Distributed Adaptive Optimization with Gradient Compression

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper presents new algorithms – SPAMS and dist-SPAMS – for tackling single-machine and distributed optimization. Unlike prior works which rely on full gradient communication between the workers and the parameter-server, we design a distributed adaptive optimization method with gradient compression coupled with an error-feedback technique to alleviate the bias introduced by the compression. While the former permits to transmit fewer bits of gradient vectors to the server, we show that using the latter, which correct for the bias, our methods reach a stationary point in $\mathcal{O}(1/\sqrt{T})$ iterations, matching that of state-of-the-art single-machine and distributed methods, without any error-feedback. We illustrate our theoretical results by showing the effectiveness of our method both under the single-machine and distributed settings on various benchmark datasets.

1 Introduction

Deep neural network has achieved the state-of-the-art learning performance on numerous AI applications, e.g., computer vision [23, 26, 47], Natural Language Processing [25, 54, 58], Reinforcement Learning [37, 45] and recommendation systems [16, 49]. With the increasing size of both data and deep networks, standard single machine training confronts with at least two major challenges:

- Due to the limited computing power of a single machine, it would take a long time to process the massive number of data samples—training would be slow.
- In many practical scenarios, data are typically stored in multiple servers, possibly at different locations, due to the storage constraints (massive user behavior data, Internet images, etc.) or privacy reasons [11]. Transmitting data might be costly.

Distributed learning framework [18] has been a common training strategy to tackle the above two issues. For example, in centralized distributed stochastic gradient descent (SGD) protocol, data are located at n local nodes, at which the gradients of the model are computed in parallel. In each iteration, a central server aggregates the local gradients, updates the global model, and transmits back the updated model to the local nodes for subsequent gradient computation. As we can see, this setting naturally solves aforementioned issues: 1) We use n computing nodes to train the model, so the time per training epoch can be largely reduced; 2) There is no need to transmit the local data to central server. Besides, distributed training also provides stronger error tolerance since the training process could continue even one local machine breaks down. As a result of these advantages, there has been a surge of study and applications on distributed systems [10, 39, 20, 24, 27, 35, 33].

Among many optimization strategies, SGD is still the most popular prototype in distributed training for its simplicity and effectiveness [14, 1, 36]. Yet, when the deep learning model is very large, the communication between local nodes and central server could be expensive. Burdensome gradient transmission would slow down the whole training system, or even be impossible because of

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

the limited bandwidth in some applications. Thus, reducing the communication cost in distributed SGD has become an active topic, and an important ingredient of large-scale distributed systems (e.g. [42]). Solutions based on quantization, sparsification and other compression techniques of the local gradients are proposed, e.g., [4, 50, 48, 46, 3, 7, 17, 52, 28]. As one would expect, in most approaches, there exists a trade-off between compression and learning performance. In general, larger bias and variance of the compressed gradients usually bring more significant performance downgrade in terms of convergence [46, 2]. Interestingly, studies (e.g., [31]) show that the technique of *error feedback* can to a large extent remedy the issue of such biased compressors, achieving same convergence rate as full-gradient SGD.

On the other hand, in recent years, adaptive optimization algorithms (e.g. AdaGrad [21], Adam [32] and AMSGrad [41]) have become popular because of their superior empirical performance. These methods use different implicit learning rates for different coordinates that keep changing adaptively throughout the training process, based on the learning trajectory. In many learning problems, adaptive methods have been shown to converge faster than SGD, sometimes with better generalization as well. However, the body of literature that combines adaptive methods with distributed training is still very limited. Meanwhile, adopting gradient compression in adaptive methods has also been rarely considered in literature. In this paper, we fill the gap by considering communication-efficient distributed adaptive optimization.

1.1 Our Contributions

We develop a simple optimization leveraging the adaptivity of AMSGrad, and the computational virtue of **Top-k** sparsification, for tackling a large finite-sum of nonconvex objective functions.

Our technique is shown to be both theoretically and empirically effective under *the classical centralized setting* and *the distributed setting*.

In this contribution,

- We derive SPAMS, a distributed optimization method with gradient compression occurring at the worker level. Our scheme is coupled with a error-feedback technique to reduce the bias implied by the compression step.
- Throughout this paper, we provide single-machine and decentralized views of our method both on the empirical and theoretical levels. We exhibits the advantage of the compression and error-feedback steps within an adaptive optimization trajectory under those two settings.
- Under mild assumption, such as nonconvexity and smoothness, we provide a non-asymptotic convergence rate of SPAMS in the general case, *i.e.,* when the number of workers is equal to n and with unspecified values for the hyperparameters. Our theoretical analysis includes the special cases of single-machine setting ($n = 1$) and exhibits a linear speedup (linear in n) of our method in the particular case of $\beta_1 = 0$.
- We highlight the effectiveness of our compressed adaptive method through several numerical experiments for single-machine and distributed optimization tasks.

We review Section 2 the contributions to date, related to compression techniques in optimization, such as quantization and sparsification, and to error feedback technique. Then, we develop in Section 3, our method, namely SPAMS, based on the **Top-k** compression method using AMSGrad as a prototype optimization algorithm for our scheme. Theoretical understanding of our method's behaviour with respect to convergence towards a stationary point is developed in Section 4 under both the decentralized setting, *i.e.,* multiple workers which communicate with a central server, and the single machine setting. We present numerical illustrations showing the advantages of our method in Section 5.

2 Related Work

2.1 Distributed SGD with Compressed Gradients

Quantization. As we mentioned before, SGD is the most commonly adopted optimization method in distributed training of deep neural nets. To reduce the expensive communication in large-scale distributed systems, extensive works have considered various compression techniques applied to the gradient transaction procedure. The first strategy is quantization. [19] condenses 32-bit floating numbers into 8-bits when representing the gradients. [42, 7, 31, 8] use the extreme 1-bit information (sign) of the gradients, combined with tricks like momentum, majority vote and memory. Other quantization-based methods include QSGD [4, 51, 57] and LPC-SVRG [55], leveraging unbiased stochastic quantization. The saving in communication of quantization methods is moderate: for example, 8-bit quantization reduces the cost to 25% (compared with 32-bit full-precision). Even in the extreme 1-bit case, the largest compression ratio is around $1/32 \approx 3.1\%$.

Sparsification. Gradient sparsification is another popular solution which may provide higher compression rate. Instead of commuting the full gradient, each local worker only passes a few coordinates to the central server and zeros out the others. Thus, we can more freely choose higher compression ratio (e.g., 1%, 0.1%), still achieving impressive performance in many applications [34]. Stochastic sparsification methods, including uniform sampling and magnitude based sampling [48], select coordinates based on some sampling probability yielding unbiased gradient compressors. Deterministic methods are simpler, e.g., Random-k, Top-k [46, 44] (selecting k elements with largest magnitude), Deep Gradient Compression [34], but usually lead to biased gradient estimation. In [28], the central server identifies heavy-hitters from the count-sketch [12] of the local gradients, which can be regarded as a noisy variant of Top-k strategy. More applications and analysis of compressed distributed SGD can be found in [30, 43, 5, 6, 29], among others.

Error Feedback. Biased gradient estimation, which is a consequence of many aforementioned methods (e.g., signSGD, Top-k), undermines the model training, both theoretically and empirically, with slower convergence and worse generalization [2, 9]. The technique of *error feedback* is able to "correct for the bias" and fix the problems. In this procedure, the difference between the true stochastic gradient and the compressed one is accumulated locally, which is then added back to the local gradients in later iterations. [46, 31] prove the $\mathcal{O}(\frac{1}{T})$ and $\mathcal{O}(\frac{1}{\sqrt{T}})$ convergence rate of EF-SGD in strongly convex and non-convex setting respectively, matching the rates of vanilla SGD [40, 22].

