**for** $t = 1, 2, ..., T$ **do**
    $g_{t,i} \leftarrow \nabla f_i(x_{t,i}) + \xi_{t,i}$
    $m_{t,i} = \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}$
    $v_{t,i} = \beta_2 v_{t-1,i} + (1-\beta_2) g_{t,i}^2$
    $\hat{v}_{t,i} = \beta_3 \hat{v}_{t,i} + (1-\beta_3) \max(\hat{v}_{t-1,i}, v_{t,i})$
    $x_{t+\frac{1}{2},i} = \sum_{j=1}^{N} W_{ij} x_{t,j}$
    $x_{t+1,i} = x_{t+\frac{1}{2},i} - \alpha \frac{m_{t,i}}{\sqrt{\hat{v}_{t,i}}}$
**end for**
```

DADAM is essentially a decentralized version of AMSGrad and the key modification is the use of a consensus step on optimization variable x to transmit information across the network, encouraging convergence. The matrix W is a doubly stochastic matrix (which satisfies A4) for achieving average consensus of x. Introducing such a mixing matrix is a standard approach for decentralizing an algorithm, such as distributed gradient descent [18, 27]. It is proven in Nazari et al. [17] that DADAM admits a non-standard regret bound in the online setting, however, whether the algorithm can converge to stationary points in standard offline settings such training neural nets is still unknown.

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

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

Proof: Consider a 1 dimensional optimization problem distributed onto two nodes

$$\min_x \frac{1}{2} \sum_{i=1}^{2} f_i(x) \tag{2}$$

where $f_i(x) = \frac{1}{2}(x - a_i)^2$ and $a_1 = 0$, $a_2 = 1$.

The network contains only two nodes and the matrix W satisfy $W_{ij} = \frac{1}{2}, \forall i, j$.

We consider running DADAM with $\beta_1 = \beta_2 = \beta_3 = 0$ and $\epsilon = 0.6$ for simplicity. Suppose we initialize DADAM at $x_{1,i} = 0, \forall i$ and use learning rate $\alpha = 0.001$. We have at $x_{1,i} = 0$, $\nabla f_1(x_{1,1}) = 0$, $\nabla f_2(x_{1,2}) = 1$, this will lead to $\hat{v}_{1,1} = 0.6$ and $\hat{v}_{1,2} = 1$. Thus, from step 1, we will have $\hat{v}_{1,2} \geq 1$. In addition, it s easy proved that with the stepsize selection, we always have $\hat{v}_{1,1} < 1$, in fact, it will not even reach 0.6. Thus, in the later iterations, the gradient of losses on node 1 and 2 will be scaled differently. This scaling is equivalent running gradient descent on a objective where the losses of the two nodes are scaled by different factors. In such a case, the algorithm will converge to a stationary point of a weighted average of the loss on node 1. Since the weight of the losses on the two nodes are different and the problem is a quadratic problem with only one minimizer and the unbalanced weights on the two functions yields a different minimizer, the algorithm will not converge to the unique stationary point of the original loss (which is $x = 0.5$). □

Theorem 1 says that though DADAM is proven to satisfy some regret bounds [17], it can fail to converge to stationary points in the nonconvex offline setting, which a common setting for training neural nets. We conjecture that this inconsistency is due to the definition of the regret in Nazari et al. [17]. In the next section, we will design decentralized adaptive gradient methods that are guaranteed to converge to stationary points.

4 Convergent decentralized adaptive gradient methods

In this section, we will discuss difficulties of designing adaptive gradient methods in decentralized optimization and introduce an algorithmic framework that will convert existing convergent adaptive gradient methods to their decentralized counterparts. By using the framework, we proposed the first convergent decentralized adaptive gradient method, converted from AMSGrad.

4.1 Importance and difficulties of consensus on adaptive learning rates

The divergent example in the previous section implies that we should synchronize the adaptive learning rates on different nodes. This can be easy to achieve in the parameter server setting where all the nodes are sending their gradients to a parameter server at each iteration. The parameter server can use the received gradients to maintain a sequence of synchronized adaptive learning rates when updating the parameters. However, in the situation of decentralized training, every node can only communication with its neighbors and a parameter server does not exist. Since every node can only communicate with its neighbors, the information for updating the adaptive learning rates can be only shared locally instead of broadcasted over the whole network, this make it impossible to obtain an synchronized updated adaptive learning rate in a single iteration using all the information in the network.

One way to solve this problem is to design communication protocols to give each node access to the same aggregated gradients over the whole network at least periodically if not at every iteration, so that the nodes can update their individual adaptive learning rates based on the same information to generate a synchronized sequence of adaptive learning rates. However, such a solution will introduce a significant amount of extra communication cost since it involves broadcasting over the network. Also, this is more of a system level solution instead of a algorithmic level solution.

Another way to solve this problem is by letting the sequences of adaptive learning rates on different nodes consent gradually, as the number of iteration grows. Intuitively, if the adaptive learning rates can consent fast enough, the difference among the adaptive learning rates on different nodes will not affect the convergence of the algorithm. The benefit of such an approach is that we do not need to introduce too much extra communication cost and as we will show later, it will produce an framework that automatically convert existing adaptive gradient methods to their decentralized counterparts. Yet, the benefits do not come for free. One need to design a way to ensure consensus of adaptive learning rates and this procedure should have a relatively low cost and be easy to implement. More importantly, such a design will further complicates the already convoluted convergence analysis of adaptive gradient methods.

In the next section, we will introduce our designed algorithmic framework based on this approach and its theoretical guarantee.

4.2 On decentralized adaptive gradient methods

As mentioned before, we need to choose a method to implement consensus of adaptive learning rates and there are many ways to do this. While each node can have different $\hat{v}_{t,i}$ in DADAM, one can keep track of the min/max/average of these adaptive learning rates and use the tracked quantity to update the adaptive learning rates. Also one can predefined some convergent lower and upper bounds to gradually synchronize the adaptive learning rates on different nodes like what the authors did for AdaBound [15]. We choose to use average consensus on $\hat{v}_{t,i}$ because in adaptive gradient methods such as AdaGrad and Adam, $\hat{v}_{t,i}$ approximate the second moment of gradient estimator, the average of estimations of second moments from different nodes is an estimation of second moment on the whole network. Also, this design will not introduce any extra tunable parameters that will complicates the parameter tuning process.

Theorem 2 presents the convergence guarantee of Algorithm 2.

Theorem 2. *Assume $\|g_{t,i}\|_\infty \leq G_\infty$, $\|\nabla f_i(x)\|_\infty \leq G_\infty$ and set $\alpha = 1/\sqrt{Td}$. When $\alpha \leq \frac{\epsilon^{0.5}}{16L}$, Algorithm 2 yields the following regret bound*

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

Algorithm 2 decentralized adaptive gradient method (with N nodes)

```
**Input:** learning rate $\alpha$, initial point $x_{1,i} = x_{init}, u_{\frac{1}{2},i} = \hat{v}_{0,i}, m_{0,i} = 0, \forall i$, mixing matrix $W$
**for** $t = 1, 2, ..., T$ **do**
    $g_{t,i} \leftarrow \nabla f_i(x_{t,i}) + \xi_{t,i}$
    $m_{t,i} = \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}$
    $\hat{v}_{t,i} = r_t(g_{1,i}, ..., g_{t-1,i})$
    $x_{t+\frac{1}{2},i} = \sum_{j=1}^{N} W_{ij} x_{t,j}$
    $\tilde{u}_{t,i} = \sum_{j=1}^{N} W_{ij} \tilde{u}_{t-\frac{1}{2},j}$
    $u_{t,i} = \max(\tilde{u}_{t,i}, \epsilon)$
    $x_{t+1,i} = x_{t+\frac{1}{2},i} - \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}}$
    $\tilde{u}_{t+\frac{1}{2},i} = \tilde{u}_{t,i} - \hat{v}_{t-1,i} + \hat{v}_{t,i}$
**end for**
```

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

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

which are constants independent of d*,* T *and* N*.*

Proof: The proof can be found in Appendix A.1.

