Journal of Machine Learning Research 24 (2023) 1-62 Submitted 2/22; Published 5/23

Buffered Asynchronous SGD for Byzantine Learning

Yi-Rui Yang YANGYR@SMAIL.NJU.EDU.CN
Wu-Jun Li[*] LIWUJUN@NJU.EDU.CN
National Key Laboratory for Novel Software Technology
Department of Computer Science and Technology
Nanjing University, Nanjing 210023, China

Editor: Sathiya Keerthi

Abstract

Distributed learning has become a hot research topic due to its wide application in cluster-based large-scale learning, federated learning, edge computing, and so on. Most traditional distributed learning methods typically assume no failure or attack. However, many unexpected cases, such as communication failure and even malicious attack, may happen in real applications. Hence, Byzantine learning (BL), which refers to distributed learning with failure or attack, has recently attracted much attention. Most existing BL methods are synchronous, which are impractical in some applications due to heterogeneous or offline workers. In these cases, asynchronous BL (ABL) is usually preferred. In this paper, we propose a novel method, called buffered asynchronous stochastic gradient descent (BASGD), for ABL. To the best of our knowledge, BASGD is the first ABL method that can resist non-omniscient attacks without storing any instances on the server. Furthermore, we also propose an improved variant of BASGD, called BASGD with momentum (BASGDm), by introducing local momentum into BASGD. Compared with those methods which need to store instances on server, BASGD and BASGDm have a wider scope of application. Both BASGD and BASGDm are compatible with various aggregation rules. Moreover, both BASGD and BASGDm are proved to be convergent and able to resist failure or attack. Empirical results show that our methods significantly outperform existing ABL baselines when there exists failure or attack on workers.

Keywords: distributed machine learning, momentum, asynchronous Byzantine learning, buffer, stochastic gradient descent

1. Introduction

Due to the wide application in cluster-based large-scale learning, federated learning (Konevcnỳ et al., 2016; Kairouz et al., 2021), edge computing (Shi et al., 2016), and so on, distributed learning has recently become a hot research topic (Zinkevich et al., 2010; Yang, 2013; Jaggi et al., 2014; Shamir et al., 2014; Zhang and Kwok, 2014; Ma et al., 2015; Lee et al., 2017; Lian et al., 2017; Zhao et al., 2017; Sun et al., 2018; Wangni et al., 2018; Zhao et al., 2018; Zhou et al., 2018; Yu et al., 2019a,b; Haddadpour et al., 2019; Assran et al., 2020; Nokleby et al., 2020). Most traditional distributed learning methods are based on stochastic gradient descent (SGD) and its variants (Bottou, 2010; Xiao, 2010; Duchi et al., 2011; Johnson and Zhang, 2013; Shalev-Shwartz and Zhang, 2013; Zhang et al., 2013; Lin et al., 2014; Schmidt

[*] Corresponding author.

et al., 2017; Zheng et al., 2017; Zhao et al., 2018; Duan et al., 2020; Zhao et al., 2021), and typically assume no failure or attack.

However, in distributed learning applications with multiple networked machines (nodes), different kinds of hardware or software failure may happen (Lamport et al., 2019; Wang et al., 2020; Kairouz et al., 2021). Representative failure includes bit-flipping in the communication media and the memory of some workers (Xie et al., 2019). In this case, small failure on some machines (workers) might cause a distributed learning method to fail. In addition, malicious attack should not be neglected in an open network where the manager (or server) generally has not much control over the workers, such as in the cases of edge computing and federated learning. Malicious workers may behave arbitrarily or even adversarially. Hence, *Byzantine learning* (BL), which refers to distributed learning with failure or attack, has attracted much attention (Diakonikolas et al., 2017; Chen et al., 2017; Damaskinos et al., 2018; Baruch et al., 2019; Diakonikolas and Kane, 2019; Wu et al., 2020; Karimireddy et al., 2022).

BL methods can be divided into two main categories: synchronous BL (SBL) methods and asynchronous BL (ABL) methods. In SBL methods, the learning information, such as the gradient in SGD, of all workers will be aggregated in a synchronous way. On the contrary, in ABL methods the learning information of workers will be aggregated in an asynchronous way.