2.2 Adaptive Optimization

In each SGD update, all the gradient coordinates share the same learning rate. This latter is either constant or decreasing through the iterations. Adaptive optimization methods cast different learning rate on each dimension. For instance, AdaGrad, developed in [21], divides the gradient element-wisely by $\sqrt{\sum_{t=1}^{T} g_t^2} \in \mathbb{R}^d$, where $g_t \in \mathbb{R}^d$ is the gradient vector at time t and d is the model dimensionality. Thus, it intrinsically assigns different learning rates to different coordinates throughout the training – elements with smaller previous gradient magnitude tend to move at larger rate via a larger steps. AdaGrad has been shown to perform well especially under some sparsity structure **BK**: sparsity in the model or the data or both?.

Other adaptive methods include AdaDelta [56] and Adam [32] which introduce momentum and moving average of second moment estimation into AdaGrad hence leading to better performances. AMSGrad [41] fixes the potential convergence issue of Adam, which will serve as the prototype in this paper. We present the pseudocode in Algorithm 1.

Algorithm 1 AMSGRAD optimization method

Input: parameter β_1, β_2, and η_t.
Initialize: $\theta_1 \in \Theta$ and $v_0 = \epsilon 1 \in \mathbb{R}^d$.
for $t = 1$ to T **do**
    Compute stochastic gradient g_t at θ_t.
    $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$.
    $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$.
    $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$.
    $\theta_{t+1} = \theta_t - \eta_t \frac{\theta_t}{\sqrt{\hat{v}_t}}$.
end for

In general, adaptive optimization methods are easier to tune in practice, and usually exhibit faster convergence than SGD. Thus, they have been widely used in training deep learning models in language and computer vision applications, e.g., [15, 53, 59]. In distributed setting, the work [38] proposes a decentralized system in online optimization. However, communication efficiency is not considered. The recent work [13] is the most relevant to our paper. Yet, their method is based on Adam, and requires every local node to store a local estimation of first and second moment, thus being less efficient. We will present more detailed comparison in Section 3.

3 Communication-Efficient Adaptive Optimization

Most modern machine learning tasks can be casted as a large finite-sum optimization problem written as:

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \tag{1}$$

where n denotes the number of workers, f_i represents the average loss (over the local data samples) for worker $i \in [\![n]\!]$ and θ the global model parameter taking value in Θ, a subset of $\mathbb{R}^d$.

3.1 Gradient Compressors

In this paper, we mainly consider deterministic q-deviate compressors defined as below.

Assumption 1. *The gradient compressor $\mathcal{C} : \mathbb{R}^d \mapsto \mathbb{R}^d$ is q-deviate: for $\forall x \in \mathbb{R}^d$, $\exists\, 0 \leq q < 1$ such that $\|\mathcal{C}(x) - x\| \leq q\,\|x\|$.*

Note that, smaller q indicates better approximation of the true gradient, and $q = 0$ implies no compression, i.e. $\mathcal{C}(x) = x$. We give two popular and highly efficient q-deviate compressors that will be compared in this paper.

Definition 1 (Top-k)**.** *For $x \in \mathbb{R}^d$, denote $\mathcal{S}$ as the size-k set of $i \in [d]$ with largest k magnitude $|x_i|$. The **Top-k** compressor is defined as $\mathcal{C}(x)_i = x_i$, if $i \in \mathcal{S}$; $\mathcal{C}(x)_i = 0$ otherwise.*

Definition 2 (Block-Sign)**.** *For $x \in \mathbb{R}^d$, define M blocks indexed by $\mathcal{B}_i$, $i = 1, ..., M$, with $d_i := |\mathcal{B}_i|$. The **Block-Sign** compressor is defined as $\mathcal{C}(x) = [sign(x_{\mathcal{B}_1})\frac{\|x_{\mathcal{B}_1}\|_1}{d_1}, ..., sign(x_{\mathcal{B}_M})\frac{\|x_{\mathcal{B}_M}\|_1}{d_M}]$.*

Remark 1. *It is well-known [46, 60] that for **Top-k**, $q^2 = 1 - \frac{k}{d}$; for **Block-Sign**, by Cauchy-Schwartz inequality we have $q^2 = 1 - \min_{i \in [M]} \frac{1}{d_i}$. **BK:** define $[M]$ and d_i*

The intuition of **Top-k** is that, it has been observed in many deep neural networks that during training, most gradients are typically very small and can be regarded as redundant—gradients with large magnitude contain most information. The **Block-Sign** compressor is a simple extension of the 1-bit **SIGN** compressor, adapted to different gradient magnitude in different blocks, which, for neural nets, are usually set as the distinct network layers. The scaling factor in Definition 2 is to preserve the (possibly very different) gradient magnitude in each layer. In principle, **Top-k** would perform the best when the gradient is sparse, or only has a few very large absolute values, while **Block-Sign** compressor would work well when most gradients have similar magnitude within each layer.

3.2 SPAMS with Error Feedback for Distributed Optimization

We present in Algorithm 2 our method based on a AMSGrad type of update in the central server and a compression coupled with an error computation on each worker.

The key difference of our **Top-k** based AMSGrad distributed optimization method compared with [13], developing a quantized variant of Adam [32] is that, in our method, only compressed gradients are transmitted from the workers to the central server. In [13], each worker keeps a local copy of the moment estimates commonly noted m and v, and compresses and transmits the ratio $\frac{m}{v}$ as a whole to the server. Thus, that method is very much like the sparsified distributed SGD, with the exception that the ratio $\frac{m}{v}$ plays the role of the gradient vector g communication-wise. In our optimization method in Algorithm 2, the moment estimates m and v are computed only at the central server, with the compressed version of the workers gradients instead of the full gradient. This constitutes be the key of our algorithm in convergence analysis and in practical benefits in the numerical runs.

Algorithm 2 Distributed SPAMS with error-feedback

1: **Input**: parameter β_1, β_2, learning rate η_t.
2: Initialize: central server parameter $\theta_1 \in \Theta \subseteq \mathbb{R}^d$; $e_{1,i} = 0$ the error accumulator for each worker; sparsity parameter k; n local workers; $m_0 = 0$, $v_0 = 0$, $\hat{v}_0 = 0$
3: **for** $t = 1$ to T **do**
4:     **parallel for worker** $i \in [n]$ **do**:
5:         Receive model parameter θ_t from central server
6:         Compute stochastic gradient $g_{t,i}$ at θ_t
7:         Compute $\tilde{g}_{t,i} = \textbf{Top-}k(g_{t,i} + e_{t,i}, k)$
8:         Update the error $e_{t+1,i} = e_{t,i} + g_{t,i} - \tilde{g}_{t,i}$
9:         Send $\tilde{g}_{t,i}$ back to central server
10:     **end parallel**
11:     **Central server do:**
12:     $\bar{g}_t = \frac{1}{n}\sum_{i=1}^n \tilde{g}_{t,i}$
13:     $m_t = \beta_1 m_{t-1} + (1-\beta_1)\bar{g}_t$
14:     $v_t = \beta_2 v_{t-1} + (1-\beta_2)\bar{g}_t^2$
15:     $\hat{v}_t = \max(v_t, \hat{v}_{t-1})$
16:     Update the global model $\theta_{t+1} = \theta_t - \eta_t \frac{m_t}{\sqrt{\hat{v}_t + \epsilon}}$
17: **end for**

4 Non-Asymptotic Convergence Analysis of SPAMS

In this section, we provide a finite time convergence result of our method, true for any termination iteration index T. We make the following assumptions.

Assumption 2. *(Smoothness) For $i \in [\![n]\!]$, f_i is L-smooth: $\|\nabla f_i(\theta) - \nabla f_i(\vartheta)\| \leq L\,\|\theta - \vartheta\|$.*

Assumption 3. *(Unbiased and Bounded gradient **per worker**) For any iteration index $t > 0$ and worker index $i \in [\![n]\!]$, the stochastic gradient is unbiased and bounded from above: $\mathbb{E}[g_{t,i}] = \nabla f_i(\theta_t)$ and $\|g_{t,i}\| \leq G_i$.*

Assumption 4. *(Bounded variance **per worker**) For any iteration index $t > 0$ and worker index $i \in [\![n]\!]$, the variance of the noisy gradient is bounded: $\mathbb{E}[|g_{t,i} - \nabla f_i(\theta_t)|^2] < \sigma_i^2$.*

Denote by $Q(\cdot)$ the quantization operator Line 7 of Algorithm 2, which takes as input a gradient vector and returns a quantized version of it, and note $\tilde{g} := Q(g)$. Assume that

4.1 General case convergence rate

We denote for all $\theta \in \Theta$, the following objective function:

$$f(\theta) := \frac{1}{n}\sum_{i=1}^n f_i(\theta)\,, \tag{2}$$

where n denotes the number of workers. In this paper, we are particularly interested in the case when the number of decentralized machines is large but we also provide theoretical and experimental insights on the single-machine case ($n = 1$).