Remark: From the theorem, it can be seen that if $\mathbb{E}\left[\sum_{t=1}^{T} \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}\right] = o(T)$ and $\bar{U}_t$ is upper bounded, the algorithm is guaranteed to converge to stationary points. Intuitively, this says that if the adaptive learning rates on different nodes do not change too fast, the algorithm can converge. This is intuitively true as in Chen et al. [5], it is shown that if such a condition is violated, an algorithm can diverge. Furthermore, the theorem shows the benefit of using more nodes. As N becomes larger, the term σ^2/N will be small, this is also justified by intuition that with the growth of N, the training process tends to be more stable.

In the following, we will present a notable special case of our algorithmic framework, decentralized AMSGrad, which is a decentralized variant of AMSGrad in our framework.

Compared with DADAM, the above algorithm uses a dynamic average consensus mechanism to keep track of average of $\{\hat{v}_{t,i}\}_{i=1}^{N}$, stored as $\tilde{u}_{t,i}$ on ith node, and uses $u_{t,i} = \max(\tilde{u}_{t,i}, \epsilon)$ for updating the adaptive learning rate for ith node. As the number of iteration grows, even though $\hat{v}_{t,i}$ on different nodes can converge to different constants, all the $u_{t,i}$ will be converge to the same number $\lim_{t\to\infty} \frac{1}{N} \sum_{i=1}^{N} \hat{v}_{t,i}$ if the limit exists. The use of this average consensus mechanism enables the consensus of adaptive learning rates on different nodes, which consequentially guarantees convergence 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 a convergent algorithm while DADAM is not.

The following theorem presents the convergent guarantee of Algorithm 3.

Algorithm 3 decentralized AMSGrad (with N nodes)

```
Input: learning rate $\alpha$, initial point $x_{1,i} = x_{init}, u_{\frac{1}{2},i} = \hat{v}_{0,i} = \epsilon\mathbf{1}$ (with $\epsilon \geq 0$), $m_{0,i} = 0, \forall i$, mixing matrix $W$
for $t = 1, 2, ..., T$ do
    $g_{t,i} \leftarrow \nabla f_i(x_{t,i}) + \xi_{t,i}$
    $m_{t,i} = \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}$
    $v_{t,i} = \beta_2 v_{t-1,i} + (1-\beta_2) g_{t,i}^2$
    $\hat{v}_{t,i} = \max(\hat{v}_{t-1,i}, v_{t,i})$
    $x_{t+\frac{1}{2},i} = \sum_{j=1}^N W_{ij} x_{t,j}$
    $\tilde{u}_{t,i} = \sum_{j=1}^N W_{ij} \tilde{u}_{t-\frac{1}{2},j}$
    $u_{t,i} = \max(\tilde{u}_{t,i}, \epsilon)$
    $x_{t+1,i} = x_{t+\frac{1}{2},i} - \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}}$
    $\tilde{u}_{t+\frac{1}{2},i} = \tilde{u}_{t,i} - \hat{v}_{t-1,i} + \hat{v}_{t,i}$
end for
```

Theorem 3. *Assume* $\|g_{t,i}\|_\infty \leq G_\infty$, $\|\nabla f_i(x)\|_\infty \leq G_\infty$ *and set* $\alpha = 1/\sqrt{Td}$. *When* $\alpha \leq \frac{\epsilon^{0.5}}{16L}$, *Algorithm 3 yields the following regret bound*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] \leq C_1'\frac{\sqrt{d}}{\sqrt{T}}\left(\mathbb{E}[f(Z_1)] - \min_z f(z)] + \frac{\sigma^2}{N}\right) + \frac{C_2'}{T} + \frac{d}{T}\sqrt{N}C_4' + \frac{\sqrt{d}}{T^{1.5}}\sqrt{N}C_5' \tag{5}$$

where

$$\begin{aligned} &C_1' = C_1,\ C_2' = C_2,\ C_3' = C_3, \\ &C_4' = C_4 G_\infty^2,\ C_5' = C_5 G_\infty^2 \end{aligned} \tag{6}$$

and C_1, C_2, C_3, C_4, C_5 *are constants independent of* d*,* T *and* N *defined in Theorem 2.*

Proof: See Appendix A.2

Remark: The above theorem says that Algorithm 3 converges with a rate of $O(\sqrt{d}/\sqrt{T})$ when T is large, which is the best known convergence rate under the given assumptions. Note that in some literature, SGD admits a convergence rate of $O(1/\sqrt{T})$ without any dimension dependency, such an improved convergence rate is under the assumption that the gradient estimator have bounded L_2 norm, which can hide a dimension dependency of $\sqrt{d}$ in the final convergence rate. One can

In the next section, we will present the proof sketch of Theorem 2 since the whole proof is complicated and the convergence analysis is one of our main contributions.

5 Experiments

In this section, we conduct experiments to test the performance of Algorithm 3 (decentralized AMSGrad) on both homogeneous data distribution and heterogeneous data distribution (i.e. the data generating distribution on different nodes are different). We compare it with DADAM and the decentralized stochastic gradient descent (DGD) [13]. The task is training a CNN with 3 convolution layers followed by a fully connected layer on MNIST. We set $\epsilon = 1e - 6$ for both decentralized AMSGrad and DADAM, the learning rate is chosen from [1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6] based on validation accuracy for all algorithms. In all the experiments, the graph contains 5 nodes and the nodes form a ring, each node can only talk with its two adjacent neighbors. We set $W_{ij} = 1/3$ if there nodes i and j are neighbors and $W_{ij} = 0$ otherwise for the mixing matrix. More details and experiments can be found in Appendix A.4.

6 Broader Impact Statement

We believe that our work stands in the line of several papers towards improving generalization and avoiding over-fitting. Indeed, the basic principle of our method is to fit any given model, in particular deep model, using an intermediate differentially-private mechanisms allowing the model to fit fresh samples while passing over the same batch of n observations. The impact of such work is straightforward and could avoid learning, and thus reproducing at testing phase, the bias existent in the training dataset.

References

[1] N. Agarwal, B. Bullins, X. Chen, E. Hazan, K. Singh, C. Zhang, and Y. Zhang. The case for full-matrix adaptive regularization. *arXiv preprint arXiv:1806.02958*, 2018.

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

[3] M. Assran, N. Loizou, N. Ballas, and M. Rabbat. Stochastic gradient push for distributed deep learning. *arXiv preprint arXiv:1811.10792*, 2018.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. 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.

[5] X. Chen, S. Liu, R. Sun, and M. Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv preprint arXiv:1808.02941*, 2018.

[6] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pages 571–582, 2014.

[7] P. Di Lorenzo and G. Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

[8] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[9] J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3): 592–606, 2011.

[10] M. Hong, D. Hajinezhad, and M.-M. Zhao. Prox-pda: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1529–1538. JMLR. org, 2017.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] X. Li and F. Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. *arXiv preprint arXiv:1805.08114*, 2018.

[13] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. 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.

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

[15] L. Luo, Y. Xiong, Y. Liu, and X. Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

[16] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[17] P. Nazari, D. A. Tarzanagh, and G. Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.

[18] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48, 2009.

[19] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[20] W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[21] S. U. Stich, J.-B. Cordonnier, and M. Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.

[22] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu. D^2: Decentralized training over decentralized data. *arXiv preprint arXiv:1803.07068*, 2018.

[23] H. Tang, X. Lian, T. Zhang, and J. Liu. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. *arXiv preprint arXiv:1905.05957*, 2019.

[24] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, pages 9850–9861, 2018.

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

[26] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes, from any initialization. *arXiv preprint arXiv:1806.01811*, 2018.

[27] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[28] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 9793–9803, 2018.

[29] F. Zou and L. Shen. On the convergence of weighted adagrad with momentum for training deep neural networks. *arXiv preprint arXiv:1808.03408*, 2018.

A Appendix

A.1 Proof of Theorem 2

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

$$Z_t = \bar{X}_t + \frac{\beta_1}{1-\beta_1}(\bar{X}_t - \bar{X}_{t-1}) \tag{7}$$

with $\bar{X}_0 \triangleq \bar{X}_1$.

