ErrorCompensatedX: error compensation for variance reduced algorithms

Hanlin Tang
Department of Computer Science
University of Rochester
tanghl1994@gmail.com

Yao Li
Department of Mathematics
Michigan State University
liyao6@msu.edu

Ji Liu
Kuaishou Technology
ji.liu.uwisc@gmail.com

Ming Yan
Department of Computational Mathematics, Science
and Technology; Department of Mathematics
Michigan State University
myan@msu.edu

Abstract

Communication cost is one major bottleneck for the scalability for distributed learning. One approach to reduce the communication cost is to compress the gradient during communication. However, directly compressing the gradient decelerates the convergence speed, and the resulting algorithm may diverge for biased compression. Recent work addressed this problem for stochastic gradient descent by adding back the compression error from the previous step. This idea was further extended to one class of variance reduced algorithms, where the variance of the stochastic gradient is reduced by taking a moving average over all history gradients. However, our analysis shows that just adding the previous step's compression error, as done in existing work, does not fully compensate the compression error. So, we propose ErrorCompensatedX, which uses the compression error from the previous two steps. We show that ErrorCompensatedX can achieve the same asymptotic convergence rate with the training without compression. Moreover, we provide a unified theoretical analysis framework for this class of variance reduced algorithms, with or without error compensation.

1 Introduction

Data compression reduces the communication volume and alleviates the communication overhead in distributed learning. E.g., Alistarh et al. (2017) compress the gradient being communicated using quantization and find that reducing half of the communication size does not degrade the convergence speed. However, the convergence speed would be slower if we further reduce the communication size, and it requires the compression to be unbiased (Alistarh et al., 2017; Tang et al., 2019). Alternatively, recent work (Stich et al., 2018a) shows that compression with error compensation, which adds back the compression error to the next round of compression, using only 3% of the original communication volume does not degrade the convergence speed, and it works for both biased and unbiased compression operators.

Despite the promising performance of error compensation on stochastic gradient descent (SGD), SGD admits a slow convergence speed if the stochastic gradient has a large variance. Variance reduction techniques, such as **Momentum SGD** (Zhang et al., 2015), **ROOT-SGD** (Li et al., 2020), **STORM** (Cutkosky & Mehta, 2020), and **IGT** (Arnold et al., 2019), are developed, and they admit increased convergence speeds either theoretically or empirically. We found that directly applying

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

error compensation to those variance reduced algorithms is not *optimal*, and their convergence speeds degrade. So a natural question arises: **what is the best compression method for variance reduced algorithms?** In this paper, we answer this question and propose ErrorCompensatedX, a general method for error compensation, and show that it admits faster convergence speeds than previous error compensation methods. The contributions of this paper can be summarized as follows:

- We propose a novel error compensation algorithm for variance reduced algorithms and SGD. Our algorithm admits a faster convergence rate compared to previous methods (Zheng et al., 2019a; Stich et al., 2018b) by fully compensating all error history.
- We provide a general theoretical analysis framework to analyze error compensated algorithms. More specifically, we decompose the convergence rate into the sum of two terms

  $$\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(\boldsymbol{x}_t)\|^2 \leq \mathcal{R}_{\text{uncompressed}} + \mathcal{R}_{\epsilon},$$

  where $\mathcal{R}_{\text{uncompressed}}$ depends only on the convergence rate of the original algorithm without compression and $\mathcal{R}_{\epsilon}$ is depends only on the magnitude of the compression error ϵ. It means that we can easily attain the convergence rate for any compressed algorithm in the form of (2). To the best of our knowledge, this is the first general result for error compensation.

2 Related Works

Variance Reduced Optimization. When the variance of the stochastic gradient is large, training using SGD becomes unstable, and it usually renders a slow convergence speed. Many studies try to reduce the variance of the stochastic gradient. For example, Momentum SGD (Zhang et al., 2015) takes a moving average over all previous stochastic gradients. However, its expected momentum does not equal to the full gradient. Recent studies take one step further by applying bias correction to the momentum. One approach is to compute the gradient at $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ using the same data sample ξ_t (Huang et al., 2020; Yuan et al., 2020; Li et al., 2020), while another direction is to compute the gradient at the extrapolated point of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ (Arnold et al., 2019; Cutkosky & Mehta, 2020). When the number of data samples is finite, some approaches compute the full gradient once in a while or memorize the latest stochastic gradient of each data sample and use its average to approximate the full gradient; examples include SVRG (Johnson & Zhang, 2013), SAGA (Defazio et al., 2014), and SARAH (Nguyen et al., 2017).

Communication Efficient Optimization. One major bottleneck of distributed learning is the communication cost. Methods, such as data compression, decentralized training, local SGD, and federated learning, were proposed to reduce the communication cost. Data compression is an important approach that can be combined with other approaches, and it is critical for communication networks with limited bandwidth. It was firstly proposed in Seide et al. (2014), where authors proposed **1-bit SGD**, which uses one bit to represent each element in the gradient but still achieves almost the same convergence speed with the uncompressed one. In Stich et al. (2018b), authors proposed a general algorithm for error compensated compression—**MEM-SGD** with theoretical analysis. They found that **MEM-SGD** has the same asymptotic convergence rate with uncompressed SGD, and more importantly, is robust to both biased and unbiased compression operators. This method can be combined with decentralized training (Vogels et al., 2020), local SGD (Xie et al., 2020), and accelerated algorithms (Gorbunov et al., 2020). Due to its promising efficiency, error compensation has been applied into many related area (Zheng et al., 2019b; Phuong & Phong, 2020; Yu et al., 2019; Shi et al., 2019; Ivkin et al., 2019; Sun et al., 2019; Basu et al., 2019; Vogels et al., 2019) to reduce the communication cost.

Another direction to minimize the side-effect of compression is to compress the difference between the current gradient and momentum. Previous work (Mishchenko et al., 2019; Liu et al., 2020) proved that this strategy achieves linear convergence when the loss function is strongly convex. However, the linear convergence requires either the full deterministic gradient, or using SVRG (Horváth et al., 2019) to control the variance when for finite sum the loss function. Therefore how they perform with general variance reduced stochastic algorithms is still an open problem.

3 Algorithm Design

3.1 Background

The underlying problem we consider can be posed as the following distributed optimization problem:

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\mathbb{E}_{\xi^{(i)}\sim\mathcal{D}_i} F\left(\boldsymbol{x};\xi^{(i)}\right)}_{:=f_i(\boldsymbol{x})}, \tag{1}$$

where n is the number of workers, $\mathcal{D}_i$ is the local data distribution for worker i (in other words, we do not assume that all nodes can access the same data set), and $F(\boldsymbol{x};\xi^{(i)})$ is the local loss function of model $\boldsymbol{x}$ given data $\xi^{(i)}$ for worker i.

One widely used method for solving (1) is SGD, which updates the model using $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma\boldsymbol{g}_t$, where $\boldsymbol{x}_t$ is the model at the t-th iteration, γ is the learning rate and $\boldsymbol{g}_t$ is the averaged stochastic gradients. However, SGD suffers from the problem of large stochastic gradient variance, and many variance reduced algorithms are developed. In our paper, we focus on the following variance reduced algorithms: Momentum SGD, STROM, ROOT-SGD[1], and IGT, because they all construct the estimator using a moving average of previous stochastic gradients, which can be summarized as

$$\boldsymbol{v}_t = (1-\alpha_t)\boldsymbol{v}_{t-1} + \alpha_t\mathscr{A}(\boldsymbol{x}_t;\xi_t), \quad \boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma\boldsymbol{v}_t, \tag{2}$$

where $\boldsymbol{v}_t$ is the gradient estimator, $\mathscr{A}(\boldsymbol{x}_t;\xi_t)$ is a variable that depends on the history models $\boldsymbol{x}_s$ (for all $s \le t$) and data sample ξ_t, and α_t is a scalar. Notice that $\mathscr{A}(\boldsymbol{x}_t;\xi_t)$ and α_t are designed differently for different algorithms. We list the different choices of those parameters for each algorithm in Table 1.

	α_t	$\mathscr{A}(\boldsymbol{x}_t;\xi_t)$
SGD	1	$\nabla F(\boldsymbol{x}_t;\xi_t)$
Momentum SGD	α	$\nabla F(\boldsymbol{x}_t;\xi_t)$
STORM	α	$\frac{1}{\alpha_t}\left(\nabla F(\boldsymbol{x}_t;\xi_t) - (1-\alpha_t)\nabla F(\boldsymbol{x}_{t-1};\xi_t)\right)$
ROOT-SGD	$1/t$	$\frac{1}{\alpha_t}\left(\nabla F(\boldsymbol{x}_t;\xi_t) - (1-\alpha_t)\nabla F(\boldsymbol{x}_{t-1};\xi_t)\right)$
IGT	α	$\nabla F\left(\boldsymbol{x}_t + \frac{1-\alpha_t}{\alpha_t}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1});\xi_t\right)$

Table 1: Different choices of $(\alpha_t, \mathscr{A}(\boldsymbol{x}_t;\xi_t))$ for each variance reduced algorithm.

Recent work suggests that instead of compressing the gradient directly (Alistarh et al., 2017), using error compensation (Stich et al., 2018b) could potentially improve the convergence speed. The idea of error compensation is quite straightforward: adding back the compression error from the previous step to compensate the side-effect of compression. Denoting $C_\omega[\cdot]$[2] as the compressing operator, the updating rule of this method follows

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma C_\omega[\boldsymbol{g}_t + \boldsymbol{\delta}_{t-1}], \quad \boldsymbol{\delta}_t = \boldsymbol{g}_t + \boldsymbol{\delta}_{t-1} - C_\omega[\boldsymbol{g}_t + \boldsymbol{\delta}_{t-1}],$$

where $\boldsymbol{\delta}_t$ is the compression error at the t-th step. Moreover, recent works (Chen et al., 2020; Wu et al., 2018) find that adding a low-pass filter to the history compression error could be helpful for stabilizing the training and further improving the training speed. Their updating rule can be summarized as

$$\boldsymbol{e}_t = (1-\beta)\boldsymbol{e}_{t-1} + \beta\boldsymbol{\delta}_{t-1}, \quad \boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma C_\omega[\boldsymbol{g}_t + \boldsymbol{e}_t], \quad \boldsymbol{\delta}_t = \boldsymbol{g}_t + \boldsymbol{e}_t - C_\omega[\boldsymbol{g}_t + \boldsymbol{e}_t],$$

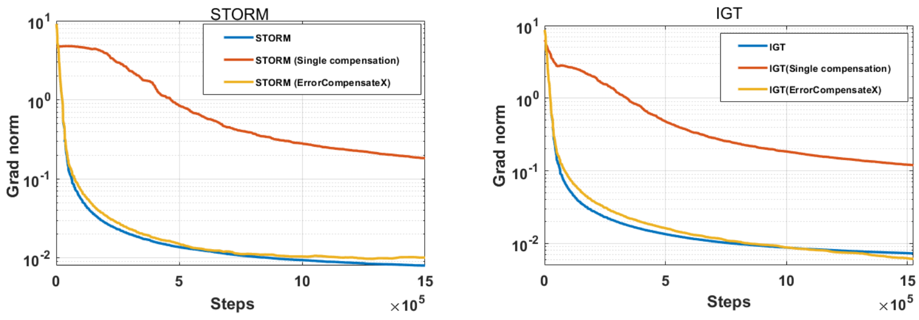
where β is a hyper-parameter of the low-pass filter.