We begin by considering the general case for Algorithm 2 when the number of worker can be large and the hyperparameters are unspecified. Under the mild assumption stated above, we derive the following convergence bound in the decentralized setting:

Theorem 1. *Denote $C_0 = \sqrt{\frac{4(1+q^2)^3}{(1-q^2)^2}G^2 + \epsilon}$, $C_1 = \frac{\beta_1}{1-\beta_1} + \frac{2q}{1-q^2}$. Under Assumption 1 to Assumption 4, with $\eta_t = \eta \leq \frac{\epsilon}{4LC_0}$, then for $T > 0$,* SPAMS *satisfies*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}[\|\nabla f(\theta_t)\|^2] \leq 2C_0\Big(\frac{\mathbb{E}[f(\theta_1) - f(\theta^*)]}{T\eta} + \frac{\eta L\sigma^2}{n\epsilon} + \frac{\eta^2 C_0 C_1^2 L G^2}{\epsilon^2} + \frac{\eta(1+C_1)G^2 d}{T\sqrt{\epsilon}} + \frac{\eta^2(1+2C_1)C_1 L G^2 d}{T\epsilon}\Big),$$

We remark from this bound in Theorem 1, that the more quantization we apply to our gradient vectors ($q \uparrow$), the larger the upper bound of the stationary condition is, *i.e.,* the slower the algorithm is. This is intuitive as using compressed quantities will definitely impact the algorithm speed. We will observe in the numerical section below that a trade-off on the level of quantization q can be found to achieve similar speed of convergence with less computation resources used throughout the training.

Corollary 1. *Under Assumption 2 to Assumption 4,setting the stepsize as $\eta_t = L\sqrt{\frac{n}{T}}$, the sequence of iterates $\{\theta_t\}_{t>0}$ output from Algorithm 2 satisfies:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2] \leq \mathcal{O}(\frac{1}{L\sqrt{nT}} + d\frac{L}{\sqrt{nT}} + \frac{1}{T} + cst.),$$

Additionally if $\beta_1 = 0$ we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2] \leq \mathcal{O}(\frac{1}{L\sqrt{nT}} + d\frac{L}{T}\sqrt{\frac{n}{T}} + \frac{1}{T}),$$

which exhibits the linear speedup of our method in the special case of $\beta_1 = 0$.

4.2 Extension to the single-machine setting

We first provide the formulation of our method in the single machine setting in Algorithm 3. Here, the data and the computation are all performed on a single machine.

Algorithm 3 SPAMS with error-feedback for a single machine

Input: parameter β_1, β_2, learning rate η_t.
Initialize: central server parameter $\theta_1 \in \Theta \subseteq \mathbb{R}^d$; $e_1 = 0$ the error accumulator; sparsity parameter k; $m_0 = 0$, $v_0 = 0$, $\hat{v}_0 = 0$
for $t = 1$ to T **do**
  Compute stochastic gradient $g_t = g_{t,i_t}$ at θ_t for randomly sampled index i_t
  Compute $\tilde{g}_t =$ **Top-**$k(g_t + e_t, k)$
  Update the error $e_{t+1} = e_t + g_t - \tilde{g}_t$
  $m_t = \beta_1 m_{t-1} + (1-\beta_1)\tilde{g}_t$
  $v_t = \beta_2 v_{t-1} + (1-\beta_2)\tilde{g}_t^2$
  $\hat{v}_t = \max(v_t, \hat{v}_{t-1})$
  Update the global model $\theta_{t+1} = \theta_t - \eta_t \frac{m_t}{\sqrt{\hat{v}_t + \epsilon}}$
end for

The convergence rate of the vector of parameters estimated via Algorithm 3 is given below:

Corollary 2. *Under Assumption 2 to Assumption 4,setting the stepsize as $\eta_t = L\sqrt{\frac{n}{T}}$, the sequence of iterates $\{\theta_t\}_{t>0}$ output from Algorithm 3 satisfies:*

complete with single machine corollary

5 Numerical Experiments

Our proposed **Top-k**-EF with AMSGrad matches that of full AMSGrad, in distributed learning. Number of local workers is 20. Error feedback fixes the convergence issue of using solely the **Top-k** gradient.

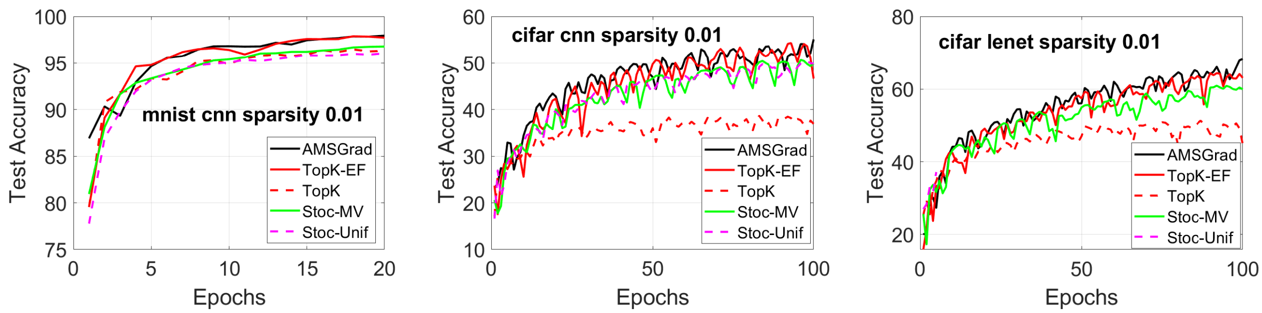

Figure 1: Test accuracy.

6 Conclusion

References

[1] Naman Agarwal, Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7575–7586, 2018.

[2] Ahmad Ajalloeian and Sebastian U Stich. Analysis of sgd with biased gradient estimators. *arXiv preprint arXiv:2008.00051*, 2020.

[3] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017.

[4] 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.

[5] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Sarit Khirirat, Nikola Konstantinov, and Cédric Renggli. The convergence of sparsified gradient methods. *arXiv preprint arXiv:1809.10505*, 2018.

[6] Debraj Basu, Deepesh Data, Can Karakus, and Suhas N. Diggavi. Qsparse-local-sgd: Distributed SGD with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14668–14679, 2019.

[7] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

[8] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[9] Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *CoRR*, abs/2002.12410, 2020.

[10] 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.

[11] Ken Chang, Niranjan Balachandar, Carson K. Lam, Darvin Yi, James M. Brown, Andrew Beers, Bruce R. Rosen, Daniel L. Rubin, and Jayashree Kalpathy-Cramer. Distributed deep learning networks among institutions for medical imaging. *J. Am. Medical Informatics Assoc.*, 25(8):945–954, 2018.

[12] Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, volume 2380 of *Lecture Notes in Computer Science*, pages 693–703. Springer, 2002.

[13] Congliang Chen, Li Shen, Haozhi Huang, Qi Wu, and Wei Liu. Quantized adam with error feedback. *arXiv preprint arXiv:2004.14180*, 2020.

[14] 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.

[15] Dami Choi, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. On empirical comparisons of optimizers for deep learning. *CoRR*, abs/1910.05446, 2019.

[16] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 191–198. ACM, 2016.

[17] Christopher De Sa, Matthew Feldman, Christopher Ré, and Kunle Olukotun. Understanding and optimizing asynchronous low-precision stochastic gradient descent. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 561–574, 2017.

[18] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew W. Senior, Paul A. Tucker, Ke Yang, and Andrew Y. Ng. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1232–1240, 2012.