Existing SBL methods mainly take three different ways to achieve resilience against *Byzantine workers* which refer to those workers with failure or attack. The first way is to filter the suspicious learning information (gradients) before averaging. Representative examples include ByzantineSGD (Alistarh et al., 2018) and Zeno (Xie et al., 2019). The second way is to replace the simple averaging aggregation operation with some more robust aggregation operations, such as median & trimmed-mean (Yin et al., 2018), geometric median (Chen et al., 2017), and centered-clipping (Karimireddy et al., 2021). Krum (Blanchard et al., 2017), RSA (Li et al., 2019), ByzantinePGD (Yin et al., 2019) and SignSGD (Seide et al., 2014; Bernstein et al., 2019; Sohn et al., 2020) also take this way. The third way is based on redundancy. In this kind of methods such as DRACO (Chen et al., 2018), DETOX (Rajput et al., 2019), ByzShield (Konstantinidis and Ramamoorthy, 2021), Byzantine resilience is achieved by assigning computation task of each gradient to several nodes. In these methods, the manager (or server) may have access to the exact true gradient despite the existence of Byzantine workers. However, methods based on redundancy usually have higher computation cost and storage cost than methods that take the other two ways.

Some recent works on SBL also reveal that using history information can strengthen the Byzantine resilience (Allen-Zhu et al., 2020; El-Mhamdi et al., 2021b; Karimireddy et al., 2021). The advantage of SBL methods is that they are relatively simple and easy to be implemented, but SBL methods will result in slow convergence when there exist heterogeneous workers. Furthermore, in some applications like federated learning and edge computing, synchronization cannot even be performed most of the time due to the offline workers (clients or edge servers). Hence, ABL methods are preferred in these cases.

To the best of our knowledge, there exist only a few ABL methods. Kardam (Damaskinos et al., 2018) introduces two filters to drop out suspicious learning information (gradients), which can still achieve good performance when the communication delay is heavy. However, when in the face of malicious attack, some work (Xie et al., 2020b) finds that Kardam also drops out most correct gradients in order to filter all faulty (failure) gradients. Hence, Kardam cannot resist malicious attack. Zeno++ (Xie et al., 2020b) and Sageflow (Park et al., 2021) need to store some training instances on the server. In some practical applications like federated learning (Kairouz et al., 2021), storing data on server will increase the risk of privacy leakage or even face legal risk. There are also existing works (El-Mhamdi et al., 2021a) that study ABL under the decentralized framework. As far as we know, under the Parameter Server framework where the server has no access to any training instances, there does not exist any ABL method that can resist malicious attack.

Moreover, in some recently proposed attacks (Xie et al., 2020a; Baruch et al., 2019), attackers are assumed to have access to all the information on other workers and use the information for attack. This type of attacks are called omniscient attacks, while the others are called non-omniscient attacks. As far as we know, there does not exist any ABL method that can resist the two omniscient attacks 'Fall of Empires' (Xie et al., 2020a) and 'A Little is Enough' (Baruch et al., 2019).

In this paper, we propose a novel method called buffered asynchronous stochastic gradient descent (BASGD) and an improved variant of BASGD called BASGD with momentum (BASGDm) for ABL. The main contributions are listed as follows:

- To the best of our knowledge, BASGD is the first ABL method that can resist non-omniscient attacks without storing any instances on the server. Compared with those methods which need to store instances on the server, BASGD has a wider scope of application.
- An improved variant of BASGD, called BASGD with momentum (BASGDm), is further proposed by introducing local momentum into BASGD. As far as we know, BASGDm is the first ABL method that can resist the two omniscient attacks 'Fall of Empires' and 'A Little is Enough'.
- Both BASGD and BASGDm are compatible with various aggregation rules. Moreover, both BASGD and BASGDm are proved to be convergent and able to resist failure or attack.
- Empirical results show that our methods significantly outperform existing ABL baselines when there exists failure or attack on workers.

2. Preliminary

In this section, we present the preliminary of this paper, including the distributed learning framework used in this paper and the definition of Byzantine worker.

2.1 Distributed Learning Framework

Many machine learning models, such as logistic regression and deep neural networks, can be formulated as the following finite sum optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{w}; z_i), \tag{1}$$

where $\mathbf{w}$ is the parameter to learn, d is the dimension of parameter, n is the number of training instances, $f(\mathbf{w}; z_i)$ is the empirical loss on the instance z_i. The goal of distributed learning is to solve the problem in (1) by designing learning algorithms based on multiple networked machines.

Although there have appeared many distributed learning frameworks, in this paper we focus on the widely used Parameter Server (PS) framework (Li et al., 2014). In a PS framework, there are several workers and one or more servers. Each worker can only communicate with server(s). There may exist more than one server in a PS framework, but for the problem of this paper, servers can be logically conceived as a unity. Without loss of generality, we will assume there is only one server in this paper. Training instances are disjointedly distributed across m workers. Let $\mathcal{D}_k$ denote the index set of training instances on worker_k, we have $\cup_{k=1}^{m}\mathcal{D}_k = \{1, 2, \ldots, n\}$ and $\mathcal{D}_k \cap \mathcal{D}_{k'} = \emptyset$ if $k \neq k'$. In this paper, we assume that the server has no access to any training instances. If two instances have the same value, they are still deemed as two distinct instances. Namely, z_i may equal $z_{i'}$ $(i \neq i')$. One popular asynchronous method to solve the problem in (1) under the PS framework is ASGD (Dean et al., 2012) (see Appendix A for details). In this paper, we assume each worker samples one instance for gradient computation each time. The analysis of the mini-batch case is similar.