Notice that even error compensated compression has been proved to be very powerful for accelerating the compressed training, it was only studied for SGD, which is a special case of (2) when $\alpha_t = 1$. For the case where $\alpha_t < 1$, previous work (Zheng et al., 2019a; Zhao et al., 2019; Wang et al., 2020) adapted the same idea from Stich et al. (2018b), which directly compresses the gradient estimator

[1] STORM (for non-convex loss function) and ROOT-SGD (strongly-convex loss function) use the same updating rule except $\alpha_t = 1/T^{\frac{2}{3}}$ for STORM and $\alpha_t = 1/T$ for ROOT-SGD.

[2] Here ω denotes the randomness of the compression operator.

$\boldsymbol{v}_t$ with the compression error from the previous step being compensated (we refer this method as **Single Compensation**). However, we find that when α_t is very small ($\mathcal{O}\left(1/T\right)$ for ROOT-SGD and $\mathcal{O}\left(1/T^{\frac{2}{3}}\right)$ for STORM, where T is the total training iterations), Single Compensation would be much slower than the uncompressed one, as shown in Figure 1.



(a)

(b)

Figure 1: Convergence speed comparison on linear regression for STORM and IGT with different compression techniques. The x-axis is the training step number, and the y-axis is the norm of the full gradient. The batch size equals 1, $\alpha_t = 1/t$, and we use 1-bit compression as described in Tang et al. (2019). Single Compensation means only compensate the compression error from the last step, and we can see that it admits a much slower convergence speed than the uncompressed one when α_t is very small. However, our proposed ErrorCompensatedX has a similar convergence rate as the uncompressed one.

3.2 Proposed ErrorCompensatedX

In this section, we provide the intuition behind our proposed method ErrorCompensatedX. For simplicity, we consider the general update in (2) with $\alpha_t = \alpha$. Let $\boldsymbol{v}_{-1} = 0$, then the uncompressed algorithm has the update

$$\boldsymbol{x}_T = \boldsymbol{x}_0 - \gamma \sum_{t=0}^{T-1} \sum_{s=0}^{t} \alpha (1-\alpha)^{t-s} \mathscr{A}(\boldsymbol{x}_s; \xi_s).$$

Note $\mathscr{A}(\boldsymbol{x}_s; \xi_s)$ is transferred from worker nodes to the server (here we aggregate the data from all workers). If the compression is applied on $\mathscr{A}(\boldsymbol{x}_s; \xi_s)$, then we have the update

$$\begin{aligned}
\boldsymbol{x}_T =& \boldsymbol{x}_0 - \gamma \sum_{t=0}^{T-1} \sum_{s=0}^{t} \alpha (1-\alpha)^{t-s} (\mathscr{A}(\boldsymbol{x}_s; \xi_s) - \boldsymbol{\delta}_s) \\
=& \boldsymbol{x}_0 - \gamma \sum_{t=0}^{T-1} \sum_{s=0}^{t} \alpha (1-\alpha)^{t-s} \mathscr{A}(\boldsymbol{x}_s; \xi_s) + \gamma \sum_{s=0}^{T-1} (1 - (1-\alpha)^{T-s}) \boldsymbol{\delta}_s.
\end{aligned}$$

Note that the sequence $\{\boldsymbol{x}_s\}_{s=0}^{T}$ is different from that of the uncompressed one. Here $\boldsymbol{\delta}_s$ is the compression error, and the compressed data $\mathscr{A}(\boldsymbol{x}_s; \xi_s) - \boldsymbol{\delta}_s$ is transferred to the server. With error compensation and additional $\boldsymbol{\delta}_{-1} = 0$, we have

$$\begin{aligned}
\boldsymbol{x}_T =& \boldsymbol{x}_0 - \gamma \sum_{t=0}^{T-1} \sum_{s=0}^{t} \alpha (1-\alpha)^{t-s} (\mathscr{A}(\boldsymbol{x}_s; \xi_s) + \boldsymbol{\delta}_{s-1} - \boldsymbol{\delta}_s) \\
=& \boldsymbol{x}_0 - \gamma \sum_{t=0}^{T-1} \sum_{s=0}^{t} \alpha (1-\alpha)^{t-s} \mathscr{A}(\boldsymbol{x}_s; \xi_s) + \gamma \sum_{s=0}^{T-1} \alpha (1-\alpha)^{T-1-s} \boldsymbol{\delta}_s.
\end{aligned}$$

Here $\boldsymbol{\delta}_s$ is the compression error that occurs while compressing $\mathscr{A}(\boldsymbol{x}_s; \xi_s) + \boldsymbol{\delta}_{s-1}$. When $\alpha = 1$, the last term disappears, and the standard SGD case has been analyzed in Tang et al. (2019); Stich et al.

(2018a). In this paper, we focus on the case where the last term does not disappear, i.e., $\alpha \in (0, 1)$. Given $\boldsymbol{\delta}_{-1} = \boldsymbol{\delta}_{-2} = 0$, when using ErrorCompensatedX, we have

$$\begin{aligned}
\boldsymbol{x}_T =& \boldsymbol{x}_0 - \gamma \sum_{t=0}^{T-1} \sum_{s=0}^{t} \alpha(1-\alpha)^{t-s}(\mathscr{A}(\boldsymbol{x}_s; \xi_s) + (1-\alpha)(\boldsymbol{\delta}_{s-1} - \boldsymbol{\delta}_{s-2}) + \boldsymbol{\delta}_{s-1} - \boldsymbol{\delta}_s) \\
=& \boldsymbol{x}_0 - \gamma \sum_{t=0}^{T-1} \sum_{s=0}^{t} \alpha(1-\alpha)^{t-s} \mathscr{A}(\boldsymbol{x}_s; \xi_s) + \alpha\gamma\boldsymbol{\delta}_{T-1}.
\end{aligned}$$

Notice that in the discussion above, we make an implicit assumption that there is only one worker in the training. For the case of multiple workers, we adapt the same strategy proposed by Tang et al. (2019), which uses a Parameter-Server communication prototype and applies error compensation for both rounds of worker-server communication.

For changing α_t, the updating rule of ErrorCompensatedX with a low-pass filter follows

$$\begin{aligned}
\boldsymbol{e}_t =& (1-\beta)\boldsymbol{e}_{t-1} + \beta\left(\frac{\alpha_{t-1}}{\alpha_t}(2-\alpha_t)\boldsymbol{\delta}_{t-1} - \frac{\alpha_{t-2}}{\alpha_t}(1-\alpha_t)\boldsymbol{\delta}_{t-2}\right), \quad \Delta_t = \mathscr{A}(\boldsymbol{x}_t; \xi_t) + \boldsymbol{e}_t, \\
\boldsymbol{v}_t =& (1-\alpha_t)\boldsymbol{v}_{t-1} + \alpha_t C_\omega[\Delta_t], \quad \boldsymbol{\delta}_t = \Delta_t - C_\omega[\Delta_t], \quad \boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma\boldsymbol{v}_t.
\end{aligned} \tag{3}$$

Here $\boldsymbol{e}_t$ is the compression error being passed after the low-pass filter and β is the hyper-parameter of the low-pass filter. Our analysis indicates that for a general case with $0 < \alpha_t \leq 1$, using just the compression error from the last step is not good enough, especially when α_t is very small. In this case, ErrorCompensatedX admits a much faster convergence speed than Single Compensation and could achieve the same asymptotic speed with the uncompressed one. We assume a Parameter-Server communication prototype for the parallel implementation of ErrorCompensatedX, and the detailed algorithm description can be find in Algorithm 1.

Algorithm 1 ErrorCompensatedX for general $\mathscr{A}(\boldsymbol{x}; \xi)$

```
Input: Initialize $\boldsymbol{x}_0$, learning rate $\gamma$, loss-pass filter parameter $\beta$, initial error on workers $\boldsymbol{\delta}_{-1}^{(i)} = \boldsymbol{\delta}_0^{(i)} = \boldsymbol{0}$, initial error on the server $\boldsymbol{\delta}_{-1} = \boldsymbol{\delta}_0 = \boldsymbol{0}$, scheduler of $\{\alpha_t\}_{t=-1}^{T}$, and number of total iterations $T$. The initial gradient estimator $\boldsymbol{v}_0 = \nabla F\left(\boldsymbol{x}_0; \xi_0^{(i)}\right)$ with total batch-size $B_0$, initial compression error buffer $\boldsymbol{e}_0^{(i)} = \boldsymbol{0}$ and $\boldsymbol{e}_0 = \boldsymbol{0}$.
for $t = 1, 2, \cdots, T-1$ do
    On worker $i$:
        Pass the worker compression error into the low-pass filter: $\boldsymbol{e}_t^{(i)} = (1-\beta)\boldsymbol{e}_{t-1}^{(i)} + \beta\left(\frac{\alpha_{t-1}}{\alpha_t}(2-\alpha_t)\boldsymbol{\delta}_{t-1}^{(i)} - \frac{\alpha_{t-2}}{\alpha_t}(1-\alpha_t)\boldsymbol{\delta}_{t-2}^{(i)}\right)$.
        Compute the error-compensated local gradient estimator: $\Delta_t^{(i)} = \mathscr{A}\left(\boldsymbol{x}_t; \xi_t^{(i)}\right) + \boldsymbol{e}_t^{(i)}$.
        Compress $\Delta_t^{(i)}$ into $C_\omega\left[\Delta_t^{(i)}\right]$ and update the local worker error $\boldsymbol{\delta}_t^{(i)} \leftarrow \Delta_t^{(i)} - C_\omega\left[\Delta_t^{(i)}\right]$.
        Send $C_\omega\left[\Delta_t^{(i)}\right]$ to the parameter server.
    On parameter server:
        Pass the server compression error into the low-pass filter: $\boldsymbol{e}_t = (1-\beta)\boldsymbol{e}_{t-1} + \beta\left(\frac{\alpha_{t-1}}{\alpha_t}(2-\alpha_t)\boldsymbol{\delta}_{t-1} - \frac{\alpha_{t-2}}{\alpha_t}(1-\alpha_t)\boldsymbol{\delta}_{t-2}\right)$.
        Average all gradient estimator received from workers: $\Delta_t = \frac{1}{n}\sum_{i=1}^{n} C_\omega\left[\Delta_t^{(i)}\right] + \boldsymbol{e}_t$.
        Compress $\Delta_t$ into $C_\omega[\Delta_t]$ and update the server error $\boldsymbol{\delta}_t = \Delta_t - C_\omega[\Delta_t]$
        Send $C_\omega[\Delta_t]$ to workers
    On worker $i$:
        Update the gradient estimator $\boldsymbol{v}_t = (1-\alpha_t)\boldsymbol{v}_{t-1} + \alpha_t C_\omega[\Delta_t]$.
        Update the model $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma\boldsymbol{v}_t$.
end for
Output: $\boldsymbol{x}_T$
```

Throughout this paper and its supplementary document, we let $f^* = \min_{\boldsymbol{x}} f(\boldsymbol{x})$ be the optimal objective value. In addition, we let $\|\cdot\|$ denote the l_2 norm of vectors, and $\lesssim$ means "less than or equal to up to a constant factor".