[19] Tim Dettmers. 8-bit approximations for parallelism in deep learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[20] 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.

[21] John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 257–269, 2010.

[22] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[24] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

[25] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 6645–6649. IEEE, 2013.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[27] 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.

[28] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. Communication-efficient distributed SGD with sketching. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13144–13154, 2019.

[29] Jiawei Jiang, Fangcheng Fu, Tong Yang, and Bin Cui. Sketchml: Accelerating distributed machine learning with data sketches. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 1269–1284. ACM, 2018.

[30] Peng Jiang and Gagan Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2530–2541, 2018.

[31] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. *arXiv preprint arXiv:1901.09847*, 2019.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] 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.

[34] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[35] 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.

[36] Hiroaki Mikami, Hisahiro Suganuma, Yoshiki Tanaka, Yuichi Kageyama, et al. Massively distributed sgd: Imagenet/resnet-50 training in a flash. *arXiv preprint arXiv:1811.05233*, 2018.

[37] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[38] 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.

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

[40] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[41] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[42] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 1058–1062. ISCA, 2014.

[43] Zebang Shen, Aryan Mokhtari, Tengfei Zhou, Peilin Zhao, and Hui Qian. Towards more efficient stochastic decentralized learning: Faster convergence and sparse communication. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4631–4640. PMLR, 2018.

[44] Shaohuai Shi, Kaiyong Zhao, Qiang Wang, Zhenheng Tang, and Xiaowen Chu. A convergence analysis of distributed sgd with communication-efficient gradient sparsification. In *IJCAI*, pages 3411–3417, 2019.

[45] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nat.*, 550(7676):354–359, 2017.

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

[47] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios D. Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.*, 2018:7068349:1–7068349:13, 2018.

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

[49] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:29–39, 2017.

[50] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *arXiv preprint arXiv:1705.07878*, 2017.

[51] Jiaxiang Wu, Weidong Huang, Junzhou Huang, and Tong Zhang. Error compensated quantized SGD and its applications to large-scale distributed optimization. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5321–5329. PMLR, 2018.

[52] Guandao Yang, Tianyi Zhang, Polina Kirichenko, Junwen Bai, Andrew Gordon Wilson, and Chris De Sa. Swalp: Stochastic weight averaging in low precision training. In *International Conference on Machine Learning*, pages 7015–7024. PMLR, 2019.

[53] Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[54] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Comput. Intell. Mag.*, 13(3):55–75, 2018.

[55] Yue Yu, Jiaxiang Wu, and Junzhou Huang. Exploring fast and communication-efficient algorithms in large-scale distributed networks. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 674–683. PMLR, 2019.

[56] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[57] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. Zipml: Training linear models with end-to-end low precision, and a little bit of deep learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 4035–4043. PMLR, 2017.

[58] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(4), 2018.

[59] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. *CoRR*, abs/2006.05987, 2020.

[60] Shuai Zheng, Ziyue Huang, and James T. Kwok. Communication-efficient distributed block-wise momentum SGD with error-feedback. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11446–11456, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? **[TODO]**
   (b) Did you describe the limitations of your work? **[TODO]**
   (c) Did you discuss any potential negative societal impacts of your work? **[TODO]**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[TODO]**
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? **[TODO]**
   (b) Did you include complete proofs of all theoretical results? **[TODO]**
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)? **[TODO]**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[TODO]**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[TODO]**
   (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)? **[TODO]**
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? **[TODO]**
   (b) Did you mention the license of the assets? **[TODO]**
   (c) Did you include any new assets either in the supplemental material or as a URL? **[TODO]**
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[TODO]**
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[TODO]**
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? **[TODO]**
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **[TODO]**
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **[TODO]**

A Intermediary Lemmas

Lemma 1. *Under Assumption1 to Assumption4 we have:*

$$\mathbb{E}\|m_t'\|^2 \leq C\sigma^2 + C_1 \sum_{\tau=1}^{t} (\beta_1^2(2-\beta_1^2))^{t-\tau} \mathbb{E}[\|\nabla f(\theta_\tau)\|^2],$$

$$\mathbb{E}[\|m_t\|^2] \leq (3q^2 + \frac{4q^2(6q^2+3)}{(1-q^2)^2} + 1)C\sigma^2 + (6q^2+3)C_1 \sum_{\tau=1}^{t} (\beta_1^2(2-\beta_1^2))^{t-\tau} \mathbb{E}[\|\nabla f(\theta_\tau)\|^2],$$

where $C_1 = (1-\beta_1^2)(1+\frac{1}{4(1-\beta_1^2)})$ *and* $C = \frac{C_1}{1-\beta_1^2(2-\beta_1^2)}$.

Proof. We have by Young's inequality

$$\begin{aligned}
\mathbb{E}[\|m_t'\|^2] &= \mathbb{E}[\|\beta_1 m_{t-1}' + (1-\beta_1)g_t\|^2] \\
&\leq (1+\frac{\rho}{2})\beta_1^2 \mathbb{E}[\|m_{t-1}'\|^2] + (1+\frac{1}{2\rho})(1-\beta_1)^2 \mathbb{E}[\|g_t\|^2].
\end{aligned}$$

Since $\mathbb{E}[\|g_t\|^2] \leq \sigma^2 + \mathbb{E}[\|\nabla f(\theta_t)\|^2]$, by choosing $\rho = 2(1-\beta_1^2)$, we derive

$$\begin{aligned}
\mathbb{E}[\|m_t'\|^2] &\leq \beta_1^2(2-\beta_1^2)\mathbb{E}[\|m_{t-1}'\|^2] + (1-\beta_1)^2(1+\frac{1}{4(1-\beta_1^2)})\mathbb{E}[\|g_t\|^2] && (3)\\
&\leq \frac{(1-\beta_1)^2}{1-\beta_1^2(2-\beta_1^2)}(1+\frac{1}{4(1-\beta_1^2)})\sigma^2 + C_1 \sum_{\tau=1}^{t} (\beta_1^2(2-\beta_1^2))^{t-\tau} \mathbb{E}[\|\nabla f(\theta_\tau)\|^2] && (4)\\
&:= C\sigma^2 + C_1 \sum_{\tau=1}^{t} (\beta_1^2(2-\beta_1^2))^{t-\tau} \mathbb{E}[\|\nabla f(\theta_\tau)\|^2], && (5)
\end{aligned}$$

due to $\beta_1 < 1$, $m_0' = 0$ and the bounded variance assumption. Here $C_1 = (1-\beta_1^2)(1+\frac{1}{4(1-\beta_1^2)})$ and $C = \frac{C_1}{1-\beta_1^2(2-\beta_1^2)}$.

For m_t which consists of the compressed stochastic gradients, first note that

$$\begin{aligned}
\mathbb{E}[\|\tilde{g}_t\|^2] &= \mathbb{E}[\|\mathcal{C}(g_t+e_t) - (g_t+e_t) + g_t + e_t - \nabla f(\theta_t) + \nabla f(\theta_t)\|^2] \\
&\leq \sigma^2 + 3\mathbb{E}[q^2\|g_t + e_t - \nabla f(\theta_t) + \nabla f(\theta_t)\|^2 + \|e_t\|^2 + \|\nabla f(\theta_t)\|^2] \\
&\leq (3q^2+1)\sigma^2 + (6q^2+3)\mathbb{E}[\|e_t\|^2 + \|\nabla f(\theta_t)\|^2] \\
&\leq (3q^2 + \frac{4q^2(6q^2+3)}{(1-q^2)^2} + 1)\sigma^2 + (6q^2+3)\mathbb{E}[\|\nabla f(\theta_t)\|^2],
\end{aligned}$$

where the first inequality is because of Assumption 1 and that the stochastic error $(g_t - \nabla f(\theta_t))$ is mean-zero and independent of other terms. The bound on $\|e_t\|^2$ in the last inequality is due to Lemma 3 of [31]. Then by similar induction we can obtain

$$\mathbb{E}[\|m_t\|^2] \leq (3q^2 + \frac{4q^2(6q^2+3)}{(1-q^2)^2} + 1)C\sigma^2 + (6q^2+3)C_1 \sum_{\tau=1}^{t} (\beta_1^2(2-\beta_1^2))^{t-\tau} \mathbb{E}[\|\nabla f(\theta_\tau)\|^2].$$