Then we have the following Lemma to characterize the difference of iterations of sequence Z_t.

Lemma 1. *For the sequence defined in (7), we have*

$$Z_{t+1} - Z_t = \alpha \frac{\beta_1}{1-\beta_1} \frac{1}{N} \sum_{i=1}^{N} m_{t-1,i} \odot (\frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}}) - \alpha \frac{1}{N} \sum_{i=1}^{N} \frac{g_{t,i}}{\sqrt{u_{t,i}}} \tag{8}$$

Proof: See Appendix A.3. □

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 $G_1, G_2, ..., G_{t-1}$), we have

$$\mathbb{E}_{G_t|G_{1:t-1}} \left[\frac{1}{N} \sum_{i=1}^{N} \frac{g_{t,i}}{\sqrt{u_{t,i}}} \right] = \frac{1}{N} \sum_{i=1}^{N} \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \tag{9}$$

By assuming smoothness (A1) we have

$$f(Z_{t+1}) \le f(Z_t) + \langle \nabla f(Z_t), Z_{t+1} - Z_t \rangle + \frac{L}{2} \|Z_{t+1} - Z_t\|^2 \tag{10}$$

Substitute (8) into the above inequality and take expectation over G_t given $G_{1:t-1}$, we have

$$\begin{aligned} \mathbb{E}_{G_t|G_{1:t-1}}[f(Z_{t+1})] \le & f(Z_t) - \alpha \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^{N} \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\rangle + \frac{L}{2} \mathbb{E}_{G_t|G_{1:t-1}} \left[\|Z_{t+1} - Z_t\|^2 \right] \\ & + \alpha \frac{\beta_1}{1-\beta_1} \mathbb{E}_{G_t|G_{1:t-1}} \left[\left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^{N} m_{t-1,i} \odot (\frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}}) \right\rangle \right] \end{aligned} \tag{11}$$

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

$$\begin{aligned} \alpha \mathbb{E} \left[\left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^{N} \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\rangle \right] \le & \mathbb{E}[f(Z_t)] - \mathbb{E}[f(Z_{t+1})] + \frac{L}{2} \mathbb{E} \left[\|Z_{t+1} - Z_t\|^2 \right] \\ & + \alpha \frac{\beta_1}{1-\beta_1} \mathbb{E} \left[\left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^{N} m_{t-1,i} \odot (\frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}}) \right\rangle \right] \end{aligned} \tag{12}$$

In addition, we have

$$\begin{aligned} & \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^{N} \frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}} \right\rangle \\ = & \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^{N} \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\rangle + \left\langle \nabla f(Z_t), \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(x_{t,i}) \odot \left(\frac{1}{\sqrt{u_{t,i}}} - \frac{1}{\sqrt{\bar{U}_t}} \right) \right\rangle \end{aligned} \tag{13}$$

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

$$\begin{aligned}
&\left\langle \nabla f(Z_t), \frac{1}{N}\sum_{i=1}^{N} \frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}} \right\rangle\\
=&\frac{1}{2}\left\|\frac{\nabla f(Z_t)}{\bar{U}_t^{1/4}}\right\|^2 + \frac{1}{2}\left\|\frac{\frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_{t,i})}{\bar{U}_t^{1/4}}\right\|^2 - \frac{1}{2}\left\|\frac{\nabla f(Z_t)-\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})}{\bar{U}_t^{1/4}}\right\|^2\\
\geq&\frac{1}{4}\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 + \frac{1}{4}\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 - \frac{1}{2}\left\|\frac{\nabla f(Z_t)-\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})}{\bar{U}_t^{1/4}}\right\|^2\\
&-\frac{1}{2}\left\|\frac{\nabla f(Z_t)-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 - \frac{1}{2}\left\|\frac{\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\\
\geq&\frac{1}{2}\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 - \frac{3}{2}\left\|\frac{\nabla f(Z_t)-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 - \frac{3}{2}\left\|\frac{\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2
\end{aligned} \tag{14}$$

where the inequalities are all due to Cauchy-Schwartz.

Substituting (14) and (13) into (12), we get

$$\begin{aligned}
\frac{1}{2}\alpha\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] \leq&\mathbb{E}[f(Z_t)] - \mathbb{E}[f(Z_{t+1})] + \frac{L}{2}\mathbb{E}\left[\|Z_{t+1}-Z_t\|^2\right]\\
&+\alpha\frac{\beta_1}{1-\beta_1}\mathbb{E}\left[\left\langle \nabla f(Z_t), \frac{1}{N}\sum_{i=1}^{N} m_{t-1,i}\odot(\frac{1}{\sqrt{u_{t-1,i}}}-\frac{1}{\sqrt{u_{t,i}}})\right\rangle\right]\\
&-\alpha\mathbb{E}\left[\left\langle \nabla f(Z_t), \frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})\odot\left(\frac{1}{\sqrt{u_{t,i}}}-\frac{1}{\sqrt{\bar{U}_t}}\right)\right\rangle\right]\\
&+\frac{3}{2}\alpha\mathbb{E}\left[\left\|\frac{\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 + \left\|\frac{\nabla f(Z_t)-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right]
\end{aligned} \tag{15}$$

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

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] \leq&\frac{2}{T\alpha}(\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + \frac{L}{T\alpha}\sum_{t=1}^{T}\mathbb{E}\left[\|Z_{t+1}-Z_t\|^2\right]\\
&+\frac{2}{T}\frac{\beta_1}{1-\beta_1}\underbrace{\sum_{t=1}^{T}\mathbb{E}\left[\left\langle \nabla f(Z_t), \frac{1}{N}\sum_{i=1}^{N} m_{t-1,i}\odot(\frac{1}{\sqrt{u_{t-1,i}}}-\frac{1}{\sqrt{u_{t,i}}})\right\rangle\right]}_{T_1}\\
&+\underbrace{\frac{2}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\langle \nabla f(Z_t), \frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})\odot\left(\frac{1}{\sqrt{\bar{U}_t}}-\frac{1}{\sqrt{u_{t,i}}}\right)\right\rangle\right]}_{T_2}\\
&+\underbrace{\frac{3}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 + \left\|\frac{\nabla f(Z_t)-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right]}_{T_3}
\end{aligned} \tag{16}$$

Now we need to upper bound all the terms on RHS of the above inequality to get the convergence rate.

For terms in T_3 in (16), we can upper bound them by

$$\left\|\frac{\nabla f(Z_t)-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 \le \frac{1}{\min_{j\in[d]}[\bar{U}_t^{1/2}]_j}\left\|\nabla f(Z_t)-\nabla f(\bar{X}_t)\right\|^2 \le L\frac{1}{\min_{j\in[d]}[\bar{U}_t^{1/2}]_j}\underbrace{\left\|Z_t-\bar{X}_t\right\|^2}_{T_4} \tag{17}$$

and

$$\begin{aligned}\left\|\frac{\frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{t,i})-\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2 \le& \frac{1}{\min_{j\in[d]}[\bar{U}_t^{1/2}]_j}\frac{1}{N}\sum_{i=1}^N\left\|\nabla f_i(x_{t,i})-\nabla f(\bar{X}_t)\right\|^2\\ \le& L\frac{1}{\min_{j\in[d]}[\bar{U}_t^{1/2}]_j}\frac{1}{N}\underbrace{\sum_{i=1}^N\left\|x_{t,i}-\bar{X}_t\right\|^2}_{T_5}\end{aligned} \tag{18}$$

using Jensen's inequality, Lipschitz continuity of f_i, and the fact that $f=\frac{1}{N}\sum_{i=1}^N f_i$. .

What we need to do next is to bound T_4 and T_5 and we will bound T_5 first.

Before we proceed into bounding T_5, we need some preparations. Let's recall the update rule of X_t, we have

$$X_t = X_{t-1}W-\alpha\frac{M_{t-1}}{\sqrt{U_{t-1}}} = X_1W^{t-1}-\alpha\sum_{k=0}^{t-2}\frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}}W^k \tag{19}$$

where we define $W^0=\mathbf{I}$.