4 Theoretical Analysis

In this section, we first summarize the general updating rule for each algorithm w/o error compensation. Then we will present a general theorem that is essential for getting the convergence rate of each algorithm. At last, we will see that by using ErrorCompensatedX, we could achieve the same asymptotic convergence rate with training without communication compression using both biased and unbiased compression.

All error compensation algorithms mentioned so far can be summarized into the formulation of

$$\begin{aligned} \boldsymbol{e}_t =& (1-\beta)\boldsymbol{e}_{t-1} + \beta\left(c_{1,t}\boldsymbol{\delta}_{t-1} - c_{2,t}\boldsymbol{\delta}_{t-2}\right), \qquad (4)\\ \boldsymbol{v}_t =& (1-\alpha_t)\boldsymbol{v}_{t-1} + \alpha_t\mathscr{A}(\boldsymbol{x}_t;\xi_t) - \eta_{1,t}\boldsymbol{\delta}_t + \eta_{2,t}\boldsymbol{e}_t, \quad \boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma\boldsymbol{v}_t. \qquad (5) \end{aligned}$$

Here $(1-\alpha_t)\boldsymbol{v}_{t-1} + \alpha_t\mathscr{A}(\boldsymbol{x}_t;\xi_t)$ originates from the original uncompressed algorithm. Other terms that depends on the compression error $\boldsymbol{\delta}_t$ indicates the influence of compression (see our supplementary material for more details about the parameter setting for different compression algorithms.)

4.1 Convergence Rate of General Algorithms

In this section, we decompose the convergence rate of ErrorCompensatedX into two parts: $\mathcal{R}_{\text{uncompressed}}(T)$ that only depends on the original algorithm without compression, and $\mathcal{R}_\epsilon$ which only depends on the way of error compensation.

Instead of investigating $\boldsymbol{x}_t$ directly, we introduce an auxiliary sequence $\{\hat{\boldsymbol{x}}_t\}$, which is defined as

$$\boldsymbol{u}_t = (1-\alpha_t)\boldsymbol{u}_{t-1} + \alpha_t\mathscr{A}(\boldsymbol{x}_t;\xi_t), \quad \hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_{t-1} - \gamma\boldsymbol{u}_t, \qquad (6)$$

and set $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0$, $\boldsymbol{u}_0 = \boldsymbol{v}_0$ for initialization. Below we are going to see that $\{\hat{\boldsymbol{x}}_t\}$ admits some nice properties and is very helpful for the analysis of the convergence rate for $\{\boldsymbol{x}_t\}$.

For $\hat{\boldsymbol{x}}_t$, there seems to be no compression error, but we still compute the gradient estimator using $\boldsymbol{x}_t$ instead $\hat{\boldsymbol{x}}_t$ directly. Therefore we also define

$$\hat{\boldsymbol{u}}_0 = \mathscr{A}(\boldsymbol{x}_0;\xi_0) = \mathscr{A}(\hat{\boldsymbol{x}}_0;\xi_0), \quad \hat{\boldsymbol{u}}_t = (1-\alpha_t)\hat{\boldsymbol{u}}_{t-1} + \alpha_t\mathscr{A}(\hat{\boldsymbol{x}}_t;\xi_t). \qquad (7)$$

In this case, in the view of $\{\hat{\boldsymbol{x}}_t\}$, it basically updates the model following

$$\hat{\boldsymbol{u}}_t = (1-\alpha_t)\hat{\boldsymbol{u}}_{t-1} + \alpha_t\mathscr{A}(\hat{\boldsymbol{x}}_t;\xi_t), \quad \hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_{t-1} - \gamma\hat{\boldsymbol{u}}_t - \underbrace{\gamma(\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t)}_{\textbf{compression bias}}.$$

Therefore $\{\hat{\boldsymbol{x}}_t\}$ is essentially using the same uncompressed gradient estimator $\mathscr{A}(\hat{\boldsymbol{x}}_t;\xi_t)$ for constructing the gradient estimator $\hat{\boldsymbol{u}}_t$ except that when it updates the model, it will use $\hat{\boldsymbol{u}}_t$ plus a compression bias term $\gamma(\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t)$.

Now we present the key theorem that concludes the convergence rate of any algorithm (for non-convex loss function) that updates the model according to (4) and (5). Before we introduce the final theorem, we first make some commonly used assumptions:

Assumption 1. *We assume $f^* > -\infty$ and make the following assumptions:*

1. ***Lipschitzian gradient:*** *$f(\cdot)$ and $F(\cdot;\xi)$ are assumed to be with L-smooth, which means*

$$\begin{aligned} \|\nabla F(\boldsymbol{x};\xi) - \nabla F(\boldsymbol{y};\xi)\| \leq& L_F\|\boldsymbol{x}-\boldsymbol{y}\|, \quad \forall \boldsymbol{x},\boldsymbol{y},\xi,\\ \|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq& L\|\boldsymbol{x}-\boldsymbol{y}\|, \quad \forall \boldsymbol{x},\boldsymbol{y},i; \end{aligned}$$

2. ***Bounded variance:*** *The variance of the stochastic gradient is bounded*

$$\mathbb{E}_{\xi^{(i)}\sim\mathcal{D}_i}\|\nabla F(\boldsymbol{x};\xi^{(i)}) - \nabla f(\boldsymbol{x})\|^2 \leq \sigma^2, \quad \forall \boldsymbol{x}, i.$$

Theorem 1. *For algorithms that follow the updating rule* (4) *and* (5)*, under Assumption 1, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \leq \underbrace{\frac{16(f(\boldsymbol{x}_0)-f^*)}{\gamma T}+\frac{8}{T}\sum_{t=0}^{T-1}A_t}_{\mathcal{R}_{uncompressed}(T)}+\underbrace{\frac{64L_{\mathscr{A}}^2+2L^2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t\|^2}_{\mathcal{R}_\epsilon},$$

where A_t is defined as

$$A_t := \mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)-\hat{\boldsymbol{u}}_t\|^2-(1-2L\gamma)\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2-\frac{\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2}{4}$$

with $\hat{\boldsymbol{u}}_t$ defined in (7)*, and $L_{\mathscr{A}}$ is the Lipschitz constant of $\mathscr{A}(\cdot;\xi)$, i.e.,*

$$\|\mathscr{A}(\boldsymbol{x};\xi)-\mathscr{A}(\boldsymbol{y};\xi)\| \leq L_{\mathscr{A}}\|\boldsymbol{x}-\boldsymbol{y}\|, \quad \forall \boldsymbol{x},\boldsymbol{y},\xi.$$

Here A_t, which depends only on the original uncompressed algorithm, indicates the bias of constructing the gradient estimator using uncompressed $\mathscr{A}(\hat{\boldsymbol{x}}_t;\xi_t)$, hence $\frac{16(f(\boldsymbol{x}_0)-f^*)}{\gamma T}+\frac{8}{T}\sum_{t=0}^{T}A_t$ is usually the convergence rate of the original algorithm without compression.

In order to get a clear vision about the influence of the compression error under different error compensation methods, in the theorem below we are going to give an upper bound of $\sum_{t=0}^{T}\mathbb{E}\|\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t\|^2$ for different error compensation methods.

Assumption 2. *The magnitude of the compression error are assumed to be bounded by a constant ϵ:*

$$\mathbb{E}_\omega\left\|\boldsymbol{\delta}_t^{(i)}\right\|^2 \leq \frac{\epsilon^2}{2}, \quad \mathbb{E}_\omega\left\|\boldsymbol{\delta}_t^{(i)}\right\|^2 \leq \frac{\epsilon^2}{2}, \forall t, i.$$

Theorem 2. *For algorithms that follow the updating rule* (4) *and* (5)*, under Assumption 2, setting $\beta=1$, $\eta_{1,t}=\eta_1$, $\eta_{2,t}=\eta_2$, $c_{1,t}=c_1$, and $c_{2,t}=c_2$, we have*

$$\begin{aligned}\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t = & -\frac{\eta_1}{\alpha}\sum_{s=0}^{t}\left(1-(1-\alpha)^{t-s+1}\right)\boldsymbol{\delta}_s+\frac{\eta_2 c_1}{\alpha}\sum_{s=0}^{t-1}\left(1-(1-\alpha)^{t-s}\right)\boldsymbol{\delta}_s \\ & +\frac{\eta_2 c_2}{\alpha}\sum_{s=0}^{t-2}\left(1-(1-\alpha)^{t-s-1}\right)\boldsymbol{\delta}_s.\end{aligned}$$

More specifically, we have

- ***No Compensation:*** $\sum_{t=0}^{T}\mathbb{E}\|\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t\|^2 \leq \frac{\gamma^2T^2\epsilon^2}{\alpha^2}$;
- ***Single Compensation:*** $\sum_{t=0}^{T}\mathbb{E}\|\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t\|^2 \leq \frac{\gamma^2\epsilon^2}{\alpha^2}$;
- ***ErrorCompensatedX:*** $\sum_{t=0}^{T}\mathbb{E}\|\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t\|^2 \leq \gamma^2\alpha^2\epsilon^2$.

4.2 Convergence Rate for Different Gradient Estimators

In this section, we apply Theorem 1 to get the specific convergence rate of ErrorCompensatedX for Momentum SGD, STORM, and IGT.

4.2.1 SGD and Momentum SGD

Since SGD is a special case of Momentum SGD when setting $\alpha=1$, the following theorem includes both SGD and Momentum SGD.

Theorem 3. *Setting $\mathscr{A}(\boldsymbol{x}_t;\xi_t)=\nabla F(\boldsymbol{x}_t;\xi_t)$, if $L\gamma \leq \frac{\alpha}{12}$, under Assumptions 1 and 2, we have*

$$\sum_{t=0}^{T-1}A_t \leq \frac{17\gamma L\sigma^2T}{3n}.$$

This leads us to the corollary below:

Corollary 4. *For ErrorCompensatedX, under Assumptions 1 and 2, setting* $\alpha_t = \alpha \in (0, 1]$, $\mathscr{A}(\boldsymbol{x}_t; \xi_t) = \nabla F(\boldsymbol{x}_t; \xi_t)$ *and* $\gamma = \min\left\{\frac{\alpha}{12L}, \sqrt{\frac{n}{T\sigma^2}}, \left(\frac{1}{\epsilon^2 T}\right)^{1/3}\right\}$*, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \lesssim \frac{\sigma}{\sqrt{nT}} + \frac{\alpha^2}{(\epsilon T)^{\frac{2}{3}}} + \frac{1}{\alpha T}.$$

The leading term admits the order of $\mathcal{O}\left(1/\sqrt{nT}\right)$, which is the same as uncompressed training.