□

Lemma 2. *Suppose* $\gamma = \beta_1/\beta_2 < 1$. *Then, for* $\forall t$,

$$\|a_t\|^2 := \|\frac{m_t}{\sqrt{\hat{v}_t + \epsilon}}\|^2 \leq \frac{(1-\beta_1)d}{(1-\beta_2)(1-\gamma)}.$$

Proof. We have

$$
\begin{aligned}
\|\frac{m_t}{\sqrt{\hat{v}_t+\epsilon}}\|^2 &= \sum_{i=1}^{d}\frac{m_{t,i}^2}{\hat{v}_{t,i}+\epsilon} \\
&\leq \frac{(1-\beta_1)^2}{1-\beta_2}\sum_{i=1}^{d}\frac{(\sum_{\tau=1}^{t}\beta_1^{t-\tau}\tilde{g}_{\tau,i})^2}{\sum_{\tau=1}^{t}\beta_2^{t-\tau}\tilde{g}_{\tau,i}^2} \\
&\overset{(a)}{\leq} \frac{(1-\beta_1)^2}{1-\beta_2}\sum_{i=1}^{d}\frac{(\sum_{\tau=1}^{t}\beta_1^{t-\tau})(\sum_{\tau=1}^{t}\beta_1^{t-\tau}\tilde{g}_{\tau,i}^2)}{\sum_{\tau=1}^{t}\beta_2^{t-\tau}\tilde{g}_{\tau,i}^2} \\
&\leq \frac{1-\beta_1}{1-\beta_2}\sum_{i=1}^{d}\frac{\sum_{\tau=1}^{t}\beta_1^{t-\tau}\tilde{g}_{\tau,i}^2}{\sum_{\tau=1}^{t}\beta_2^{t-\tau}\tilde{g}_{\tau,i}^2} \\
&\leq \frac{(1-\beta_1)d}{1-\beta_2}\sum_{\tau=1}^{t}\gamma^{\tau} \\
&\leq \frac{(1-\beta_1)d}{(1-\beta_2)(1-\gamma)},
\end{aligned}
$$

where (a) is a consequence of Cauchy-Schwartz inequality. □

Lemma 3. *Define*

$$
\begin{aligned}
H_t &:= \mathbb{E}[\sum_{i=1}^{d}|\frac{1}{\sqrt{\hat{v}_{t-1}+\epsilon}}-\frac{1}{\sqrt{\hat{v}_t+\epsilon}}|] \\
S_t &:= \sum_{\tau=1}^{t}(\beta_1^2(2-\beta_1^2))^{t-\tau}\mathbb{E}[\|\nabla f(\theta_\tau)\|^2])
\end{aligned}
$$

then the following inequalities hold:

$$
\begin{aligned}
&\sum_{t=2}^{T}\sum_{\tau=0}^{t-2}\beta_1^{\tau}S_{t-\tau} \leq \frac{1}{(1-\beta_1)(1-\beta_1^2(2-\beta_1^2))}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2] \\
&\sum_{t=2}^{T}\sum_{\tau=0}^{t-2}\beta_1^{\tau}H_{t-\tau} \leq \frac{d}{(1-\beta)\sqrt{\epsilon}}.
\end{aligned}
$$

Proof. By arranging terms, it holds that

$$
\begin{aligned}
\sum_{t=2}^{T}\sum_{\tau=0}^{t-2}\beta_1^{\tau}S_{t-\tau} &\leq \sum_{t=2}^{T}(\sum_{\tau=0}^{T-t}\beta_1^{T-t-\tau})S_t \\
&\leq \frac{1}{1-\beta_1}\sum_{t=2}^{T}\sum_{\tau=1}^{t}(\beta_1^2(2-\beta_1^2))^{t-\tau}\mathbb{E}[\|\nabla f(\theta_\tau)\|^2]) \\
&\leq \frac{1}{1-\beta_1}\sum_{t=1}^{T}(\sum_{\tau=0}^{T-t-1}(\beta_1^2(2-\beta_1^2))^{T-t-\tau})\mathbb{E}[\|\nabla f(\theta_t)\|^2] \\
&\leq \frac{1}{(1-\beta_1)(1-\beta_1^2(2-\beta_1^2))}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2].
\end{aligned}
$$

Using similar strategy, we can write

$$\begin{aligned}\sum_{t=2}^{T}\sum_{\tau=0}^{t-2}\beta_1^{\tau}H_{t-\tau} &\leq \sum_{t=2}^{T}(\sum_{\tau=0}^{T-t}\beta_1^{T-t-\tau})H_t\\ &\leq \frac{1}{1-\beta}\sum_{t=2}^{T}\mathbb{E}[\sum_{i=1}^{d}|\frac{1}{\sqrt{\hat{v}_{t-1}+\epsilon}}-\frac{1}{\sqrt{\hat{v}_t+\epsilon}}|\\ &\leq \frac{d}{(1-\beta)\sqrt{\epsilon}},\end{aligned}$$

where the last inequality is derived by cancelling terms due to the fact that $\{\hat{v}_t\}_{t>0}$ is a non-decreasing sequence, hence $\hat{v}_t \leq \hat{v}_{t-1}$. This completes the proof of the lemma. □

Lemma 4. *For the error sequence e_t in* SPAMS*, under Assumption 4, we have for $\forall t$,*

$$\mathbb{E}[\|e_{t+1}\|^2] \leq \frac{4q^2}{(1-q^2)^2}\sigma^2 + \frac{2q^2}{1-q^2}\sum_{\tau=1}^{t}(\frac{1+q^2}{2})^{t-\tau}\mathbb{E}[\|\nabla f(\theta_\tau)\|^2].$$

Proof. We start by using Assumption 1 and Young's inequality to get

$$\begin{aligned}\|e_{t+1}\|^2 &= \|g_t + e_t - \mathcal{C}(g_t+e_t)\|^2\\ &\leq q^2\|g_t+e_t\|^2\\ &\leq q^2(1+\rho)\|e_t\|^2 + q^2(1+\frac{1}{\rho})\|g_t\|^2\\ &\leq \frac{1+q^2}{2}\|e_t\|^2 + \frac{2q^2}{1-q^2}\|g_t\|^2,\end{aligned}$$

by choosing $\rho = \frac{1-q^2}{2q^2}$. Now by recursion and the initialization $e_1 = 0$, we have

$$\begin{aligned}\mathbb{E}[\|e_{t+1}\|^2] &\leq \frac{2q^2}{1-q^2}\sum_{\tau=1}^{t}(\frac{1+q^2}{2})^{t-\tau}\mathbb{E}[\|g_\tau\|^2]\\ &\leq \frac{4q^2}{(1-q^2)^2}\sigma^2 + \frac{2q^2}{1-q^2}\sum_{\tau=1}^{t}(\frac{1+q^2}{2})^{t-\tau}\mathbb{E}[\|\nabla f(\theta_\tau)\|^2],\end{aligned}$$

which proves the lemma. Meanwhile, we also have the absolute bound $\|e_t\|^2 \leq \frac{4q^2}{(1-q^2)^2}G^2$. □

Lemma 5. *For the moving average error sequence $\mathcal{E}_t$, it holds that*

$$\sum_{t=1}^{T}\mathbb{E}[\|\mathcal{E}_t\|^2] \leq \frac{4Tq^2}{(1-q^2)^2\epsilon}\sigma^2 + \frac{4q^2}{(1-q^2)^2\epsilon}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2].$$