Since W is a symmetric matrix, we can decompose it as $W=Q\Lambda Q^T$ where Q is a orthonormal matrix and Λ is a diagonal matrix whose diagonal elements correspond to eigenvalues of W in an descending order, i.e. $\Lambda_{ii}=\lambda_i$ with λ_i being ith largest eigenvalue of W. In addition, because W is a doubly stochastic matrix, we know $\lambda_1=1$ and $q_1=\frac{\mathbf{1}_N}{\sqrt{N}}$

With eigen-decomposition of W, we can rewrite T_5 as

$$\sum_{i=1}^N\left\|x_{t,i}-\bar{X}_t\right\|^2 = \|X_t-\bar{X}_t\mathbf{1}_N^T\|_F^2 = \|X_tQQ^T-X_t\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\|_F^2 = \sum_{l=2}^N\|X_tq_l\|^2 \tag{20}$$

In addition, we can rewrite (19) as

$$X_t = X_1W^{t-1}-\alpha\sum_{k=0}^{t-2}\frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}}W^k = X_1-\alpha\sum_{k=0}^{t-2}\frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}}Q\Lambda^kQ^T \tag{21}$$

where the last equality is because $x_{1,i}=x_{1,j},\forall i,j$ and thus $X_1W=X_1$.

Then we have when $l>1$,

$$X_tq_l = (X_1-\alpha\sum_{k=0}^{t-2}\frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}}Q\Lambda^kQ^T)q_l = -\alpha\sum_{k=0}^{t-2}\frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}}q_l\lambda_l^k \tag{22}$$

because Q is orthonormal and $X_1q_l=x_{1,1}\mathbf{1}_N^Tq_l=x_{1,1}\sqrt{N}q_1^Tq_l=0,\forall l\neq 1$.

Combining (20) and (22), we have

$$T_5=\sum_{i=1}^N\left\|x_{t,i}-\bar{X}_t\right\|^2=\sum_{l=2}^N\|X_tq_l\|^2=\sum_{l=2}^N\alpha^2\left\|\sum_{k=0}^{t-2}\frac{M_{t-k-1}}{\sqrt{U_{t-k-1}}}\lambda_l^kq_l\right\|^2\le\alpha^2\left(\frac{1}{1-\lambda}\right)^2NdG_\infty^2\frac{1}{\epsilon} \tag{23}$$

where the last inequality follows from the fact that $g_{t,i}\le G_\infty$, $\|q_l\|=1$, and $|\lambda_l|\le\lambda<1$.

Now let us turn to T_4, it can be rewritten as

$$\left\|Z_t - \bar{X}_t\right\|^2 = \left\|\frac{\beta_1}{1-\beta_1}(\bar{X}_t - \bar{X}_{t-1})\right\|^2 = \left(\frac{\beta_1}{1-\beta_1}\right)^2 \alpha^2 \left\|\frac{1}{N}\sum_{i=1}^{N} \frac{m_{t-1,i}}{\sqrt{u_{t-1,i}}}\right\|^2 \le \left(\frac{\beta_1}{1-\beta_1}\right)^2 \alpha^2 d \frac{G_\infty^2}{\epsilon} \tag{24}$$

Now we know both T_4 and T_5 are in the order of $O(\alpha^2)$ and thus T_3 is in the order of $O(\alpha^2)$.

Next we will bound T_2 and T_1. Define $G_1 \triangleq \max_{t\in[T]} \max_{i\in[N]} \|\nabla f_i(x_{t,i})\|_\infty$, $G_2 \triangleq \max_{t\in[T]} \|\nabla f(Z_t)\|_\infty$, $G_3 \triangleq \max_{t\in[T]} \max_{i\in[N]} \|g_{t,i}\|_\infty$ and $G_\infty = \max(G_1, G_2, G_3)$

Then we have

$$\begin{aligned}
T_2 =& \sum_{t=1}^{T} \mathbb{E}\left[\left\langle \nabla f(Z_t), \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_{t,i}) \odot \left(\frac{1}{\sqrt{\bar{U}_t}} - \frac{1}{\sqrt{u_{t,i}}}\right)\right\rangle\right] \\
\le& \sum_{t=1}^{T} \mathbb{E}\left[G_\infty^2 \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\left|\frac{1}{\sqrt{[\bar{U}_t]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}}\right|\right] \\
=& \sum_{t=1}^{T} \mathbb{E}\left[G_\infty^2 \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\left|\frac{1}{\sqrt{[\bar{U}_t]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}}\right| \frac{\sqrt{[\bar{U}_t]_j} + \sqrt{[u_{t,i}]_j}}{\sqrt{[\bar{U}_t]_j} + \sqrt{[u_{t,i}]_j}}\right] \\
=& \sum_{t=1}^{T} \mathbb{E}\left[G_\infty^2 \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\left|\frac{[\bar{U}_t]_j - [u_{t,i}]_j}{[\bar{U}_t]_j\sqrt{[u_{t,i}]_j} + \sqrt{[\bar{U}_t]_j}[u_{t,i}]_j}\right|\right] \\
\le& \mathbb{E}\Bigg[\underbrace{\sum_{t=1}^{T} G_\infty^2 \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\left|\frac{[\bar{U}_t]_j - [u_{t,i}]_j}{2\epsilon^{1.5}}\right|}_{T_6}\Bigg]
\end{aligned} \tag{25}$$

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

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

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

where the second inequality is due to Lemma 4 and the fact that $U_t = \max(\tilde{U}_t, \epsilon)$ element-wisely.

Theorem 4. *Given a set of numbers $a_1, ..., a_n$ and denote their mean to be $\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$. In addition, define $b_i(r) \triangleq= \max(a_i, r)$ and $\bar{b}(r) = \frac{1}{n}\sum_{i=1}^{n} b_i(r)$. For any r and r' with $r' \ge r$ we have*

$$\sum_{i=1}^{n} |b_i(r) - \bar{b}(r)| \ge \sum_{i=1}^{n} |b_i(r') - \bar{b}(r')| \tag{26}$$

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

$$\sum_{i=1}^{n} |b_i(r) - \bar{b}(r)| = \sum_{i=1}^{n} |a_i - \bar{a}| \tag{27}$$

Proof: See Appendix A.3. □

Recall from update rule of U_t, by defining $\hat{V}_{-1} \triangleq \hat{V}_0$ and $U_0 \triangleq U_{1/2}$, we have $\forall t \ge 0$

$$\tilde{U}_{t+1} = (\tilde{U}_t - \hat{V}_{t-1} + \hat{V}_t)W \tag{28}$$

and thus

$$\tilde{U}_t = \tilde{U}_0 W^t + \sum_{k=1}^{t} (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) W^k = \tilde{U}_0 + \sum_{k=1}^{t} (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) Q \Lambda^k Q^T \tag{29}$$

Then we further have when $l \neq 1$,

$$\tilde{U}_t q_l = (\tilde{U}_0 + \sum_{k=1}^{t} (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) Q \Lambda^k Q^T) q_l = \sum_{k=1}^{t} (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) q_l \lambda_l^k \tag{30}$$

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 $q_1 = \frac{1}{\sqrt{N}} \mathbf{1}_N^T$) and $q_i^T q_j = 0$ when $i \neq j$.