4.2.2 STOchastic Recursive Momentum (STORM)

Theorem 5. *Setting* $\mathscr{A}(\boldsymbol{x}_t; \xi_t) = \frac{1}{\alpha_t}\left(\nabla F(\boldsymbol{x}_t; \xi_t) - (1-\alpha_t)\nabla F(\boldsymbol{x}_{t-1}; \xi_t)\right)$*, if* $\gamma \leq \frac{1}{4L}$ *and* $\alpha \geq \frac{8(L^2 + L_F^2)\gamma^2}{n}$*, under Assumptions 1 and 2, we have*

$$\sum_{t=0}^{T-1} A_t \leq \frac{2\alpha\sigma^2 T}{n} + \frac{\sigma^2}{n\alpha B_0}.$$

This leads us to the corollary below:

Corollary 6. *For ErrorCompensatedX, under Assumptions 1 and 2, setting* $\gamma = \min\left\{\frac{1}{4L}, \left(\frac{n^2}{\sigma^2 T}\right)^{\frac{1}{3}}, \left(\frac{n}{\epsilon^2 T}\right)^{\frac{1}{7}}\right\}$, $\alpha_t = \frac{8L^2\gamma^2}{n}$, $B_0 = \frac{\sigma^{\frac{8}{3}} T^{\frac{1}{3}}}{n^{\frac{2}{3}}}$*, and* $\mathscr{A}(\boldsymbol{x}_t; \xi_t) = \frac{1}{\alpha_t}\left(\nabla F(\boldsymbol{x}_t; \xi_t) - (1-\alpha_t)\nabla F(\boldsymbol{x}_{t-1}; \xi_t)\right)$*, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \lesssim \left(\frac{\sigma}{nT}\right)^{\frac{2}{3}} + \left(\frac{\epsilon^2}{nT^6}\right)^{\frac{1}{7}} + \frac{1}{T}.$$

The leading term admits the order of $\mathcal{O}\left(1/(nT)^{\frac{2}{3}}\right)$, which is the same as uncompressed training (Yuan et al., 2020).

4.2.3 Implicit Gradient Transport (IGT)

For IGT, we need to make some extra assumptions (as listed below) for theoretical analysis.

Assumption 3. *The Hessian* $\nabla^2 f(\cdot)$ *is* ρ*-Lipschitz continuous, i.e., for every* $\boldsymbol{x}, \ \boldsymbol{y} \in \mathbb{R}^d$,

$$\left\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{y})\right\| \leq \rho\|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y}.$$

The magnitude of the stochastic gradient is upper bounded, i.e.,

$$\mathbb{E}\|\nabla F(\boldsymbol{x}; \xi) - \nabla F(\boldsymbol{y}; \xi)\|^2 \leq \Delta^2, \quad \forall \boldsymbol{x}, \boldsymbol{y}.$$

Now we are ready to present the result of IGT:

Theorem 7. *Setting* $\mathscr{A}(\boldsymbol{x}_t; \xi_t) = \nabla F\left(\boldsymbol{x}_t + \frac{1-\alpha_t}{\alpha_t}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}); \xi_t\right)$*, if* $\gamma \leq \frac{1}{2L}$*, under Assumptions 1, 2 and 3, we have*

$$\sum_{t=0}^{T-1} A_t \leq \frac{\alpha\sigma^2 T}{n} + \frac{\sigma^2}{\alpha n B_0} + \frac{\rho^2\gamma^4\Delta^4 T}{\alpha^4}.$$

This leads us to the corollary below:

Corollary 8. *For ErrorCompensatedX, under Assumptions 1, 2 and 3, by setting* $\gamma = \min\left\{\frac{1}{2L}, \left(\frac{n^4}{\sigma^8 T^5}\right)^{\frac{1}{9}}, \left(\frac{n}{\epsilon^2 T}\right)^{\frac{1}{7}}\right\}$, $\alpha = \left(\frac{n^5}{\sigma^8 T^4}\right)^{\frac{1}{9}}$ *and* $B_0 = 1$*, we have*

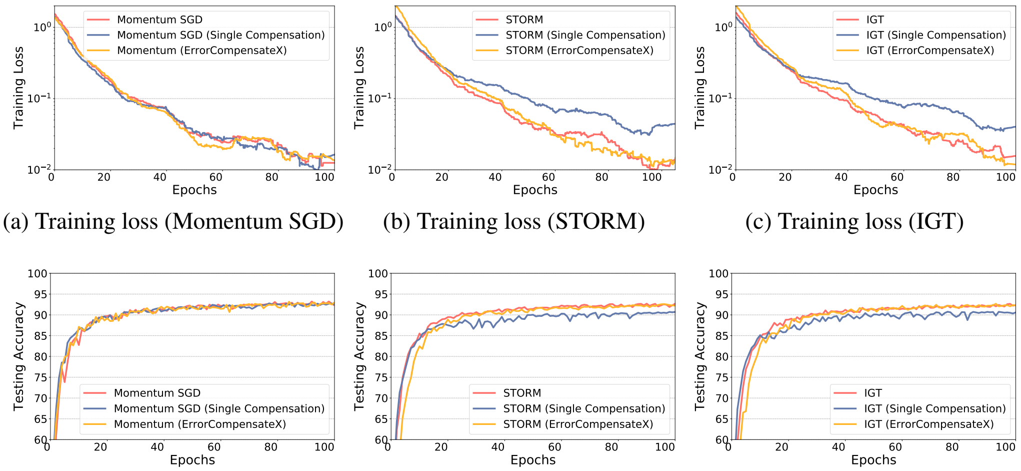
$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \lesssim \left(\frac{\sigma^8}{n^4 T^4}\right)^{\frac{1}{9}} + \left(\frac{\epsilon^2}{nT^6}\right)^{\frac{1}{7}} + \frac{1}{T}.$$

The leading term admits the order of $\mathcal{O}(1/(nT)^{\frac{4}{9}})$, which is even worse than that of Momentum SGD. A sharper rate of IGT for a general non-convex loss function is still an open problem.

5 Numerical Experiments

In this section, we train ResNet-50 (He et al., 2016) on CIFAR10, which consists of 50000 training images and 10000 testing images, each has 10 labels. We run the experiments on eight workers, each having a 1080Ti GPU. The batch size on each worker is 16 and the total batch size is 128. For each choice of $\mathscr{A}(\boldsymbol{x}_t;\xi_t)$, we evaluate four implementations: 1) **Original algorithm** without compression. 2) **No compensation**, which compresses the data directly without the previous information. 3) **Single compensation**, which compresses the gradient estimator with compression error from the last step added. 4) **ErrorCompensatedX**.

We use the 1-bit compression in Tang et al. (2019), which leads to an overall 96% of communication volume reduction. We find that for both STORM and IGT, setting $\alpha_t = \frac{1}{1+c_0 t}$, where c_0 is a constant and t is the training step count, would make the training faster. We grid search the best learning rate from $\{0.5, 0.1, 0.001\}$ and c_0 from $\{0.1, 0.05, 0.001\}$, and find that the best learning rate is 0.01 with $c_0 = 0.05$ for both original STORM and IGT. So we use this configuration for the other three implementations. For Momentum, usually we will not set α_t to be too small, therefore we only set $\alpha_t = 0.1$, and the best learning rate is 0.1 after the same grid search. We set $\beta = 0.3$ for the low-pass filter in all cases.



(a) Training loss (Momentum SGD)

(b) Training loss (STORM)

(c) Training loss (IGT)

(d) Testing accuracy (Momentum SGD)

(e) Testing accuracy (STORM)

(f) Testing accuracy (IGT)

Figure 2: Epoch-wise convergence comparison on ResNet-50 for Momenum SGD (left column), STORM (middle column), and IGT (right column) with different communication implementations. We do not include the result of no compensation because it diverges after a few steps of training.

As illustrated in Figure 2, for both STORM and IGT, ErrorCompensatedX achieves almost the same convergence speed with the original algorithm, while Single Compensation fails to achieve this. We find that with increasing magnitude of c_0, this gap of convergence speed would increase, which further validate our claim that ErrorCompensatedX is more essential when α_t gets small. For Momentum SGD, all the three implementations (except compression without error compensation) achieves similar convergence speed. This is because $\alpha_t = 0.1$ is comparably large.

6 Conclusion and Remarks

In this paper, we address an important problem for communication efficient distributed training: how to fully compensate the compression error for variance reduced optimization algorithms. In our paper, we consider a more general class of optimization algorithms (including SGD and Momentum SGD), and we propose a novel method: ErrorCompensatedX, which utilize the compression error from the last two steps, in order to fully compensate the history compression error while previous method fails to do. From the theoretical perspective, we provide a unified theoretical analysis framework that gives an intuitive evaluation for the side-effect of the compression, and shows that

ErrorCompensatedX admits the same asymptotic convergence rate with each of the original algorithm. Numerical experiments are implemented to show ErrorCompensatedX's convergence and its better performance comparing to other implementations.

Acknowledgments

Yao Li and Ming Yan are partially supported by the NSF grant DMS-2012439 and a Facebook Faculty Research Award.

References

Alistarh, D., Grubic, D., Li, J., Tomioka, R., & Vojnovic, M. (2017). Qsgd: Communication-efficient sgd via gradient quantization and encoding. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 30 (pp. 1709–1720).: Curran Associates, Inc.

Arnold, S. M. R., Manzagol, P.-A., Babanezhad, R., Mitliagkas, I., & Roux, N. L. (2019). Reducing the variance in online optimization by transporting past gradients.

Basu, D., Data, D., Karakus, C., & Diggavi, S. (2019). Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 14695–14706). Curran Associates, Inc.

Chen, C.-Y., Ni, J., Lu, S., Cui, X., Chen, P.-Y., Sun, X., Wang, N., Venkataramani, S., Srinivasan, V. V., Zhang, W., & Gopalakrishnan, K. (2020). Scalecom: Scalable sparsified gradient compression for communication-efficient distributed training. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems*, volume 33 (pp. 13551–13563).: Curran Associates, Inc.

Cutkosky, A. & Mehta, H. (2020). Momentum improves normalized SGD. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research* (pp. 2260–2268).: PMLR.

Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, volume 27 (pp. 1646–1654).: Curran Associates, Inc.

Gorbunov, E., Kovalev, D., Makarenko, D., & Richtárik, P. (2020). Linearly converging error compensated sgd.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).

Horváth, S., Kovalev, D., Mishchenko, K., Stich, S., & Richtárik, P. (2019). Stochastic distributed learning with gradient quantization and variance reduction.

Huang, F., Tao, L., & Chen, S. (2020). Accelerated stochastic gradient-free and projection-free methods. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research* (pp. 4519–4530).: PMLR.

Ivkin, N., Rothchild, D., Ullah, E., braverman, V., Stoica, I., & Arora, R. (2019). Communication-efficient distributed sgd with sketching. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 13144–13154). Curran Associates, Inc.

Johnson, R. & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, volume 26 (pp. 315–323).: Curran Associates, Inc.

Li, C. J., Mou, W., Wainwright, M. J., & Jordan, M. I. (2020). Root-sgd: Sharp nonasymptotics and asymptotic efficiency in a single algorithm.

Liu, X., Li, Y., Tang, J., & Yan, M. (2020). A double residual compression algorithm for efficient distributed learning. In S. Chiappa & R. Calandra (Eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research* (pp. 133–143).: PMLR.

Mishchenko, K., Gorbunov, E., Takac, M., & Richtarik, P. (2019). Distributed learning with compressed gradient differences.