Proof. Denote $K_t := \sum_{\tau=1}^{t}(\frac{1+q^2}{2})^{t-\tau}\mathbb{E}[\|\nabla f(\theta_\tau)\|^2]$ and $K_0 = 0$. We have

$$\begin{aligned}\mathbb{E}[\|\mathcal{E}_t\|^2] &= \mathbb{E}[\|\frac{(1-\beta_1)\sum_{\tau=1}^{t}\beta_1^{t-\tau}e_\tau}{\sqrt{\hat{v}_t+\epsilon}}\|^2]\\ &\leq \frac{(1-\beta_1)^2}{\epsilon}\sum_{i=1}^{d}\mathbb{E}[(\sum_{\tau=1}^{t}\beta_1^{t-\tau}e_{\tau,i})^2]\\ &\overset{(a)}{\leq} \frac{(1-\beta_1)^2}{\epsilon}\sum_{i=1}^{d}\mathbb{E}[(\sum_{\tau=1}^{t}\beta_1^{t-\tau})(\sum_{\tau=1}^{t}\beta_1^{t-\tau}e_{\tau,i}^2)]\\ &\leq \frac{1-\beta_1}{\epsilon}\sum_{\tau=1}^{t}\beta_1^{t-\tau}\mathbb{E}[\|e_\tau\|^2]\\ &\overset{(b)}{\leq} \frac{4q^2}{(1-q^2)^2\epsilon}\sigma^2 + \frac{2q^2(1-\beta_1)}{(1-q^2)\epsilon}\sum_{\tau=1}^{t}\beta_1^{t-\tau}K_\tau,\end{aligned}$$

where (a) is due to Cauchy-Schwartz and (b) is a result of Lemma 4. Summing over $t = 1, ..., T$ and using the similar technique as in Lemma 3 leads to

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}[\|\mathcal{E}_t\|^2] &= \frac{4Tq^2}{(1-q^2)^2\epsilon}\sigma^2 + \frac{2q^2(1-\beta_1)}{(1-q^2)\epsilon}\sum_{t=1}^{T}\sum_{\tau=1}^{t}\beta_1^{t-\tau}K_\tau \\ &\leq \frac{4Tq^2}{(1-q^2)^2\epsilon}\sigma^2 + \frac{2q^2}{(1-q^2)\epsilon}\sum_{t=1}^{T}\sum_{\tau=1}^{t}(\frac{1+q^2}{2})^{t-\tau}\mathbb{E}[\|\nabla f(\theta_\tau)\|^2] \\ &\leq \frac{4Tq^2}{(1-q^2)^2\epsilon}\sigma^2 + \frac{4q^2}{(1-q^2)^2\epsilon}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2],\end{aligned}$$

which gives the desired result.

$\square$

Lemma 6. *It holds that* $\forall t \in [T]$, $\forall i \in [d]$, $\hat{v}_{t,i} \leq \frac{4(1+q^2)^3}{(1-q^2)^2}G^2$.

Proof. For any t, by Lemma 4 and Assumption 3 we have

$$\begin{aligned}\|\tilde{g}_t\|^2 &= \|\mathcal{C}(g_t + e_t)\|^2 \\ &\leq \|\mathcal{C}(g_t + e_t) - (g_t + e_t) + (g_t + e_t)\|^2 \\ &\leq 2(q^2+1)\|g_t + e_t\|^2 \\ &\leq 4(q^2+1)(G^2 + \frac{4q^2}{(1-q^2)^2}G^2) \\ &= \frac{4(1+q^2)^3}{(1-q^2)^2}G^2.\end{aligned}$$

It's then easy to show by the updating rule of $\hat{v}_t$,

$$\hat{v}_{t,i} = (1-\beta_2)\sum_{\tau=1}^{t}\tilde{g}_{t,i}^2 \leq \frac{4(1+q^2)^3}{(1-q^2)^2}G^2.$$

$\square$

B Proof of Theorem 1

Theorem. *Denote* $C_0 = \sqrt{\frac{4(1+q^2)^3}{(1-q^2)^2}G^2 + \epsilon}$, $C_1 = \frac{\beta_1}{1-\beta_1} + \frac{2q}{1-q^2}$. *Under Assumption 1 to Assumption 4, with* $\eta_t = \eta \le \frac{\epsilon}{4LC_0}$, *then for* $T > 0$, SPAMS *satisfies*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2] \le 2C_0\Big(\frac{\mathbb{E}[f(\theta_1)-f(\theta^*)]}{T\eta} + \frac{\eta L\sigma^2}{n\epsilon} + \frac{\eta^2 C_0 C_1^2 LG^2}{\epsilon^2} + \frac{\eta(1+C_1)G^2 d}{T\sqrt{\epsilon}} + \frac{\eta^2(1+2C_1)C_1 LG^2 d}{T\epsilon}\Big),$$

Proof. We first clarify some notations. At time t, let the full-precision gradient of the j-th worker be $g_{t,j}$, the error accumulator be $e_{t,j}$, and the compressed gradient be $\tilde{g}_{t,j} = \mathcal{C}(g_{t,j} + e_{t,j})$. Denote $\bar{g}_t = \frac{1}{n}\sum_{j=1}^{N} g_{t,j}$, $\bar{\tilde{g}}_t = \frac{1}{n}\sum_{j=1}^{N}\tilde{g}_{t,j}$ and $\bar{e}_t = \frac{1}{n}\sum_{j=1}^{n} e_{t,j}$. The second moment computed by the compressed gradients is denoted as $v_t = \beta_2 v_{t-1} + (1-\beta_2)\bar{\tilde{g}}_t^2$, and $\hat{v}_t = \max\{\hat{v}_{t-1}, v_t\}$. Also, the first order moving average sequence

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\bar{\tilde{g}}_t \quad \text{and} \quad m'_t = \beta_1 m'_{t-1} + (1-\beta_1)\bar{g}_t.$$

By construction we have $m'_t = (1-\beta_1)\sum_{i=1}^{k}\beta_1^{t-i}\bar{g}_t$.

Denote the following auxiliary sequences,

$$\mathcal{E}_{t+1} := (1-\beta_1)\sum_{\tau=1}^{t+1}\beta_1^{t+1-\tau}\bar{e}_\tau$$
$$\theta'_{t+1} := \theta_{t+1} - \eta\frac{\mathcal{E}_{t+1}}{\sqrt{\hat{v}_t + \epsilon}}.$$

Then,

$$\begin{aligned}
\theta'_{t+1} &= \theta_{t+1} - \eta\frac{\mathcal{E}_{t+1}}{\sqrt{\hat{v}_t+\epsilon}} \\
&= \theta_t - \eta\frac{(1-\beta_1)\sum_{\tau=1}^{t}\beta_1^{t-\tau}\bar{\tilde{g}}_\tau + (1-\beta_1)\sum_{\tau=1}^{t+1}\beta_1^{t+1-\tau}\bar{e}_\tau}{\sqrt{\hat{v}_t+\epsilon}} \\
&= \theta_t - \eta\frac{(1-\beta_1)\sum_{\tau=1}^{t}\beta_1^{t-\tau}(\bar{\tilde{g}}_\tau + \bar{e}_{\tau+1}) + (1-\beta)\beta_1^t\bar{e}_1}{\sqrt{\hat{v}_t+\epsilon}} \\
&= \theta_t - \eta\frac{(1-\beta_1)\sum_{\tau=1}^{t}\beta_1^{t-\tau}\bar{e}_\tau}{\sqrt{\hat{v}_t+\epsilon}} - \eta\frac{m'_t}{\sqrt{\hat{v}_t+\epsilon}} \\
&= \theta_t - \eta\frac{\mathcal{E}_t}{\sqrt{\hat{v}_{t-1}+\epsilon}} - \eta\frac{m'_t}{\sqrt{\hat{v}_t+\epsilon}} + \eta(\frac{1}{\sqrt{\hat{v}_{t-1}+\epsilon}} - \frac{1}{\sqrt{\hat{v}_t+\epsilon}})\mathcal{E}_t \\
&\overset{(a)}{=} \theta'_t - \eta\frac{m'_t}{\sqrt{\hat{v}_t+\epsilon}} + \eta(\frac{1}{\sqrt{\hat{v}_{t-1}+\epsilon}} - \frac{1}{\sqrt{\hat{v}_t+\epsilon}})\mathcal{E}_t \\
&:= \theta'_t - \eta a'_t + \eta D_t\mathcal{E}_t,
\end{aligned}$$