Note by definition of $\|\cdot\|_{abs}$, we have $\forall A, B$, $\|A + B\|_{abs} \le \|A\|_{abs} + \|B\|_{abs}$, then we have

$$\begin{aligned}
T_6 \le & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \| - \sum_{l=2}^{N} \tilde{U}_t q_l q_l^T \|_{abs} \\
= & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \| - \sum_{k=1}^{t} (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) \sum_{l=2}^{N} q_l \lambda_l^k q_l^T \|_{abs} \\
\le & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^{t} \| (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) \sum_{l=2}^{N} q_l \lambda_l^k q_l^T \|_{abs} \\
= & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^{t} \sum_{j=1}^{d} \| \sum_{l=2}^{N} q_l \lambda_l^k q_l^T (-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j \|_1 \\
\le & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^{t} \sum_{j=1}^{d} \| \sum_{l=2}^{N} q_l \lambda_l^k q_l^T \|_1 \| (-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j \|_1 \\
\le & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^{t} \sum_{j=1}^{d} \sqrt{N} \| \sum_{l=2}^{N} q_l \lambda_l^k q_l^T \|_2 \| (-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j \|_1 \\
\le & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^{t} \sum_{j=1}^{d} \| (-\hat{V}_{t-1-k} + \hat{V}_{t-k})^T e_j \|_1 \sqrt{N} \lambda^k \\
= & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \sum_{k=1}^{t} \| (-\hat{V}_{t-1-k} + \hat{V}_{t-k}) \|_{abs} \sqrt{N} \lambda^k \\
= & \frac{G_\infty^2}{N} \sum_{t=1}^{T} \frac{1}{2\epsilon^{1.5}} \sum_{o=0}^{t-1} \| (-\hat{V}_{o-1} + \hat{V}_{o}) \|_{abs} \sqrt{N} \lambda^{t-o} \\
= & \frac{G_\infty^2}{N} \frac{1}{2\epsilon^{1.5}} \sum_{o=0}^{T-1} \sum_{t=o+1}^{T} \| (-\hat{V}_{o-1} + \hat{V}_{o}) \|_{abs} \sqrt{N} \lambda^{t-o} \\
\le & \frac{G_\infty^2}{\sqrt{N}} \frac{1}{2\epsilon^{1.5}} \sum_{o=0}^{T-1} \frac{\lambda}{1-\lambda} \| (-\hat{V}_{o-1} + \hat{V}_{o}) \|_{abs}
\end{aligned} \tag{31}$$

where $\lambda = \max(|\lambda_2|, |\lambda_N|)$.

Combining (25) and (31), we have

$$T_2 \leq \frac{G_\infty^2}{\sqrt{N}} \frac{1}{2\epsilon^{1.5}} \frac{\lambda}{1-\lambda} \mathbb{E}\left[\sum_{o=0}^{T-1} \|(-\hat{V}_{o-1} + \hat{V}_o)\|_{abs}\right] \tag{32}$$

Now we need to bound T_1, we have

$$\begin{aligned}
T_1 =& \sum_{t=1}^{T} \mathbb{E}\left[\left\langle \nabla f(Z_t), \frac{1}{N}\sum_{i=1}^{N} m_{t-1,i} \odot (\frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}}) \right\rangle\right] \\
\leq& \sum_{t=1}^{T} \mathbb{E}\left[G_\infty^2 \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d} \left| \frac{1}{\sqrt{[u_{t-1,i}]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}} \right|\right] \\
=& \sum_{t=1}^{T} \mathbb{E}\left[G_\infty^2 \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d} \left| \left(\frac{1}{\sqrt{[u_{t-1,i}]_j}} - \frac{1}{\sqrt{[u_{t,i}]_j}}\right) \frac{\sqrt{[u_{t,i}]_j} + \sqrt{[u_{t-1,i}]_j}}{\sqrt{[u_{t,i}]_j} + \sqrt{[u_{t-1,i}]_j}} \right|\right] \\
\leq& \sum_{t=1}^{T} \mathbb{E}\left[G_\infty^2 \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d} \left| \frac{1}{2\epsilon^{1.5}} ([u_{t-1,i}]_j - [u_{t,i}]_j) \right|\right] \\
\overset{(a)}{\leq}& \sum_{t=1}^{T} \mathbb{E}\left[G_\infty^2 \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d} \frac{1}{2\epsilon^{1.5}} |([\tilde{u}_{t-1,i}]_j - [\tilde{u}_{t,i}]_j)| \right] \\
=& G_\infty^2 \frac{1}{2\epsilon^{1.5}} \frac{1}{N} \mathbb{E}\left[\sum_{t=1}^{T} \|\tilde{U}_{t-1} - \tilde{U}_t\|_{abs}\right]
\end{aligned} \tag{33}$$

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

$$\begin{aligned}
&\sum_{t=1}^{T}\|\tilde{U}_{t-1}-\tilde{U}_t\|_{abs}\\
=&\sum_{t=1}^{T}\|\tilde{U}_{t-1}-(\tilde{U}_{t-1}-\hat{V}_{t-2}+\hat{V}_{t-1})W\|_{abs}\\
=&\sum_{t=1}^{T}\|\tilde{U}_{t-1}(I-W)+(-\hat{V}_{t-2}+\hat{V}_{t-1})W\|_{abs}\\
=&\sum_{t=1}^{T}\|\tilde{U}_{t-1}(QQ^T-Q\Lambda Q^T)+(-\hat{V}_{t-2}+\hat{V}_{t-1})W\|_{abs}\\
=&\sum_{t=1}^{T}\|\tilde{U}_{t-1}(\sum_{l=2}^{N}q_l(1-\lambda_l)q_l^T)+(-\hat{V}_{t-2}+\hat{V}_{t-1})W\|_{abs}\\
\leq&\sum_{t=1}^{T}\|\sum_{k=1}^{t-1}(-\hat{V}_{t-2-k}+\hat{V}_{t-1-k})\sum_{l=2}^{N}q_l\lambda_l^k(1-\lambda_l)q_l^T\|_{abs}+\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})W\|_{abs}\\
\leq&\sum_{t=1}^{T}\left(\sum_{k=1}^{t-1}\|-\hat{V}_{t-2-k}+\hat{V}_{t-1-k}\|_{abs}\sqrt{N}\lambda^k\right)+\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\\
=&\sum_{t=1}^{T}\left(\sum_{o=1}^{t-1}\|-\hat{V}_{o-2}+\hat{V}_{o-1}\|_{abs}\sqrt{N}\lambda^{t-o}\right)+\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\\
=&\sum_{o=1}^{T-1}\sum_{t=o+1}^{T}\left(\|-\hat{V}_{o-2}+\hat{V}_{o-1}\|_{abs}\sqrt{N}\lambda^{t-o}\right)+\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\\
\leq&\sum_{o=1}^{T-1}\frac{\lambda}{1-\lambda}\left(\|-\hat{V}_{o-2}+\hat{V}_{o-1}\|_{abs}\sqrt{N}\right)+\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\\
\leq&\frac{1}{1-\lambda}\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\sqrt{N}
\end{aligned}\tag{34}$$

Combining (33) and (34), we have

$$T_1\leq G_\infty^2\frac{1}{2\epsilon^{1.5}}\frac{1}{N}\mathbb{E}\left[\frac{1}{1-\lambda}\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\sqrt{N}\right]\tag{35}$$

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

$$
\begin{aligned}
&\|Z_{t+1}-Z_t\|^2\\
=&\left\|\alpha\frac{\beta_1}{1-\beta_1}\frac{1}{N}\sum_{i=1}^N m_{t-1,i}\odot(\frac{1}{\sqrt{u_{t-1,i}}}-\frac{1}{\sqrt{u_{t,i}}})-\alpha\frac{1}{N}\sum_{i=1}^N\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\\
\le&2\alpha^2\left\|\frac{\beta_1}{1-\beta_1}\frac{1}{N}\sum_{i=1}^N m_{t-1,i}\odot(\frac{1}{\sqrt{u_{t-1,i}}}-\frac{1}{\sqrt{u_{t,i}}})\right\|^2+2\alpha^2\left\|\frac{1}{N}\sum_{i=1}^N\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\\
\le&2\alpha^2\left(\frac{\beta_1}{1-\beta_1}\right)^2G_\infty^2\frac{1}{N}\sum_{i=1}^N\sum_{j=1}^d\frac{1}{\sqrt{\epsilon}}\left|\frac{1}{\sqrt{[u_{t-1,i}]_j}}-\frac{1}{\sqrt{[u_{t,i}]_j}}\right|+2\alpha^2\left\|\frac{1}{N}\sum_{i=1}^N\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\\
\le&2\alpha^2\left(\frac{\beta_1}{1-\beta_1}\right)^2G_\infty^2\frac{1}{N}\sum_{i=1}^N\sum_{j=1}^d\frac{1}{\sqrt{\epsilon}}\left|\frac{[u_{t,i}]_j-[u_{t-1,i}]_j}{2\epsilon^{1.5}}\right|+2\alpha^2\left\|\frac{1}{N}\sum_{i=1}^N\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\\
\le&2\alpha^2\left(\frac{\beta_1}{1-\beta_1}\right)^2G_\infty^2\frac{1}{N}\sum_{i=1}^N\sum_{j=1}^d\frac{1}{2\epsilon^2}\left|[\tilde{u}_{t,i}]_j-[\tilde{u}_{t-1,i}]_j\right|+2\alpha^2\left\|\frac{1}{N}\sum_{i=1}^N\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\\
=&2\alpha^2\left(\frac{\beta_1}{1-\beta_1}\right)^2G_\infty^2\frac{1}{N}\frac{1}{2\epsilon^2}\|\tilde{U}_t-\tilde{U}_{t-1}\|_{abs}+2\alpha^2\left\|\frac{1}{N}\sum_{i=1}^N\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2
\end{aligned}
\tag{36}
$$

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

$$
\begin{aligned}
&\sum_{t=1}^T\mathbb{E}[\|Z_{t+1}-Z_t\|^2]\\
\le&2\alpha^2\left(\frac{\beta_1}{1-\beta_1}\right)^2G_\infty^2\frac{1}{N}\frac{1}{2\epsilon^2}\mathbb{E}\left[\sum_{t=1}^T\|\tilde{U}_t-\tilde{U}_{t-1}\|_{abs}\right]+2\alpha^2\sum_{t=1}^T\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\right]\\
\le&\alpha^2\left(\frac{\beta_1}{1-\beta_1}\right)^2\frac{G_\infty^2}{\sqrt{N}}\frac{1}{\epsilon^2}\frac{1}{1-\lambda}\mathbb{E}\left[\sum_{t=1}^T\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\right]+2\alpha^2\sum_{t=1}^T\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\right]
\end{aligned}
\tag{37}
$$

where the last inequality is due to (34).