Nguyen, L. M., Liu, J., Scheinberg, K., & Takac, M. (2017). Sarah: A novel method for machine learning problems using stochastic recursive gradient. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research* (pp. 2613–2621). International Convention Centre, Sydney, Australia: PMLR.

Phuong, T. T. & Phong, L. T. (2020). Distributed sgd with flexible gradient compression. *IEEE Access*, 8, 64707–64717.

Seide, F., Fu, H., Droppo, J., Li, G., & Yu, D. (2014). 1-bit stochastic gradient descent and application to data-parallel distributed training of speech dnns. In *Interspeech 2014*.

Shi, S., Wang, Q., Zhao, K., Tang, Z., Wang, Y., Huang, X., & Chu, X. (2019). A distributed synchronous sgd algorithm with global top-k sparsification for low bandwidth networks. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 2238–2247).

Stich, S. U., Cordonnier, J.-B., & Jaggi, M. (2018a). Sparsified sgd with memory. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 31* (pp. 4447–4458). Curran Associates, Inc.

Stich, S. U., Cordonnier, J.-B., & Jaggi, M. (2018b). Sparsified sgd with memory. In *Advances in Neural Information Processing Systems* (pp. 4447–4458).

Sun, J., Chen, T., Giannakis, G., & Yang, Z. (2019). Communication-efficient distributed learning via lazily aggregated quantized gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 3370–3380). Curran Associates, Inc.

Tang, H., Yu, C., Lian, X., Zhang, T., & Liu, J. (2019). Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *International Conference on Machine Learning* (pp. 6155–6165).: PMLR.

Vogels, T., Karimireddy, S. P., & Jaggi, M. (2019). Powersgd: Practical low-rank gradient compression for distributed optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 14259–14268). Curran Associates, Inc.

Vogels, T., Karimireddy, S. P., & Jaggi, M. (2020). Powergossip: Practical low-rank communication compression in decentralized deep learning.

Wang, J., Tantia, V., Ballas, N., & Rabbat, M. (2020). Slowmo: Improving communication-efficient distributed sgd with slow momentum. In *International Conference on Learning Representations*.

Wu, J., Huang, W., Huang, J., & Zhang, T. (2018). Error compensated quantized SGD and its applications to large-scale distributed optimization. *CoRR*, abs/1806.08054.

Xie, C., Zheng, S., Koyejo, O., Gupta, I., Li, M., & Lin, H. (2020). Cser: Communication-efficient sgd with error reset.

Yu, Y., Wu, J., & Huang, L. (2019). Double quantization for communication-efficient distributed optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 4438–4449). Curran Associates, Inc.

Yuan, H., Lian, X., Liu, J., & Zhou, Y. (2020). Stochastic recursive momentum for policy gradient methods.

Zhang, S., Choromanska, A., & LeCun, Y. (2015). Deep learning with elastic averaging sgd.

Zhao, S.-Y., Gao, H., & Li, W.-J. (2019). On the convergence of memory-based distributed sgd.

Zheng, S., Huang, Z., & Kwok, J. (2019a). Communication-efficient distributed blockwise momentum sgd with error-feedback. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 32 (pp. 11450–11460).: Curran Associates, Inc.

Zheng, S., Huang, Z., & Kwok, J. (2019b). Communication-efficient distributed blockwise momentum sgd with error-feedback. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 11450–11460). Curran Associates, Inc.

Supplementary

In the supplementary, we first prove the updating rule of ErrorCompensatedX. Then we will present the detailed proof for Theorem 1 followed by the detailed convergence rate result for each algorithm.

7 Updating rule of ErrorCompensatedX

The updating rule of Algorithm 1 admits the form

$$\begin{aligned}\overline{\boldsymbol{e}}_t =&(1-\beta)\overline{\boldsymbol{e}}_{t-1}+\beta\left(\frac{\alpha_{t-1}}{\alpha_t}(2-\alpha_t)\overline{\boldsymbol{\delta}}_{t-1}+\frac{\alpha_{t-2}}{\alpha_t}(1-\alpha_t)\overline{\boldsymbol{\delta}}_{t-2}\right),\\ \boldsymbol{v}_t =&(1-\alpha_t)\boldsymbol{v}_{t-1}+\alpha_t\overline{\boldsymbol{b}}_t(\boldsymbol{x}_t)-\alpha_t\overline{\boldsymbol{\delta}}_t+\alpha_t\overline{\boldsymbol{e}}_t,\\ \boldsymbol{x}_{t+1} =&\boldsymbol{x}_t-\gamma\boldsymbol{v}_t.\end{aligned}$$

where

$$\begin{aligned}\overline{\boldsymbol{e}}_t :=&\frac{1}{n}\sum_i \boldsymbol{e}_t^{(i)}+\boldsymbol{e}_t,\\ \overline{\boldsymbol{\delta}}_t :=&\boldsymbol{\delta}_t+\frac{1}{n}\sum_{i=1}^n\boldsymbol{\delta}_t^{(i)},\\ \overline{\boldsymbol{b}}_t(\boldsymbol{x}) :=&\frac{1}{n}\sum_i\mathscr{A}\left(\boldsymbol{x}_t;\xi_t^{(i)}\right).\end{aligned}$$

Proof. From the algorithm description in Algorithm 1, we shall see that we are essentially using $\boldsymbol{v}_t$ as the gradient estimator of the full gradient. From the updating rule of $\boldsymbol{v}_t$, we have

$$\begin{aligned}\boldsymbol{v}_t =&(1-\alpha_t)\boldsymbol{v}_{t-1}+\alpha_t C_\omega\left[\Delta_t\right]\\ =&(1-\alpha_t)\boldsymbol{v}_{t-1}+\alpha_t\left(\Delta_t-\boldsymbol{\delta}_t\right)\\ =&(1-\alpha_t)\boldsymbol{v}_{t-1}+\frac{\alpha_t}{n}\sum_i C_\omega\left[\Delta_t^{(i)}\right]+\alpha_t\boldsymbol{e}_t-\alpha_t\boldsymbol{\delta}_t\\ =&(1-\alpha_t)\boldsymbol{v}_{t-1}+\frac{\alpha_t}{n}\sum_i\left(\Delta_t^{(i)}-\boldsymbol{\delta}_t^{(i)}\right)+\alpha_t\boldsymbol{e}_t-\alpha_t\boldsymbol{\delta}_t\\ =&(1-\alpha_t)\boldsymbol{v}_{t-1}+\frac{\alpha_t}{n}\sum_i\left(\mathscr{A}\left(\boldsymbol{x}_t;\xi_t^{(i)}\right)+\boldsymbol{e}_t^{(i)}-\boldsymbol{\delta}_t^{(i)}\right)+\alpha_t\boldsymbol{e}_t-\alpha_t\boldsymbol{\delta}_t\\ =&(1-\alpha)\boldsymbol{v}_{t-1}+\alpha_t\overline{\boldsymbol{b}}_t(\boldsymbol{x}_t)+\alpha_t\overline{\boldsymbol{e}}_t-\alpha_t\overline{\boldsymbol{\delta}}_t.\end{aligned}$$

Notice that in the deduction above we continuously using the fact that $C_\omega[\boldsymbol{x}]=\boldsymbol{x}-\boldsymbol{\delta}$. For $\overline{\boldsymbol{e}}_t$, we have

$$\begin{aligned}\overline{\boldsymbol{e}}_t =&\frac{1}{n}\sum_i\boldsymbol{e}_t^{(i)}+\boldsymbol{e}_t\\ =&\frac{1-\beta}{n}\sum_i\boldsymbol{e}_{t-1}^{(i)}+\frac{\beta}{n}\sum_i\left(\frac{\alpha_{t-1}}{\alpha_t}(2-\alpha_t)\delta_{t-1}^{(i)}+\frac{\alpha_{t-2}}{\alpha_t}(1-\alpha_t)\delta_{t-2}^{(i)}\right)\\ &+(1-\beta)\boldsymbol{e}_{t-1}+\beta\left(\frac{\alpha_{t-1}}{\alpha_t}(2-\alpha_t)\delta_{t-1}+\frac{\alpha_{t-2}}{\alpha_t}(1-\alpha_t)\delta_{t-2}\right)\\ =&(1-\beta)\overline{\boldsymbol{e}}_{t-1}+\beta\left(\frac{\alpha_{t-1}}{\alpha_t}(2-\alpha_t)\overline{\boldsymbol{\delta}}_{t-1}+\frac{\alpha_{t-2}}{\alpha_t}(1-\alpha_t)\overline{\boldsymbol{\delta}}_{t-2}\right),\end{aligned}$$

which completes the proof. □

8 General Formulation of the Updating Rule

As mentioned in Section 4, all the error compensation mentioned so far admits the updating rule of (4) and (5). Therefore in Table 2 we list the specific choice of parameters for each algorithm.

	$\eta_{1,t}$	$\eta_{2,t}$	$c_{1,t}$	$c_{2,t}$
Without Compensation	1	0	0	0
Single Compensation	1	1	1	0
ErrorCompensatedX	α_t	α_t	$2-\alpha_t$	$1-\alpha_t$

Table 2: Different choices of $\eta_{1,t}$, $\eta_{2,t}$, $c_{1,t}$, and $c_{2,t}$ for each algorithm.

9 Proof of Theorems

9.1 Proof of Theorem 1

Proof. In the view of $\{\hat{\boldsymbol{x}}_t\}$, we have

$$
\begin{aligned}
&\mathbb{E}f(\hat{\boldsymbol{x}}_{t+1}) - \mathbb{E}f(\hat{\boldsymbol{x}}_t)\\
\le&\mathbb{E}\langle \nabla f(\hat{\boldsymbol{x}}_t), \hat{\boldsymbol{x}}_{t+1} - \hat{\boldsymbol{x}}_t\rangle + \frac{L}{2}\mathbb{E}\|\hat{\boldsymbol{x}}_{t+1} - \hat{\boldsymbol{x}}_t\|^2\\
=&-\gamma\mathbb{E}\langle \nabla f(\hat{\boldsymbol{x}}_t), \boldsymbol{u}_t\rangle + \frac{L\gamma^2}{2}\mathbb{E}\|\boldsymbol{u}_t\|^2\\
=&-\gamma\mathbb{E}\langle \nabla f(\hat{\boldsymbol{x}}_t), \hat{\boldsymbol{u}}_t + (\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t)\rangle + \frac{L\gamma^2}{2}\mathbb{E}\|\hat{\boldsymbol{u}}_t + (\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t)\|^2\\
\le&-\gamma\mathbb{E}\langle \nabla f(\hat{\boldsymbol{x}}_t), \hat{\boldsymbol{u}}_t\rangle - \gamma\mathbb{E}\langle \nabla f(\hat{\boldsymbol{x}}_t), \boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\rangle + L\gamma^2\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 + L\gamma^2\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2\\
\le&-\gamma\mathbb{E}\langle \nabla f(\hat{\boldsymbol{x}}_t), \hat{\boldsymbol{u}}_t\rangle + \frac{\gamma}{4}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2 + \gamma\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2 + L\gamma^2\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 + L\gamma^2\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2\\
\le&-\frac{\gamma}{2}\left(\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2 + \mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 - \mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t) - \hat{\boldsymbol{u}}_t\|^2\right) + \frac{\gamma}{4}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2\\
&+\gamma\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2 + L\gamma^2\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 + L\gamma^2\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2\\
\le&-\frac{\gamma}{8}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2 - \frac{\gamma}{2}\left((1-2L\gamma)\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 - \mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t) - \hat{\boldsymbol{u}}_t\|^2 + \frac{\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2}{4}\right)\\
&+(1+L\gamma)\gamma\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2.
\end{aligned}
$$