where (a) uses the fact that for every $j \in [n]$, $\tilde{g}_{t,j} + e_{t+1,j} = g_{t,j} + e_{t,j}$, and $e_{t,1} = 0$ at initialization. Further define the virtual iterates:

$$x_{t+1} := \theta'_{t+1} - \eta\frac{\beta_1}{1-\beta_1}a'_t = \theta'_{t+1} - \eta\frac{\beta_1}{1-\beta_1}\frac{m'_t}{\sqrt{\hat{v}_t+\epsilon}},$$

which follows the recurrence:

$$
\begin{aligned}
x_{t+1} &= \theta'_{t+1} - \eta \frac{\beta_1}{1-\beta_1} \frac{m'_t}{\sqrt{\hat{v}_t + \epsilon}} \\
&= \theta'_t - \eta \frac{m'_t}{\sqrt{\hat{v}_t + \epsilon}} - \eta \frac{\beta_1}{1-\beta_1} \frac{m'_t}{\sqrt{\hat{v}_t + \epsilon}} + \eta D_t \mathcal{E}_t \\
&= \theta'_t - \eta \frac{\beta_1 m'_{t-1} + (1-\beta_1)\bar{g}_t + \frac{\beta_1^2}{1-\beta_1} m'_{t-1} + \beta_1 \bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} + \eta D_t \mathcal{E}_t \\
&= \theta'_t - \eta \frac{\beta_1}{1-\beta_1} \frac{m'_{t-1}}{\sqrt{\hat{v}_t + \epsilon}} - \eta \frac{\bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} + \eta D_t \mathcal{E}_t \\
&= x_t - \eta \frac{\bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} + \eta \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} + \eta D_t \mathcal{E}_t.
\end{aligned}
$$

When summing over $t = 1, ..., T$, the difference sequence D_t satisfies the following bounds.

Lemma 7. *Let $D_t := \frac{1}{\sqrt{\hat{v}_{t-1}+\epsilon}} - \frac{1}{\sqrt{\hat{v}_t+\epsilon}}$ be defined as above. Then,*

$$
\sum_{t=1}^{T} \|D_t\|_1 \leq \frac{d}{\sqrt{\epsilon}}, \quad \sum_{t=1}^{T} \|D_t\|^2 \leq \frac{d}{\epsilon}
$$

Proof. By the updating rule of SPAMS, $\hat{v}_{t-1} \leq \hat{v}_t$ for $\forall t$. Therefore, by the initialization $\hat{v}_0 = 0$, we have

$$
\begin{aligned}
\sum_{t=1}^{T} \|D_t\|_1 &= \sum_{t=1}^{T} \sum_{i=1}^{d} \left(\frac{1}{\sqrt{\hat{v}_{t-1} + \epsilon}} - \frac{1}{\sqrt{\hat{v}_t + \epsilon}}\right) \\
&= \sum_{i=1}^{d} \left(\frac{1}{\sqrt{\hat{v}_0 + \epsilon}} - \frac{1}{\sqrt{\hat{v}_T + \epsilon}}\right) \\
&\leq \frac{d}{\sqrt{\epsilon}}.
\end{aligned}
$$

For the sum of squared l_2 norm, note the fact that for $a \geq b > 0$, it holds that

$$
(a-b)^2 \leq (a-b)(a+b) = a^2 - b^2.
$$

Thus,

$$
\begin{aligned}
\sum_{t=1}^{T} \|D_t\|^2 &= \sum_{t=1}^{T} \sum_{i=1}^{d} \left(\frac{1}{\sqrt{\hat{v}_{t-1} + \epsilon}} - \frac{1}{\sqrt{\hat{v}_t + \epsilon}}\right)^2 \\
&\leq \sum_{t=1}^{T} \sum_{i=1}^{d} \left(\frac{1}{\hat{v}_{t-1} + \epsilon} - \frac{1}{\hat{v}_t + \epsilon}\right) \\
&\leq \frac{d}{\epsilon}.
\end{aligned}
$$

□

By Assumption 2 we have

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

Taking expectation w.r.t. the randomness at time t, we obtain

$$
\begin{aligned}
&\mathbb{E}[f(x_{t+1})] - f(x_t) \\
&\le -\eta\mathbb{E}[\langle \nabla f(x_t), \frac{\bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} \rangle] + \eta\mathbb{E}[\langle \nabla f(x_t), \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} + D_t \mathcal{E}_t \rangle] \\
&\qquad\qquad\qquad\qquad + \frac{\eta^2 L}{2}\mathbb{E}[\| \frac{\bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} - \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} - D_t \mathcal{E}_t \|^2] \\
&= \underbrace{-\eta\mathbb{E}[\langle \nabla f(\theta_t), \frac{\bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} \rangle]}_{I} + \underbrace{\eta\mathbb{E}[\langle \nabla f(x_t), \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} + D_t \mathcal{E}_t \rangle]}_{II} \\
&\qquad + \underbrace{\frac{\eta^2 L}{2}\mathbb{E}[\| \frac{\bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} - \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} - D_t \mathcal{E}_t \|^2]}_{III} + \underbrace{\eta\mathbb{E}[\langle \nabla f(\theta_t) - \nabla f(x_t), \frac{\bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} \rangle]}_{IV},
\end{aligned} \tag{6}
$$

Bounding term I. We have

$$
\begin{aligned}
I &= -\eta\mathbb{E}[\langle \nabla f(\theta_t), \frac{\bar{g}_t}{\sqrt{\hat{v}_{t-1} + \epsilon}}] - \eta\mathbb{E}[\langle \nabla f(\theta_t), (\frac{1}{\sqrt{\hat{v}_t + \epsilon}} - \frac{1}{\sqrt{\hat{v}_{t-1} + \epsilon}}) \bar{g}_t \rangle] \\
&\le -\eta\mathbb{E}[\langle \nabla f(\theta_t), \frac{\nabla f(\theta_t)}{\sqrt{\hat{v}_{t-1} + \epsilon}}] + \eta G^2 \mathbb{E}[\|D_t\|]. \\
&\le -\frac{\eta}{\sqrt{\frac{4(1+q^2)^3}{(1-q^2)^2} G^2 + \epsilon}} \mathbb{E}[\|\nabla f(\theta_t)\|^2] + \eta G^2 \mathbb{E}[\|D_t\|_1],
\end{aligned} \tag{7}
$$

where we use Assumption 3, Lemma 6 and the fact that l_2 norm is no larger than l_1 norm.

Bounding term II. It holds that

$$
\begin{aligned}
II &\le \eta(\mathbb{E}[\langle \nabla f(\theta_t), \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} + D_t \mathcal{E}_t \rangle] + \mathbb{E}[\langle \nabla f(x_t) - \nabla f(\theta_t), \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} + D_t \mathcal{E}_t \rangle]) \\
&\le \eta\mathbb{E}[\|\nabla f(\theta_t)\| \| \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} + D_t \mathcal{E}_t \|] + \eta^2 \ L\mathbb{E}[\| \frac{\frac{\beta_1}{1-\beta_1} m'_{t-1} + \mathcal{E}_t}{\sqrt{\hat{v}_{t-1} + \epsilon}} \| \| \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} + D_t \mathcal{E}_t \|] \\
&\le \eta C_1 G^2 \mathbb{E}[\|D_t\|_1] + \frac{\eta^2 C_1^2 L G^2}{\sqrt{\epsilon}} \mathbb{E}[\|D_t\|_1],
\end{aligned} \tag{8}
$$

where $C_1 := \frac{\beta_1}{1-\beta_1} + \frac{2q}{1-q^2}$. The second inequality is because of smoothness of $f(\theta)$, and the last inequality is due to Lemma 4, Assumption 3 and the property of norms.