In PS-based ASGD, the server is responsible for updating and maintaining the latest parameter. The number of iterations that the server has already executed is used as the global logical clock of the server. In the beginning, iteration number $t = 0$. Each time an SGD step is executed, t will increase by 1 immediately. The parameter after t iterations is denoted as $\mathbf{w}^t$. If the server sends parameters to worker_k at iteration t', some SGD steps may have been executed before the server receives gradient from worker_k next time at iteration t. Thus, we define the *delay* of worker_k at iteration t as $\tau_k^t = t - t'$. Worker_k is *heavily delayed* at iteration t if $\tau_k^t > \tau_{max}$, where τ_{max} is a pre-defined non-negative constant. In the analysis of this work, we will mainly consider the partially asynchronous setting (Bertsekas et al., 1989), where a limited number of heavily delayed workers at each iteration are assumed. When there is no confusion, we will omit the word 'partially' in the following text.

2.2 Byzantine Worker

For workers that have sent gradients (one or more) to the server at iteration t, we call worker_k *loyal worker* if it has finished all the tasks without any fault and each sent gradient is correctly received by the server. Otherwise, worker_k is called *Byzantine worker*. If worker_k is a Byzantine worker, it means the received gradient from worker_k is not credible, which can be an arbitrary value. Formally, we denote the gradient computed by worker_k at iteration t as $\mathbf{g}_k^t$. Then, we have:

$$\mathbf{g}_k^t = \begin{cases} \nabla f(\mathbf{w}^{t'}; z_i), & \text{if worker\_}k \text{ is loyal at iteration } t; \\ *, & \text{if worker\_}k \text{ is Byzantine at iteration } t, \end{cases}$$

where $0 \leq t' \leq t$, and i is randomly sampled from $\mathcal{D}_k$. '$*$' represents an arbitrary value. Our definition of Byzantine worker is consistent with most previous works (Blanchard et al.,

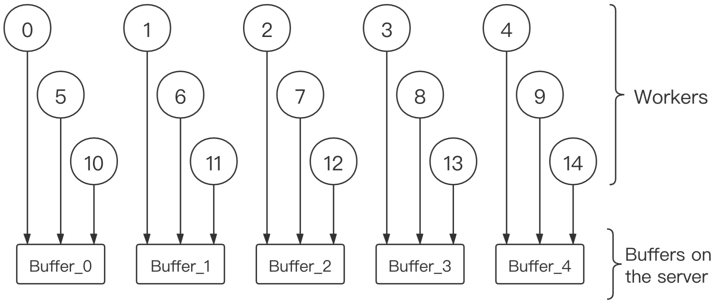


Figure 1: An example of buffers. Each circle represents a worker, and the number is the worker ID. There are 15 workers and 5 buffers. The gradient received from worker_s is stored in buffer_$\{s \bmod 5\}$.

2017; Xie et al., 2019, 2020b). Either accidental failure or malicious attack will result in Byzantine workers.

We would also like to clarify that, in vanilla ASGD, there is at most one gradient sent from any worker_k at each iteration. However, in the new method called BASGD that we will present in Section 3, there are possibly multiple gradients sent from any worker_k at each iteration.

3. Buffered Asynchronous SGD

In synchronous BL, gradients from all workers are received at each iteration. We can compare the gradients with each other, and then filter suspicious ones, or use more robust aggregation rules such as median and trimmed-mean for updating. However, in asynchronous BL, only one gradient is received at a time. Without any training instances stored on the server, it is difficult for the server to identify whether a received gradient is credible or not.

In order to deal with this problem in asynchronous BL, we propose a novel method called buffered asynchronous SGD (BASGD). BASGD introduces B buffers ($0 < B \leq m$) on the server, and the gradient used for updating parameters will be aggregated from these buffers. The detail of the learning procedure of BASGD is presented in Algorithm 1.

In this section, we will first introduce the three key components of BASGD: buffer, aggregation function, and mapping table. At the end of this section, we will also introduce an improved variant of BASGD which is called buffered asynchronous SGD with momentum (BASGDm).