Summing up the inequality above from $t=0$ to $t=T$, with rearrangement we get

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2\\
\le&\frac{8(f(\hat{\boldsymbol{x}}_0) - f(\hat{\boldsymbol{x}}_T))}{\gamma T} - \frac{4}{T}\sum_{t=0}^{T-1}\left((1-2L\gamma)\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 - \mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t) - \hat{\boldsymbol{u}}_t\|^2 + \frac{\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2}{4}\right)\\
&+\frac{8+8L\gamma}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2\\
\le&\frac{8(f(\boldsymbol{x}_0) - f^*)}{\gamma T} - \frac{4}{T}\sum_{t=0}^{T-1}\left((1-2L\gamma)\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 - \mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t) - \hat{\boldsymbol{u}}_t\|^2 + \frac{\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2}{4}\right)\\
&+\frac{8+8L\gamma}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2. \qquad (8)
\end{aligned}
$$

In order to upper bound $\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2$, we have

$$
\begin{aligned}
&\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2\\
=&\mathbb{E}\|(1-\alpha)(\boldsymbol{u}_{t-1} - \hat{\boldsymbol{u}}_{t-1}) + \alpha(\mathscr{A}(\boldsymbol{x}_t;\xi_t) - \mathscr{A}(\hat{\boldsymbol{x}}_t;\xi_t))\|^2\\
\leq& (1+\alpha)\,\mathbb{E}\|(1-\alpha)(\boldsymbol{u}_{t-1} - \hat{\boldsymbol{u}}_{t-1})\|^2 + \left(1+\frac{1}{\alpha}\right)\mathbb{E}\|\alpha(\mathscr{A}(\boldsymbol{x}_t;\xi_t) - \mathscr{A}(\hat{\boldsymbol{x}}_t;\xi_t))\|^2\\
=& \left(1-\alpha^2\right)(1-\alpha)\mathbb{E}\|\boldsymbol{u}_{t-1} - \hat{\boldsymbol{u}}_{t-1}\|^2 + \left(\alpha^2+\alpha\right)\mathbb{E}\|\mathscr{A}(\boldsymbol{x}_t;\xi_t) - \mathscr{A}(\hat{\boldsymbol{x}}_t;\xi_t)\|^2\\
\leq& (1-\alpha)\,\mathbb{E}\|\boldsymbol{u}_{t-1} - \hat{\boldsymbol{u}}_{t-1}\|^2 + 2\alpha\mathbb{E}\|\mathscr{A}(\boldsymbol{x}_t;\xi_t) - \mathscr{A}(\hat{\boldsymbol{x}}_t;\xi_t)\|^2\\
=&(1-\alpha)^t\mathbb{E}\|\boldsymbol{u}_0 - \boldsymbol{u}_0(\boldsymbol{y})\|^2 + 2\alpha\sum_{s=0}^{t}(1-\alpha)^{t-s}\mathbb{E}\|\mathscr{A}(\boldsymbol{x}_s;\xi_s) - \mathscr{A}(\hat{\boldsymbol{x}}_s;\xi_s)\|^2\\
=&2\alpha\sum_{s=0}^{t}(1-\alpha)^{t-s}\mathbb{E}\|\mathscr{A}(\boldsymbol{x}_s;\xi_s) - \mathscr{A}(\hat{\boldsymbol{x}}_s;\xi_s)\|^2 \quad (\boldsymbol{u}_0 - \boldsymbol{u}_0(\boldsymbol{y}) = \boldsymbol{0})\\
\leq&2\alpha L_{\mathscr{A}}^2\sum_{s=0}^{t}(1-\alpha)^{t-s}\mathbb{E}\|\boldsymbol{x}_s - \hat{\boldsymbol{x}}_s\|^2.
\end{aligned}
$$

Therefore we have

$$
\begin{aligned}
\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_t\|^2 \leq&2\alpha L_{\mathscr{A}}^2\sum_{t=0}^{T-1}\sum_{s=0}^{t}(1-\alpha)^{t-s}\mathbb{E}\|\boldsymbol{x}_s - \hat{\boldsymbol{x}}_s\|^2\\
=&2\alpha L_{\mathscr{A}}^2\sum_{s=0}^{T-1}\sum_{t=s}^{T-1}(1-\alpha)^{t-s}\mathbb{E}\|\boldsymbol{x}_s - \hat{\boldsymbol{x}}_s\|^2\\
\leq&2L_{\mathscr{A}}^2\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|^2. \qquad (9)
\end{aligned}
$$

Combining (8) and (9) together we get

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2\\
\leq&\frac{8(f(\boldsymbol{x}_0) - f^*)}{\gamma T} - \frac{4}{T}\sum_{t=0}^{T}\left((1-2L\gamma)\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 - \mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t) - \hat{\boldsymbol{u}}_t\|^2 + \frac{\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2}{4}\right)\\
&+\frac{16(1+L\gamma)L_{\mathscr{A}}^2}{T}\sum_{t=0}^{T}\mathbb{E}\|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|^2.
\end{aligned}
$$

Hence $\|\nabla f(\boldsymbol{x}_t)\|^2$ can be upper bounded by using

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2\\
\le&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)+(\nabla f(\boldsymbol{x}_t)-\nabla f(\hat{\boldsymbol{x}}_t))\|^2\\
\le&\frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2+\frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)-\nabla f(\hat{\boldsymbol{x}}_t)\|^2\\
\le&\frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2+\frac{2L^2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t\|^2\\
\le&\frac{16(f(\boldsymbol{x}_0)-f^*)}{\gamma T}-\frac{8}{T}\sum_{t=0}^{T-1}\left((1-2L\gamma)\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2-\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)-\hat{\boldsymbol{u}}_t\|^2+\frac{\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2}{4}\right)\\
&+\frac{32(1+L\gamma)L_{\mathscr{A}}^2+2L^2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t-\hat{\boldsymbol{u}}_t\|^2.
\end{aligned}
$$

If $L\gamma\le 1$, we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2\\
\le&\frac{16(f(\boldsymbol{x}_0)-f^*)}{\gamma T}-\frac{8}{T}\sum_{t=0}^{T-1}\left((1-2L\gamma)\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2-\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)-\hat{\boldsymbol{u}}_t\|^2+\frac{\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2}{4}\right)\\
&+\frac{64L_{\mathscr{A}}^2+2L^2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t\|^2,
\end{aligned}
$$

which completes the proof. □

9.2 Proof of Theorem 3, 5 and 7

In this section, we are going to present the proof of theorems for different gradient estimators. We start with the key lemma for **Momentum SGD.**

Lemma 9. *For two nonnegative sequences $\{a_t\}$ and $\{b_t\}$ that satisfy*

$$a_t\le\rho a_{t-1}+b_t,$$

where $\rho\in(0,1)$ is a constant, we have

$$\sum_{t=0}^{T}a_t\le\frac{\sum_{t=1}^{T}b_t+a_0}{1-\rho}.$$

Proof. Since

$$a_t\le\rho a_{t-1}+b_t,$$

adding the inequality above from $t=1$ to $t=T$ we get

$$
\begin{aligned}
\sum_{t=1}^{T}a_t\le&\rho\sum_{t=0}^{T-1}a_t+\sum_{t=1}^{T}b_t\\
\le&\rho\sum_{t=0}^{T}a_t+\sum_{t=1}^{T}b_t.
\end{aligned}
$$

Adding a_0 to both side we get

$$\sum_{t=0}^{T} a_t \leq \rho \sum_{t=0}^{T} a_t + \sum_{t=1}^{T} b_t + a_0$$

and the proof is complete by combining the sum of a_t and dividing both sides by $1-\rho$. □

Proof of Theorem 3. For **Momentum SGD**, denote $\boldsymbol{m}_t = \alpha \sum_{s=0}^{t} (1-\alpha)^{t-s} \nabla f(\boldsymbol{x}_s)$, then we have

$$\begin{aligned}
&\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{u}_t\|^2 - \mathbb{E}\|\boldsymbol{u}_t\|^2 \\
=&\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 - 2\mathbb{E}\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t\rangle \\
=&\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 - 2\mathbb{E}\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{m}_t\rangle - 2\mathbb{E}\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t - \boldsymbol{m}_t\rangle \\
=&-\mathbb{E}\|\boldsymbol{m}_t\|^2 + \mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{m}_t\|^2 - 2\mathbb{E}\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t - \boldsymbol{m}_t\rangle. \qquad (10)
\end{aligned}$$

For $\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{m}_t\|^2$ we have

$$\begin{aligned}
&\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{m}_t\|^2 \\
=&\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - (1-\alpha)\boldsymbol{m}_{t-1} - \alpha f(\boldsymbol{x}_t)\|^2 \\
=&\mathbb{E}\|(1-\alpha)(\nabla f(\boldsymbol{x}_{t-1}) - \boldsymbol{m}_{t-1} + \nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_{t-1}))\|^2 \\
\leq& (1+\alpha)(1-\alpha)^2\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1}) - \boldsymbol{m}_{t-1}\|^2 + \left(1+\frac{1}{\alpha}\right)(1-\alpha)^2\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_{t-1}))\|^2 \\
\leq&(1-\alpha)\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1}) - \boldsymbol{m}_{t-1}\|^2 + \frac{1}{\alpha}\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_{t-1}))\|^2 \\
\leq&(1-\alpha)\mathbb{E}\|\nabla f(\boldsymbol{x}_{t-1}) - \boldsymbol{m}_{t-1}\|^2 + \frac{L^2\gamma^2}{\alpha}\mathbb{E}\|\boldsymbol{u}_t\|^2.
\end{aligned}$$

Therefore from Lemme 9,

$$\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{m}_t\|^2 \leq \frac{L^2\gamma^2}{\alpha^2} \sum_{t=0}^{T-1} \mathbb{E}\|\boldsymbol{u}_t\|^2. \qquad (11)$$

For $\mathbb{E}\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t - \boldsymbol{m}_t\rangle$, we have

$$\begin{aligned}
\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t - \boldsymbol{m}_t\right\rangle =&\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_t), (1-\alpha)(\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}) + \alpha(\boldsymbol{g}_t - \nabla f(\boldsymbol{x}_t))\right\rangle \\
=&\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_t), (1-\alpha)(\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1})\right\rangle \\
=&(1-\alpha)\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_{t-1}), \boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\right\rangle \\
&+ (1-\alpha)\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_{t-1}), \boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\right\rangle, \qquad (12)
\end{aligned}$$

where $\boldsymbol{g}_t = \nabla F(\boldsymbol{x}_t; \xi_t)$.