Now let's bound the last term on RHS of the above inequality. A trivial bound can be

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

due to $\|g_{t,i}\|\le G_\infty$ and $[u_{t,i}]_j\ge\epsilon,\forall j$ (this is easy to verify from update rule of $u_{t,i}$ and the assumption that $[v_{t,i}]_j\ge\epsilon,\forall i$). 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

$$\begin{aligned}
&\mathbb{E}_{G_t|G_{1:t-1}}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\right]\\
=&\mathbb{E}_{G_t|G_{1:t-1}}\left[\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\langle\frac{\nabla f_i(x_{t,i})+\xi_{t,i}}{\sqrt{u_{t,i}}},\frac{\nabla f_j(x_{t,j})+\xi_{t,j}}{\sqrt{u_{t,j}}}\right\rangle\right]\\
\overset{(a)}{=}&\mathbb{E}_{G_t|G_{1:t-1}}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}}\right\|^2\right]+\mathbb{E}_{G_t|G_{1:t-1}}\left[\frac{1}{N^2}\sum_{i=1}^{N}\left\|\frac{\xi_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\right]\\
\overset{(b)}{=}&\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}}\right\|^2+\frac{1}{N^2}\sum_{i=1}^{N}\sum_{l=1}^{d}\frac{\mathbb{E}_{G_t|G_{1:t-1}}[[\xi_{t,i}]_l^2]}{[u_{t,i}]_l}\\
\overset{(c)}{\le}&\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}}\right\|^2+\frac{d}{N}\frac{\sigma^2}{\epsilon}
\end{aligned}\tag{38}$$

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

Then we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\right]=&\mathbb{E}_{G_{1:t-1}}\left[\mathbb{E}_{G_t|G_{1:t-1}}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\right]\right]\\
\le&\mathbb{E}_{G_{1:t-1}}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}}\right\|^2+\frac{d}{N}\frac{\sigma^2}{\epsilon}\right]\\
=&\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}}\right\|^2\right]+\frac{d}{N}\frac{\sigma^2}{\epsilon}
\end{aligned}\tag{39}$$

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

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}}\right\|^2\right]=&\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}+\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})\odot\left(\frac{1}{\sqrt{u_{t,i}}}-\frac{1}{\sqrt{\bar{U}_t}}\right)\right\|^2\right]\\
\le&2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}\right\|^2\right]+2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_{t,i})\odot\left(\frac{1}{\sqrt{u_{t,i}}}-\frac{1}{\sqrt{\bar{U}_t}}\right)\right\|^2\right]\\
\le&2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}\right\|^2\right]+2\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\left\|\nabla f_i(x_{t,i})\odot\left(\frac{1}{\sqrt{u_{t,i}}}-\frac{1}{\sqrt{\bar{U}_t}}\right)\right\|^2\right]\\
\le&2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}\right\|^2\right]+2\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}G_\infty^2\frac{1}{\sqrt{\epsilon}}\left\|\frac{1}{\sqrt{u_{t,i}}}-\frac{1}{\sqrt{\bar{U}_t}}\right\|_1\right]
\end{aligned}\tag{40}$$

Summing over T, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{u_{t,i}}}\right\|^2\right]\\
\leq& 2\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}\right\|^2\right]+2\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}G_\infty^2\frac{1}{\sqrt{\epsilon}}\left\|\frac{1}{\sqrt{u_{t,i}}}-\frac{1}{\sqrt{\bar{U}_t}}\right\|_1\right]
\end{aligned}\tag{41}$$

For the last term on RHS of (41), we can bound it similarly as what we did for T_2 from (25) to (31), which yields

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}G_\infty^2\frac{1}{\sqrt{\epsilon}}\left\|\frac{1}{\sqrt{u_{t,i}}}-\frac{1}{\sqrt{\bar{U}_t}}\right\|_1\right]\\
\leq&\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}G_\infty^2\frac{1}{\sqrt{\epsilon}}\frac{1}{2\epsilon^{1.5}}\left\|u_{t,i}-\bar{U}_t\right\|_1\right]\\
=&\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{N}G_\infty^2\frac{1}{2\epsilon^2}\left\|\bar{U}_t\mathbf{1}^T-U_t\right\|_{abs}\right]\\
\leq&\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{N}G_\infty^2\frac{1}{2\epsilon^2}\|-\sum_{l=2}^{N}\tilde{U}_t q_l q_l^T\|_{abs}\right]\\
\leq&\frac{1}{\sqrt{N}}G_\infty^2\frac{1}{2\epsilon^2}\mathbb{E}\left[\sum_{o=0}^{T-1}\frac{\lambda}{1-\lambda}\|(-\hat{V}_{o-1}+\hat{V}_o)\|_{abs}\right]
\end{aligned}\tag{42}$$

Further, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}\right\|^2\right]\\
\leq& 2\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(\bar{X}_t)}{\sqrt{\bar{U}_t}}\right\|^2\right]+2\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(\bar{X}_t)-\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}\right\|^2\right]\\
=& 2\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\sqrt{\bar{U}_t}}\right\|^2\right]+2\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(\bar{X}_t)-\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}\right\|^2\right]
\end{aligned}\tag{43}$$

and the last term on RHS of the above inequality can be bounded following similar procedures from (18) to (23), as what we did for T_3. Completing the procedures yields

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{\nabla f_i(\bar{X}_t)-\nabla f_i(x_{t,i})}{\sqrt{\bar{U}_t}}\right\|^2\right]\\
\leq&\sum_{t=1}^{T}\mathbb{E}\left[L\frac{1}{\epsilon}\frac{1}{N}\sum_{i=1}^{N}\left\|x_{t,i}-\bar{X}_t\right\|^2\right]\\
\leq&\sum_{t=1}^{T}\mathbb{E}\left[L\frac{1}{\epsilon}\frac{1}{N}\alpha^2\left(\frac{1}{1-\lambda}\right)NdG_\infty^2\frac{1}{\epsilon}\right]\\
=&TL\frac{1}{\epsilon^2}\alpha^2\left(\frac{1}{1-\lambda}\right)dG_\infty^2
\end{aligned}\tag{44}$$

Finally, combining (39) to (44), we get

$$
\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\frac{g_{t,i}}{\sqrt{u_{t,i}}}\right\|^2\right] \\
\le& 4\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\sqrt{\bar{U}_t}}\right\|^2\right] + 4TL\frac{1}{\epsilon^2}\alpha^2\left(\frac{1}{1-\lambda}\right)dG_\infty^2 \\
&+ 2\frac{1}{\sqrt{N}}G_\infty^2\frac{1}{2\epsilon^2}\mathbb{E}\left[\sum_{o=0}^{T-1}\frac{\lambda}{1-\lambda}\|(-\hat{V}_{o-1}+\hat{V}_o)\|_{abs}\right] + T\frac{d}{N}\frac{\sigma^2}{\epsilon} \\
\le& 4\frac{1}{\sqrt{\epsilon}}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] + 4TL\frac{1}{\epsilon^2}\alpha^2\left(\frac{1}{1-\lambda}\right)dG_\infty^2 \\
&+ 2\frac{1}{\sqrt{N}}G_\infty^2\frac{1}{2\epsilon^2}\mathbb{E}\left[\sum_{o=0}^{T-1}\frac{\lambda}{1-\lambda}\|(-\hat{V}_{o-1}+\hat{V}_o)\|_{abs}\right] + T\frac{d}{N}\frac{\sigma^2}{\epsilon}.