Bounding term III. This term can be bounded as follows:

$$
\begin{aligned}
III &\le \eta^2 L\mathbb{E}[\| \frac{\bar{g}_t}{\sqrt{\hat{v}_t + \epsilon}} \|^2] + \eta^2 L\mathbb{E}[\| \frac{\beta_1}{1-\beta_1} D_t m'_{t-1} - D_t \mathcal{E}_t \|^2]] \\
&\le \frac{\eta^2 L}{\epsilon}\mathbb{E}[\| \frac{1}{n}\sum_{j=1}^{i} g_{t,j} - \nabla f(\theta_t) + \nabla f(\theta_t) \|^2] + \eta^2 L\mathbb{E}[\|D_t(\frac{\beta_1}{1-\beta_1} m'_{t-1} - \mathcal{E}_t)\|^2] \\
&\overset{(a)}{\le} \frac{\eta^2 L}{\epsilon}\mathbb{E}[\|\nabla f(\theta_t)\|^2] + \frac{\eta^2 L \sigma^2}{n\epsilon} + \eta^2 C_1^2 L G^2 \mathbb{E}[\|D_t\|^2],
\end{aligned} \tag{9}
$$

where (a) follows from $\nabla f(\theta_t) = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\theta_t)$ and Assumption 4 that $g_{t,j}$ is unbiased of $\nabla f_j(\theta_t)$ and has bounded variance σ^2.

Bounding term IV. We have

$$
\begin{aligned}
IV &= \eta\mathbb{E}[\langle \nabla f(\theta_t) - \nabla f(x_t), \frac{\bar{g}_t}{\sqrt{\hat{v}_{t-1}+\epsilon}}\rangle] + \eta\mathbb{E}[\langle \nabla f(\theta_t) - \nabla f(x_t), (\frac{1}{\sqrt{\hat{v}_t+\epsilon}} - \frac{1}{\sqrt{\hat{v}_{t-1}+\epsilon}})\bar{g}_t\rangle] \\
&\le \eta\mathbb{E}[\langle \nabla f(\theta_t) - \nabla f(x_t), \frac{\nabla f(\theta_t)}{\sqrt{\hat{v}_{t-1}+\epsilon}}\rangle] + \eta^2 L\mathbb{E}[\| \frac{\frac{\beta_1}{1-\beta_1}m'_{t-1} + \mathcal{E}_t}{\sqrt{\hat{v}_{t-1}+\epsilon}}\| \|D_t g_t\|] \\
&\overset{(a)}{\le} \frac{\eta\rho}{2\epsilon}\mathbb{E}[\|\nabla f(\theta_t)\|^2] + \frac{\eta}{2\rho}\mathbb{E}[\|\nabla f(\theta_t) - \nabla f(x_t)\|^2] + \frac{\eta^2 C_1 L G^2}{\sqrt{\epsilon}}\mathbb{E}[\|D_t\|] \\
&\overset{(b)}{\le} \frac{\eta\rho}{2\epsilon}\mathbb{E}[\|\nabla f(\theta_t)\|^2] + \frac{\eta^3 L}{2\rho}\mathbb{E}[\| \frac{\frac{\beta_1}{1-\beta_1}m'_{t-1} + \mathcal{E}_t}{\sqrt{\hat{v}_{t-1}+\epsilon}}\|^2] + \frac{\eta^2 C_1 L G^2}{\sqrt{\epsilon}}\mathbb{E}[\|D_t\|_1] \\
&\le \frac{\eta\rho}{2\epsilon}\mathbb{E}[\|\nabla f(\theta_t)\|^2] + \frac{\eta^3 C_1^2 L G^2}{2\rho\epsilon} + \frac{\eta^2 C_1 L G^2}{\sqrt{\epsilon}}\mathbb{E}[\|D_t\|_1],
\end{aligned}
\tag{10}
$$

where (a) is due to Young's inequality and (b) is based on Assumption 2.

Now integrating (7), (8), (9) and (10) into (6), we obtain

$$
\begin{aligned}
&\mathbb{E}[f(x_{t+1})] - f(x_t) \\
&\le (-\frac{\eta}{C_0} + \frac{\eta^2 L}{\epsilon} + \frac{\eta\rho}{2\epsilon})\mathbb{E}[\|\nabla f(\theta_t)\|^2] + \frac{\eta^2 L\sigma^2}{n\epsilon} + \frac{\eta^3 C_1^2 L G^2}{2\rho\epsilon} \\
&\qquad + (\eta(1+C_1)G^2 + \frac{\eta^2(1+C_1)C_1 L G^2}{\sqrt{\epsilon}})\mathbb{E}[\|D_t\|_1] + \eta^2 C_1^2 L G^2\mathbb{E}[\|D_t\|^2].
\end{aligned}
$$

Denote $C_0 := \sqrt{\frac{4(1+q^2)^3}{(1-q^2)^2}G^2 + \epsilon}$. Setting $\eta \le \frac{\epsilon}{4LC_0}$ and choosing $\rho = \frac{\epsilon}{2C_0}$, we obtain

$$
\begin{aligned}
&\mathbb{E}[f(x_{t+1})] - f(x_t) \\
&\le -\frac{\eta}{2C_0}\mathbb{E}[\|\nabla f(\theta_t)\|^2] + \frac{\eta^2 L\sigma^2}{n\epsilon} + \frac{\eta^3 C_0 C_1^2 L G^2}{\epsilon^2} \\
&\qquad + (\eta(1+C_1)G^2 + \frac{\eta^2(1+C_1)C_1 L G^2}{\sqrt{\epsilon}})\mathbb{E}[\|D_t\|_1] + \eta^2 C_1^2 L G^2\mathbb{E}[\|D_t\|^2].
\end{aligned}
$$

Summing over $t = 1, ..., T$, we get

$$
\begin{aligned}
&\mathbb{E}[f(x_{T+1}) - f(x_1)] \\
&\le -\frac{\eta}{2C_0}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2] + \frac{T\eta^2 L\sigma^2}{n\epsilon} + \frac{T\eta^3 C_0 C_1^2 L G^2}{\epsilon^2} \\
&\qquad + (\eta(1+C_1)G^2 + \frac{\eta^2(1+C_1)C_1 L G^2}{\sqrt{\epsilon}})\sum_{t=1}^{T}\mathbb{E}[\|D_t\|_1] + \eta^2 C_1^2 L G^2\sum_{t=1}^{T}\mathbb{E}[\|D_t\|^2 \\
&\le -\frac{\eta}{2C_0}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2] + \frac{T\eta^2 L\sigma^2}{n\epsilon} + \frac{T\eta^3 C_0 C_1^2 L G^2}{\epsilon^2} + \frac{\eta(1+C_1)G^2 d}{\sqrt{\epsilon}} + \frac{\eta^2(1+2C_1)C_1 L G^2 d}{\epsilon},
\end{aligned}
$$

where the last inequality follows from Lemma 7. Re-arranging terms, we get that when $\eta \le \frac{\epsilon}{4LC_0}$,

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\theta_t)\|^2] &\le 2C_0\Big(\frac{\mathbb{E}[f(x'_1) - f(x'_{T+1})]}{T\eta} + \frac{\eta L\sigma^2}{n\epsilon} + \frac{\eta^2 C_0 C_1^2 L G^2}{\epsilon^2} \\
&\qquad + \frac{\eta(1+C_1)G^2 d}{T\sqrt{\epsilon}} + \frac{\eta^2(1+2C_1)C_1 L G^2 d}{T\epsilon}\Big) \\
&\le 2C_0\Big(\frac{\mathbb{E}[f(\theta_1) - f(\theta^*)]}{T\eta} + \frac{\eta L\sigma^2}{n\epsilon} + \frac{\eta^2 C_0 C_1^2 L G^2}{\epsilon^2} \\
&\qquad + \frac{\eta(1+C_1)G^2 d}{T\sqrt{\epsilon}} + \frac{\eta^2(1+2C_1)C_1 L G^2 d}{T\epsilon}\Big),
\end{aligned}
$$

where $C_0 = \sqrt{\frac{4(1+q^2)^3}{(1-q^2)^2} G^2 + \epsilon}$, $C_1 = \frac{\beta_1}{1-\beta_1} + \frac{2q}{1-q^2}$, and the last inequality is because $\theta_1' = \theta_1$, and $\theta^* := \arg\min_\theta f(\theta)$. This completes the proof.

$\square$