Notice that for $\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_{t-1})$, we have

$$\begin{aligned}
&\mathbb{E}\left\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_{t-1})\right\|^2 \\
\leq&L^2\gamma^2\mathbb{E}\|\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1} + \boldsymbol{m}_{t-1}\|^2 \\
\leq&2L^2\gamma^2\mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + 2L^2\gamma^2\mathbb{E}\|\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\|^2. \qquad (13)
\end{aligned}$$

Combining (12) and (13) together we get

$$\begin{aligned}
&\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t - \boldsymbol{m}_t\right\rangle \\
\leq&(1-\alpha)\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_{t-1}), \boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\right\rangle + (1-\alpha)\mathbb{E}\left(\frac{1}{2\gamma L}\left\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_{t-1})\right\|^2 + \frac{\gamma L}{2}\mathbb{E}\left\|\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\right\|^2\right) \\
\leq&(1-\alpha)\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_{t-1}), \boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\right\rangle + (1-\alpha)\mathbb{E}\left(\gamma L\mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + \frac{3\gamma L}{2}\mathbb{E}\left\|\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\right\|^2\right),
\end{aligned}$$

Denote $c_t := \gamma L\mathbb{E}\|\boldsymbol{m}_{t-1}\|^2 + \frac{3\gamma L}{2}\mathbb{E}\left\|\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\right\|^2$, then we get

$$\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t - \mathbb{E}\boldsymbol{u}_t\right\rangle = (1-\alpha)^t\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_0), \boldsymbol{u}_0 - \boldsymbol{m}_0\right\rangle + \sum_{s=1}^{t}(1-\alpha)^{t-s}c_s.$$

Since $\mathbb{E}\mathscr{A}(\boldsymbol{x}_0;\xi_0) = \mathbb{E}\nabla F(\boldsymbol{x}_0;\xi_0) = \nabla F(\boldsymbol{x}_0)$, which means $\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_0), \boldsymbol{u}_0 - \boldsymbol{m}_0 \right\rangle = 0$. So the equation above becomes

$$\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t - \boldsymbol{m}_t \right\rangle = \sum_{s=1}^{t}(1-\alpha)^{t-s}c_s,$$

and

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{u}_t - \mathbb{E}\boldsymbol{u}_t \right\rangle \leq & \frac{\sum_{t=0}^{T-1}|c_t|}{1-(1-\alpha)} \\ = & \frac{\gamma L}{\alpha}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{m}_t\|^2 + \frac{3\gamma L}{2\alpha}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t - \boldsymbol{m}_t\|^2. \end{aligned} \tag{14}$$

Combing (14), (11) and (10), we get

$$\begin{aligned}&\sum_{t=0}^{T-1}\left(\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{u}_t\|^2 - (1-2L\gamma)\mathbb{E}\|\boldsymbol{u}_t\|^2\right) \\ \leq & -\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{m}_t\|^2 + \left(\frac{L^2\gamma^2}{\alpha^2} + 2L\gamma\right)\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t\|^2 + \frac{2\gamma L}{\alpha}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{m}_t\|^2 + \frac{3\gamma L}{\alpha}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t - \boldsymbol{m}_t\|^2 \\ = & -\left(1 - \frac{2L^2\gamma^2}{\alpha^2} - 4L\gamma - \frac{2\gamma L}{\alpha}\right)\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{m}_t\|^2 + \left(\frac{2L^2\gamma^2}{\alpha^2} + \frac{3\gamma L}{\alpha} + 4L\gamma\right)\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t - \boldsymbol{m}_t\|^2,\end{aligned} \tag{15}$$

For $\mathbb{E}\left\|\boldsymbol{u}_t - \boldsymbol{m}_t\right\|^2$, we have

$$\begin{aligned}\mathbb{E}\|\boldsymbol{u}_t - \boldsymbol{m}_t\|^2 = & \mathbb{E}\|(1-\alpha)(\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}) + \alpha(\boldsymbol{g}_t - \nabla f(\boldsymbol{x}_t))\|^2 \\ = & \mathbb{E}_{1:t-1}\mathbb{E}_t\|(1-\alpha)(\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}) + \alpha(\boldsymbol{g}_t - \nabla f(\boldsymbol{x}_t))\|^2 \\ = & \mathbb{E}_{1:t-1}\mathbb{E}_t\|(1-\alpha)(\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1})\|^2 + \mathbb{E}_{1:t-1}\mathbb{E}_t\|\alpha(\boldsymbol{g}_t - \nabla f(\boldsymbol{x}_t))\|^2 \\ \leq & (1-\alpha)^2\mathbb{E}\|\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\|^2 + \alpha^2\sigma^2 \\ \leq & (1-\alpha)\mathbb{E}\|\boldsymbol{u}_{t-1} - \boldsymbol{m}_{t-1}\|^2 + \alpha^2\sigma^2 \\ = & (1-\alpha)^t\mathbb{E}\|\boldsymbol{u}_0 - \boldsymbol{m}_0\|^2 + \alpha^2\sum_{s=1}^{t}(1-\alpha)^{t-s}\sigma^2.\end{aligned}$$

Therefore we have

$$\mathbb{E}\|\boldsymbol{u}_t - \boldsymbol{m}_t\|^2 \leq \alpha\sigma^2.$$

So (15) becomes

$$\begin{aligned}&\sum_{t=0}^{T-1}\left(\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{u}_t\|^2 - (1-2L\gamma)\mathbb{E}\|\boldsymbol{u}_t\|^2\right) \\ \leq & -\left(1 - \frac{2L^2\gamma^2}{\alpha^2} - \frac{4\gamma L}{\alpha} - 2L\gamma\right)\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{m}_t\|^2 + \left(\frac{2L^2\gamma^2}{\alpha^2} + \frac{3\gamma L}{2\alpha} + 4L\gamma\right)\sum_{t=0}^{T-1}\alpha\sigma^2\end{aligned}$$

Therefore, for **Momentum SGD**

$$\begin{aligned}\sum_{t=0}^{T-1}A_t = & \sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{u}_t\|^2 - (1-2L\gamma)\mathbb{E}\|\boldsymbol{u}_t\|^2 - \frac{1}{4}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \\ \leq & -\left(1 - \frac{2L^2\gamma^2}{\alpha^2} - \frac{4\gamma L}{\alpha} - 2L\gamma\right)\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{m}_t\|^2 + \left(\frac{2L\gamma}{\alpha} + \frac{3}{2} + 4\alpha\right)\gamma LT\sigma^2.\end{aligned}$$

So if $\gamma L \leq \frac{\alpha}{12}$ and $\gamma L \leq \frac{1}{4}$, we have

$$\sum_{t=0}^{T-1} A_t \leq \frac{17\gamma L\sigma^2 T}{3}$$

□

Proof of Theorem 5. For STROM, the most important part is to upper bound $\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_t - \nabla f(\boldsymbol{x}_t)\|^2$. Therefore we first focus on this term.

Denoting $\boldsymbol{e}_t := \boldsymbol{u}_t - \nabla f(\boldsymbol{x}_t)$ and $\overline{\boldsymbol{g}}_t(\boldsymbol{x}) := \frac{1}{n}\sum_i \nabla F(\boldsymbol{x}; \xi_t^{(i)})$, we get

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{e}_t\|^2\\
=&\mathbb{E}\|(1-\alpha)\boldsymbol{e}_{t-1} + (\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)) - (1-\alpha)(\overline{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \nabla f(\boldsymbol{x}_{t-1}))\|^2\\
=&\mathbb{E}\|(1-\alpha)\boldsymbol{e}_{t-1}\|^2 + \mathbb{E}\|(\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)) - (1-\alpha)(\overline{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) - \nabla f(\boldsymbol{x}_{t-1}))\|^2\\
=&(1-\alpha)^2\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + \mathbb{E}\left\|\alpha(\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)) + (1-\alpha)\left(\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \overline{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) + \nabla f(\boldsymbol{x}_{t-1}) - \nabla f(\boldsymbol{x}_t)\right)\right\|^2\\
\leq&(1-\alpha)^2\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + 2\alpha^2\mathbb{E}\left\|\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\right\|^2 + 2(1-\alpha)^2\mathbb{E}\left\|\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \overline{\boldsymbol{g}}_t(\boldsymbol{x}_{t-1}) + \nabla f(\boldsymbol{x}_{t-1}) - \nabla f(\boldsymbol{x}_t)\right\|^2\\
=&(1-\alpha)^2\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + 2\alpha^2\mathbb{E}\left\|\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\right\|^2\\
&+\frac{2(1-\alpha)^2}{n^2}\mathbb{E}\left\|\sum_{i=1}^n\left(\nabla F(\boldsymbol{x}_t;\xi_t^{(i)}) - \nabla f_i(\boldsymbol{x}_t) - \nabla F(\boldsymbol{x}_{t-1};\xi_t^{(i)}) + \nabla f_i(\boldsymbol{x}_{t-1})\right)\right\|^2\\
=&(1-\alpha)^2\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + 2\alpha^2\mathbb{E}\left\|\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\right\|^2\\
&+\frac{2(1-\alpha)^2}{n^2}\sum_{i=1}^n\mathbb{E}\left\|\nabla F(\boldsymbol{x}_t;\xi_t^{(i)}) - \nabla f_i(\boldsymbol{x}_t) - \nabla F(\boldsymbol{x}_{t-1};\xi_t^{(i)}) + \nabla f_i(\boldsymbol{x}_{t-1})\right\|^2\\
\leq&(1-\alpha)^2\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + 2\alpha^2\mathbb{E}\left\|\overline{\boldsymbol{g}}_t(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\right\|^2\\
&+\frac{4(1-\alpha)^2}{n^2}\sum_{i=1}^n\mathbb{E}\left\|\nabla F(\boldsymbol{x}_t;\xi_t^{(i)}) - \nabla F(\boldsymbol{x}_{t-1};\xi_t^{(i)})\right\|^2 + \frac{4(1-\alpha)^2}{n^2}\sum_{i=1}^n\mathbb{E}\left\|\nabla f_i(\boldsymbol{x}_t) - \nabla f_i(\boldsymbol{x}_{t-1})\right\|^2\\
\leq&(1-\alpha)^2\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + \frac{2\alpha^2\sigma^2}{n} + \frac{4(1-\alpha)^2(L^2+L_F^2)}{n}\mathbb{E}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2\\
\leq&(1-\alpha)^2\mathbb{E}\|\boldsymbol{e}_{t-1}\|^2 + \frac{2\alpha^2\sigma^2}{n} + \frac{4(1-\alpha)^2(L^2+L_F^2)\gamma^2}{n}\mathbb{E}\|\boldsymbol{u}_{t-1}\|^2. \qquad (16)
\end{aligned}$$