\end{aligned}
\tag{45}
$$

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

Combining all above, we can have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] \\
\le&\frac{2}{T\alpha}(\mathbb{E}[f(Z_1)]-\mathbb{E}[f(Z_{T+1})]) \\
&+\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}\left[\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\right] \\
&+\frac{8L}{T}\alpha\frac{1}{\sqrt{\epsilon}}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] + 8L^2\alpha\frac{1}{\epsilon^2}\alpha^2\left(\frac{1}{1-\lambda}\right)dG_\infty^2 \\
&+\frac{4L}{T}\alpha\frac{1}{\sqrt{N}}G_\infty^2\frac{1}{2\epsilon^2}\mathbb{E}\left[\sum_{o=0}^{T-1}\frac{\lambda}{1-\lambda}\|(-\hat{V}_{o-1}+\hat{V}_o)\|_{abs}\right] + 2L\alpha\frac{d}{N}\frac{\sigma^2}{\epsilon} \\
&+\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}\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\right] \\
&+\frac{2}{T}\frac{G_\infty^2}{\sqrt{N}}\frac{1}{2\epsilon^{1.5}}\frac{\lambda}{1-\lambda}\mathbb{E}\left[\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\right] \\
&+\frac{3}{T}\left(\sum_{t=1}^{T}L\left(\frac{1}{1-\lambda}\right)^2\alpha^2dG_\infty^2\frac{1}{\epsilon^{1.5}}+\sum_{t=1}^{T}L\left(\frac{\beta_1}{1-\beta_1}\right)^2\alpha^2d\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^2d\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^3L^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}\left[\sum_{t=1}^{T}\|(-\hat{V}_{t-2}+\hat{V}_{t-1})\|_{abs}\right].
\end{aligned}
\tag{46}
$$

Set $\alpha = \frac{1}{\sqrt{dT}}$ and when $\alpha \leq \frac{\epsilon^{0.5}}{16L}$, we further have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] \\
\leq&\frac{4}{T\alpha}(\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + 4L\alpha\frac{d}{N}\frac{\sigma^2}{\epsilon} \\
&+ 6\alpha^2 d\left(\left(\frac{\beta_1}{1-\beta_1}\right)^2 + \left(\frac{1}{1-\lambda}\right)^2\right)L\frac{G_\infty^2}{\epsilon^{1.5}} + 16\alpha^3 L^2\left(\frac{1}{1-\lambda}\right)d\frac{G_\infty^2}{\epsilon^2} \\
&+ \frac{2}{T\epsilon^{1.5}}\frac{G_\infty^2}{\sqrt{N}}\frac{1}{1-\lambda}\left(L\alpha\left(\frac{\beta_1}{1-\beta_1}\right)^2\frac{1}{\epsilon^{0.5}} + \lambda + \frac{\beta_1}{1-\beta_1} + 2L\alpha\frac{1}{\epsilon^{0.5}}\lambda\right)\mathbb{E}\left[\sum_{t=1}^{T}\|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}\right] \\
=&\frac{4\sqrt{d}}{\sqrt{T}}(\mathbb{E}[f(Z_1)] - \mathbb{E}[f(Z_{T+1})]) + 4L\frac{\sqrt{d}}{\sqrt{T}}\frac{1}{N}\frac{\sigma^2}{\epsilon} \\
&+ 6\frac{1}{T}\left(\left(\frac{\beta_1}{1-\beta_1}\right)^2 + \left(\frac{1}{1-\lambda}\right)^2\right)L\frac{G_\infty^2}{\epsilon^{1.5}} + 16\frac{1}{T^{1.5}d^{0.5}}L^2\left(\frac{1}{1-\lambda}\right)\frac{G_\infty^2}{\epsilon^2} \\
&+ \frac{2}{T\epsilon^{1.5}}\frac{G_\infty^2}{\sqrt{N}}\frac{1}{1-\lambda}\left(\frac{L}{\sqrt{Td}}\left(\frac{\beta_1}{1-\beta_1}\right)^2\frac{1}{\epsilon^{0.5}} + \lambda + \frac{\beta_1}{1-\beta_1} + 2\frac{L}{\sqrt{Td}}\frac{1}{\epsilon^{0.5}}\lambda\right)\mathbb{E}\left[\sum_{t=1}^{T}\|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}\right] \\
\leq&C_1\frac{\sqrt{d}}{\sqrt{T}}\left(\mathbb{E}[f(Z_1)] - \min_z f(z) + \frac{\sigma^2}{N}\right) + \frac{1}{T}C_2 + \frac{1}{T^{1.5}d^{0.5}}C_3 \\
&+ \left(\frac{1}{TN^{0.5}}C_4 + \frac{1}{T^{1.5}d^{0.5}N^{0.5}}C_5\right)\mathbb{E}\left[\sum_{t=1}^{T}\|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}\right]
\end{aligned}
\tag{47}
$$

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 RHS of (46) to the LHS to cancel it using the assumption $8L\alpha\frac{1}{\sqrt{\epsilon}} \leq \frac{1}{2}$ followed by multiplying both sides by 2, and the constants introduced in the last step are defined as following

$$
\begin{aligned}
C_1 =& \max(4, 4L/\epsilon) \\
C_2 =& 6\left(\left(\frac{\beta_1}{1-\beta_1}\right)^2 + \left(\frac{1}{1-\lambda}\right)^2\right)L\frac{G_\infty^2}{\epsilon^{1.5}} \\
C_3 =& 16L^2\left(\frac{1}{1-\lambda}\right)\frac{G_\infty^2}{\epsilon^2} \\
C_4 =& \frac{2}{\epsilon^{1.5}}\frac{1}{1-\lambda}\left(\lambda + \frac{\beta_1}{1-\beta_1}\right)G_\infty^2 \\
C_5 =& \frac{2}{\epsilon^2}\frac{1}{1-\lambda}L\left(\frac{\beta_1}{1-\beta_1}\right)^2 G_\infty^2 + \frac{4}{\epsilon^2}\frac{\lambda}{1-\lambda}LG_\infty^2.
\end{aligned}
\tag{48}
$$

Substituting into $Z_1 = \bar{X}_1$ completes the proof □

A.2 Proof of Theorem 3

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

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] \leq& C_1\frac{\sqrt{d}}{\sqrt{T}}\left(\mathbb{E}[f(\bar{X}_1)] - \min_z f(z)] + \frac{\sigma^2}{N}\right) + \frac{1}{T}C_2 + \frac{1}{T^{1.5}d^{0.5}}C_3 \\
&+ \left(\frac{1}{TN^{0.5}}C_4 + \frac{1}{T^{1.5}d^{0.5}N^{0.5}}C_5\right)\mathbb{E}\left[\sum_{t=1}^{T}\|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}\right]
\end{aligned}
\tag{49}
$$

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

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

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}\right] \\
=&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{d} |-[\hat{v}_{t-2,i}]_j + [\hat{v}_{t-1,i}]_j|\right] \\
=&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{d} (-[\hat{v}_{t-2,i}]_j + [\hat{v}_{t-1,i}]_j)\right] \\
=&\mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{d} (-[\hat{v}_{-1,i}]_j + [\hat{v}_{T-1,i}]_j)\right] \\
=&\mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{d} (-[\hat{v}_{0,i}]_j + [\hat{v}_{T-1,i}]_j)\right]
\end{aligned} \tag{50}$$

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

Further, because $\|g_{t,i}\|_\infty \le G_\infty, \forall t, i$ and $v_{t,i}$ is a exponential moving average of $g_{k,i}^2, k = 1, 2, ..., t$, we know $|[v_{t,i}]_j| \le G_\infty^2, \forall t, i, j$. In addition, by update rule of $\hat{V}_t$, we also know each element of $\hat{V}_t$ also cannot be greater than G_∞^2, i.e. $|[\hat{v}_{t,i}]_j| \le G_\infty^2, \forall t, i, j$.