Using Lemma 9, we get

$$\begin{aligned}
\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{e}_t\|^2 \leq&\frac{\sum_{t=0}^{T-1}\left(\frac{2\alpha^2\sigma^2}{n} + \frac{4(1-\alpha)^2(L^2+L_F^2)\gamma^2}{n}\mathbb{E}\|\boldsymbol{u}_{t-1}\|^2\right) + \mathbb{E}\|\boldsymbol{u}_0 - \nabla f(\boldsymbol{x}_0)\|^2}{1-(1-\alpha)^2}\\
\leq&\frac{1}{\alpha}\left(\frac{2\alpha^2\sigma^2 T}{n} + \frac{4(1-\alpha)^2(L^2+L_F^2)\gamma^2}{n}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_{t-1}\|^2\right) + \frac{\mathbb{E}\|\boldsymbol{u}_0 - \nabla f(\boldsymbol{x}_0)\|^2}{\alpha}\\
=&\frac{2\alpha\sigma^2 T}{n} + \frac{4(1-\alpha)^2(L^2+L_F^2)\gamma^2}{\alpha n}\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{u}_{t-1}\|^2 + \frac{\mathbb{E}\|\boldsymbol{u}_0 - \nabla f(\boldsymbol{x}_0)\|^2}{\alpha}. \qquad (17)
\end{aligned}$$

Since $A_t = \mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t) - \hat{\boldsymbol{u}}_t\|^2 - (1-2L\gamma)\mathbb{E}\|\hat{\boldsymbol{u}}_t\|^2 - \frac{\mathbb{E}\|\nabla f(\hat{\boldsymbol{x}}_t)\|^2}{4}$, therefore if $\frac{4(1-\alpha)^2(L^2+L_F^2)\gamma^2}{\alpha n} \leq \frac{1}{2}$ and $1 - 2L\gamma \geq \frac{1}{2}$, then we have

$$\begin{aligned}
\sum_{t=0}^{T-1} A_t \leq&\frac{2\alpha\sigma^2 T}{n} + \frac{\mathbb{E}\|\boldsymbol{u}_0 - \nabla f(\boldsymbol{x}_0)\|^2}{\alpha}\\
\leq&\frac{2\alpha\sigma^2 T}{n} + \frac{\sigma^2}{n\alpha B_0}.
\end{aligned}$$

□

Proof of Theorem 7. Here we introduce an auxiliary variable that is defined as

$$Z(\boldsymbol{x},\boldsymbol{y}) := \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}) - \langle \nabla^2 f(\boldsymbol{y}), \boldsymbol{x}-\boldsymbol{y}\rangle.$$

Then from the bounded Hessian in Assumption 3, we have

$$\|Z(\boldsymbol{x},\boldsymbol{y})\| \leq \rho\|\boldsymbol{x}-\boldsymbol{y}\|^2.$$

Moreover, since $\boldsymbol{u}_t$ is a weighted average over all history stochastic gradients, under the bounded stochastic gradient of Assumption 3, we have

$$\mathbb{E}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2 \leq \gamma^2\Delta^2.$$

Denote $\boldsymbol{e}_t := \boldsymbol{v}_t - \nabla f(\boldsymbol{x}_t)$, we get

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{e}_t\|^2\\
=&\mathbb{E}\left\|(1-\alpha)\boldsymbol{e}_{t-1} + \alpha\nabla F\left(\boldsymbol{x}_t + \frac{1-\alpha}{\alpha}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}); \xi_t^{(i)}\right) - \nabla f(\boldsymbol{x}_t) + (1-\alpha)\nabla f(\boldsymbol{x}_{t-1})\right\|^2\\
=&\mathbb{E}\left\|(1-\alpha)\boldsymbol{e}_{t-1} + \alpha\nabla f\left(\boldsymbol{x}_t + \frac{1-\alpha}{\alpha}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\right) - \nabla f(\boldsymbol{x}_t) + (1-\alpha)\nabla f(\boldsymbol{x}_{t-1})\right\|^2\\
&+\mathbb{E}\left\|\alpha\nabla F\left(\boldsymbol{x}_t + \frac{1-\alpha}{\alpha}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}); \xi_t^{(i)}\right) - \alpha\nabla f\left(\boldsymbol{x}_t + \frac{1-\alpha}{\alpha}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\right)\right\|^2\\
\leq&\mathbb{E}\left\|(1-\alpha)\boldsymbol{e}_{t-1} + \alpha\nabla f\left(\boldsymbol{x}_t + \frac{1-\alpha}{\alpha}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\right) - \nabla f(\boldsymbol{x}_t) + (1-\alpha)\nabla f(\boldsymbol{x}_{t-1})\right\|^2 + \frac{\alpha^2\sigma^2}{n}\\
=&\mathbb{E}\left\|(1-\alpha)\boldsymbol{e}_{t-1} + (1-\alpha)Z(\boldsymbol{x}_{t-1},\boldsymbol{x}_t) + \alpha Z\left(\boldsymbol{x}_t + \frac{1-\alpha}{\alpha}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}), \boldsymbol{x}_t\right)\right\|^2 + \frac{\alpha^2\sigma^2}{n}\\
\leq&(1+\alpha)\,\mathbb{E}\,\|(1-\alpha)\boldsymbol{e}_{t-1}\|^2 + \left(1+\frac{1}{\alpha}\right)\mathbb{E}\left\|(1-\alpha)Z(\boldsymbol{x}_{t-1},\boldsymbol{x}_t) + \alpha Z\left(\boldsymbol{x}_t + \frac{1-\alpha}{\alpha}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}), \boldsymbol{x}_t\right)\right\|^2\\
&+\frac{\alpha^2\sigma^2}{n} \qquad \text{(applying } (x+y)^2 \leq (1+\alpha)x^2 + (1+\frac{1}{\alpha})y^2\text{)}\\
\leq&(1-\alpha)\,\mathbb{E}\,\|\boldsymbol{e}_{t-1}\|^2 + \frac{(\alpha+1)}{\alpha^2}\mathbb{E}\,\|(1-\alpha)Z(\boldsymbol{x}_{t-1},\boldsymbol{x}_t)\|^2\\
&+\frac{(\alpha+1)}{\alpha(1-\alpha)}\mathbb{E}\left\|\alpha Z\left(\boldsymbol{x}_t + \frac{1-\alpha}{\alpha}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}), \boldsymbol{x}_t\right)\right\|^2 + \frac{\alpha^2\sigma^2}{n}\\
\leq&(1-\alpha)\,\mathbb{E}\,\|\boldsymbol{e}_{t-1}\|^2 + \frac{(1-\alpha^2)(1-\alpha)\rho^2}{\alpha^3}\mathbb{E}\,\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|^4 + \frac{\alpha^2\sigma^2}{n}\\
\leq&(1-\alpha)\,\mathbb{E}\,\|\boldsymbol{e}_{t-1}\|^2 + \frac{(1-\alpha)\rho^2}{\alpha^3}\mathbb{E}\,\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|^4 + \frac{\alpha^2\sigma^2}{n}.
\end{aligned} \tag{18}$$

In this case, we also have $\mathbb{E}_{\xi_{t-1}}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2 \leq \gamma^2\Delta^2$, and (18) becomes

$$\mathbb{E}\|\boldsymbol{e}_t\|^2 \leq (1-\alpha)\,\mathbb{E}\,\|\boldsymbol{e}_{t-1}\|^2 + \frac{(1-\alpha)\rho^2\gamma^4\Delta^4}{\alpha^3} + \frac{\alpha^2\sigma^2}{n}. \tag{19}$$

Using Lemma 9 and (19), we get

$$\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{v}_t\|^2 \leq \frac{\sum_{t=1}^{T}\left(\frac{\alpha^2\sigma^2}{n} + \frac{\rho^2\gamma^4\Delta^4}{\alpha^3}\right) + \mathbb{E}\|\boldsymbol{e}_0\|^2}{1-(1-\alpha)} \leq \frac{\alpha\sigma^2 T}{n} + \frac{\sigma^2}{\alpha n B_0} + \frac{\rho^2\gamma^4\Delta^4 T}{\alpha^4}.$$

The lemma is proved. □

10 Proof of Corollary

10.1 Proof of Corollary 4

Proof. Combining Theorem 1 and 3 together, we shall get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \leq \frac{16(f(\boldsymbol{x}_0) - f^*)}{\gamma T} + \frac{136\gamma L\sigma^2}{3n} + (64L_{\mathscr{A}}^2 + 2L^2)\gamma^2\alpha^2 T\epsilon^2.$$

Since $L_{\mathscr{A}} = L$, after setting $\gamma = \min\left\{\frac{\alpha}{12L}, \frac{\sqrt{n}}{\sqrt{T}\sigma}, \left(\frac{1}{\epsilon^2 T}\right)^{\frac{1}{3}}\right\}$, it can be easily verified that we have

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \leq \mathcal{O}\left(\frac{\sigma}{\sqrt{nT}} + \frac{\alpha^2}{(\epsilon T)^{\frac{2}{3}}} + \frac{1}{\alpha T}\right),$$

where we treat $f(\boldsymbol{x}_0) - f^*$ and L as constants. □

10.2 Proof of Corollary 6

Proof. Combining Theorem 1 and 5 together, and setting $\alpha = \frac{8L^2\gamma^2}{n}$ we shall get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \leq \frac{16(f(\boldsymbol{x}_0) - f^*)}{\gamma T} + \frac{128L^2\gamma^2\sigma^2}{n^2} + \frac{\sigma^2}{L^2\gamma^2 B_0 T} + \frac{64(64L_{\mathscr{A}}^2 + 2L^2)L^4\gamma^6\epsilon^2}{n}.$$

Since $L_{\mathscr{A}} = 2L$, after setting $\gamma = \min\left\{\frac{1}{4L}, \left(\frac{n^2}{\sigma^2 T}\right)^{\frac{1}{3}}, \left(\frac{n}{\epsilon^2 T}\right)^{\frac{1}{7}}\right\}$ and $B_0 = \frac{\sigma^{\frac{8}{3}}T^{\frac{1}{3}}}{n^{\frac{2}{3}}}$, then it can be easily verified that we have

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \leq \mathcal{O}\left(\left(\frac{\sigma}{nT}\right)^{\frac{2}{3}} + \left(\frac{\epsilon^2}{nT^6}\right)^{\frac{1}{7}} + \frac{1}{T}\right),$$

where we treat $f(\boldsymbol{x}_0) - f^*$, L as constants. □

10.3 Proof of Corollary 8

Proof. Combining Theorem 1 and 7 together, we shall get

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \leq \frac{16(f(\boldsymbol{x}_0) - f^*)}{\gamma T} + \frac{8\alpha\sigma^2}{n} + \frac{8\sigma^2}{\alpha n B_0 T} + \frac{8\rho^2\gamma^4\Delta^4}{\alpha^4} + \frac{8(64L_{\mathscr{A}}^2 + 2L^2)L^2\gamma^6\epsilon^2}{n}.$$

Since $L_{\mathscr{A}} = 2L$, after setting $\gamma = \min\left\{\frac{1}{2L}, \left(\frac{n^4}{\sigma^8 T^5}\right)^{\frac{1}{9}}, \left(\frac{n}{\epsilon^2 T}\right)^{\frac{1}{7}}\right\}$, $\alpha = \left(\frac{n^5}{\sigma^8 T^4}\right)^{\frac{1}{9}}$ and $B_0 = 1$, then it can be easily verified that we have

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}_t)\|^2 \leq \mathcal{O}\left(\left(\frac{\sigma^8}{n^4 T^4}\right)^{\frac{1}{9}} + \left(\frac{\epsilon^2}{nT^6}\right)^{\frac{1}{7}} + \frac{1}{T}\right),$$

where we treat $f(\boldsymbol{x}_0) - f^*$, L and ρ as constants. □