Given the fact that $[\hat{v}_{0,i}]_j \ge 0$, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \|(-\hat{V}_{t-2} + \hat{V}_{t-1})\|_{abs}\right] \\
=&\mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{d} (-[\hat{v}_{0,i}]_j + [\hat{v}_{T-1,i}]_j)\right] \\
\le&\mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{d} G_\infty^2\right] \\
=&NdG_\infty^2
\end{aligned} \tag{51}$$

Substituting the above into (49), we have

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\frac{\nabla f(\bar{X}_t)}{\bar{U}_t^{1/4}}\right\|^2\right] \le& C_1\frac{\sqrt{d}}{\sqrt{T}}\left(\mathbb{E}[f(\bar{X}_1)] - \min_z f(z) + \frac{\sigma^2}{N}\right) + \frac{1}{T}C_2 + \frac{1}{T^{1.5}d^{0.5}}C_3 \\
&+ \frac{d}{T}C_4\sqrt{N}G_\infty^2 + \frac{\sqrt{d}}{T^{1.5}}C_5\sqrt{N}G_\infty^2 \\
=&C_1'\frac{\sqrt{d}}{\sqrt{T}}\left(\mathbb{E}[f(\bar{X}_1)] - \min_z f(z) + \frac{\sigma^2}{N}\right) + \frac{1}{T}C_2' + \frac{1}{T^{1.5}d^{0.5}}C_3' \\
&+ \frac{d}{T}\sqrt{N}C_4' + \frac{\sqrt{d}}{T^{1.5}}\sqrt{N}C_5'
\end{aligned} \tag{52}$$

where we have

$$\begin{aligned}
C_1' &= C_1 \\
C_2' &= C_2 \\
C_3' &= C_3 \\
C_4' &= C_4 G_\infty^2 \\
C_5' &= C_5 G_\infty^2
\end{aligned} \tag{53}$$

The proof is complete. □

A.3 Proof of Lemmas

Lemma 1. *For the sequence defined in (7), we have*

$$Z_{t+1} - Z_t = \alpha \frac{\beta_1}{1-\beta_1} \frac{1}{N} \sum_{i=1}^{N} m_{t-1,i} \odot \left(\frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}}\right) - \alpha \frac{1}{N} \sum_{i=1}^{N} \frac{g_{t,i}}{\sqrt{u_{t,i}}} \tag{8}$$

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

$$\begin{aligned}
\bar{X}_{t+1} =& \frac{1}{N} \sum_{i=1}^{N} x_{t+1,i} \\
=& \frac{1}{N} \sum_{i=1}^{N} \left(x_{t+0.5,i} - \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}} \right) \\
=& \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=1}^{N} W_{ij} x_{t,j} - \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}} \right) \\
\overset{(i)}{=}& \left(\frac{1}{N} \sum_{j=1}^{N} x_{t,j} \right) - \frac{1}{N} \sum_{i=1}^{N} \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}} \\
=& \bar{X}_t - \frac{1}{N} \sum_{i=1}^{N} \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}}
\end{aligned} \tag{54}$$

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

Then, we have

$$\begin{aligned}
Z_{t+1} - Z_t =& \bar{X}_{t+1} - \bar{X}_t + \frac{\beta_1}{1-\beta_1}(\bar{X}_{t+1} - \bar{X}_t) - \frac{\beta_1}{1-\beta_1}(\bar{X}_{t+1} - \bar{X}_t) \\
=& \frac{1}{1-\beta_1}(\bar{X}_{t+1} - \bar{X}_t) - \frac{\beta_1}{1-\beta_1}(\bar{X}_{t+1} - \bar{X}_t) \\
=& \frac{1}{1-\beta_1} \left(-\frac{1}{N} \sum_{i=1}^{N} \alpha \frac{m_{t,i}}{\sqrt{u_{t,i}}} \right) - \frac{\beta_1}{1-\beta_1} \left(-\frac{1}{N} \sum_{i=1}^{N} \alpha \frac{m_{t-1,i}}{\sqrt{u_{t-1,i}}} \right) \\
=& \frac{1}{1-\beta_1} \left(-\frac{1}{N} \sum_{i=1}^{N} \alpha \frac{\beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}}{\sqrt{u_{t,i}}} \right) - \frac{\beta_1}{1-\beta_1} \left(-\frac{1}{N} \sum_{i=1}^{N} \alpha \frac{m_{t-1,i}}{\sqrt{u_{t-1,i}}} \right) \\
=& \alpha \frac{\beta_1}{1-\beta_1} \frac{1}{N} \sum_{i=1}^{N} m_{t-1,i} \odot \left(\frac{1}{\sqrt{u_{t-1,i}}} - \frac{1}{\sqrt{u_{t,i}}}\right) - \alpha \frac{1}{N} \sum_{i=1}^{N} \frac{g_{t,i}}{\sqrt{u_{t,i}}}
\end{aligned} \tag{55}$$

which is the desired result. □

Theorem 4. *Given a set of numbers $a_1, ..., a_n$ and denote their mean to be $\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$. In addition, define $b_i(r) \triangleq= \max(a_i, r)$ and $\bar{b}(r) = \frac{1}{n}\sum_{i=1}^{n} b_i(r)$. For any r and r' with $r' \geq r$ we* *have*

$$\sum_{i=1}^{n} |b_i(r) - \bar{b}(r)| \geq \sum_{i=1}^{n} |b_i(r') - \bar{b}(r')| \tag{26}$$

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

$$\sum_{i=1}^{n} |b_i(r) - \bar{b}(r)| = \sum_{i=1}^{n} |a_i - \bar{a}| \tag{27}$$

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

$$h(r) = \sum_{i=1}^{n} |b_i(r) - \bar{b}(r)| = \sum_{i=1}^{n} |\max(a_i, r) - \frac{1}{n} \sum_{j=1}^{n} \max(a_j, r)|, \tag{56}$$

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

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

$$h(r) = \sum_{i=1}^{l(r)} (\bar{b}(r) - r) + \sum_{i=l(r)+1}^{s(r)} (\bar{b}(r) - a_i) + \sum_{i=s(r)+1}^{n} (a_i - \bar{b}(r)). \tag{57}$$

and

$$\bar{b}(r) = \frac{1}{n} \left(l(r)r + \sum_{i=l(r)+1}^{n} a_i \right) \tag{58}$$

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

$$\begin{aligned} h'(r) =& l(r)(\frac{l(r)}{n} - 1) + (s(r) - l(r))\frac{l(r)}{n} - (n - s(r))\frac{l(r)}{n} \\ =& \frac{l(r)}{n}((l(r) - n) + (s(r) - l(r)) - (n - s(r))) \end{aligned} \tag{59}$$

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

$$h'(r) \leq 0 \tag{60}$$

which means $h(r)$ is non-increasing in each piece. Combining with the fact that $h(r)$ is continuous, (26) is proven.

When $r \leq a(i)$, we have $b(i) = \max(a_i, r) = r, \forall r \in [n]$ and $\bar{b}(r) = \frac{1}{n} \sum_{i=1}^{n} a_i = \bar{a}$ which proves (27). □

A.4 Additional experiments and details

In this section, we compare the learning curves of different algorithms 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 and there are no overlapping labels between different nodes. For all algorithms, we compare stepsizes in the set [1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6].