3.1 Buffer

In BASGD, the m workers do the same job as that in ASGD, while the updating rule on the server is modified. More specifically, there are B buffers ($0 < B \leq m$) on server. When a gradient $\mathbf{g}$ from worker_s is received, it will be temporarily stored in buffer b, where $b = s \ mod \ B$, as illustrated in Figure 1. Only when each buffer has stored at least one gradient, a new SGD step will be executed. Please note that no matter whether an SGD step

Algorithm 1 Buffered Asynchronous SGD (BASGD)

```
Server:
Input: learning rate $\eta$, reassignment interval $\Delta$,
  buffer number $B$, aggregation function: $Aggr(\cdot)$;
Initialization: model parameter $\mathbf{w}^0$;
Set $\mathbf{h}_b \leftarrow \mathbf{0}$ and $N_b^0 \leftarrow 0$ for all $b = 0, \dots, B-1$;
Initialize mapping table $\beta_s \leftarrow s$ $(s = 0, 1, \dots, m-1)$;
Send initial $\mathbf{w}^0$ to all workers;
Set $t \leftarrow 0$, and start the timer;
repeat
  Wait until receiving $\mathbf{g}$ from some worker_$s$;
  Choose buffer: $b \leftarrow \beta_s \ mod \ B$;
  Let $N_b^t \leftarrow N_b^t + 1$, and $\mathbf{h}_b \leftarrow \frac{(N_b^t - 1)\mathbf{h}_b + \mathbf{g}}{N_b^t}$;
  if $N_b^t > 0$ for each $b \in \{0, \dots, B-1\}$ then
    Aggregate: $\mathbf{G}^t = Aggr([\mathbf{h}_0, \dots, \mathbf{h}_{B-1}])$;
    Execute SGD step: $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta \cdot \mathbf{G}^t$;
    Zero out buffers: $\mathbf{h}_b \leftarrow \mathbf{0}$, $N_b^t \leftarrow 0$ $(b = 0, \dots, B-1)$;
    Set $t \leftarrow t + 1$, and restart the timer;
  end if
  if the timer has exceeded $\Delta$ seconds then
    Zero out buffers: $\mathbf{h}_b \leftarrow \mathbf{0}$, $N_b^t \leftarrow 0$ $(b = 0, \dots, B-1)$;
    Modify the mapping table $\{\beta_s\}_{s=0}^{m-1}$ for buffer reassignment, and restart the timer;
  end if
  Send the latest parameters back to worker_$s$, no matter whether an SGD step is executed
  or not.
until stop criterion is satisfied
Notify all workers to stop;


Worker_$k$:   $(k = 0, 1, ..., m-1)$
repeat
  Wait until receiving the latest parameter $\mathbf{w}$ from server;
  Randomly sample an index $i$ from $\mathcal{D}_k$;
  Compute $\nabla f(\mathbf{w}; z_i)$;
  Send $\nabla f(\mathbf{w}; z_i)$ to server;
until receive server's notification to stop
```

is executed or not, the server will immediately send the latest parameters back to the worker after receiving a gradient. Hence, BASGD introduces no barrier and is an asynchronous algorithm.

For each buffer b, more than one gradient may have been received at iteration t. We will store the average of these gradients (denoted by $\mathbf{h}_b$) in buffer b. Assume that there are already $(N-1)$ gradients $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{N-1}$ which should be stored in buffer b, and

$\mathbf{h}_{b(old)} = \frac{1}{N-1}\sum_{i=1}^{N-1}\mathbf{g}_i$. When the N-th gradient $\mathbf{g}_N$ is received, the new average value is:

$$\mathbf{h}_{b(new)} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{g}_i = \frac{N-1}{N}\cdot\mathbf{h}_{b(old)} + \frac{1}{N}\cdot\mathbf{g}_N.$$

This is the updating rule for each buffer b when a gradient is received. We use N_b^t to denote the total number of gradients stored in buffer b at the t-th iteration. After the parameter $\mathbf{w}$ is updated, all buffers will be zeroed out at once. With the benefit of buffers, the server has access to B candidate gradients when updating model parameters. Thus, a more reliable (robust) gradient can be aggregated from the B gradients of buffers, if a proper aggregation function $Aggr(\cdot)$ is chosen.

Please note that from the perspective of workers, BASGD is fully asynchronous, since a worker will immediately receive the latest parameter from the server after sending a gradient to the server, without waiting for other workers. Meanwhile, from the perspective of the server, BASGD is semi-asynchronous because the server will not update the model until all buffers are filled. Actually, it is a necessity to limit the updating frequency in ABL when the server has no instances. If the server always updates the model when receiving a gradient, it will be easily foiled when Byzantine workers send gradients much more frequently than others. A similar conclusion has been proved in previous works (Damaskinos et al., 2018).

3.2 Aggregation Function

When an SGD step is ready to be executed, there are B buffers providing candidate gradients. An aggregation function is needed to get the final gradient for updating. A naive way is to take the mean of all candidate gradients. However, the mean value is sensitive to outliers which are common in BL. For designing proper aggregation functions, we first define the q-Byzantine Robust (q-BR) condition to quantitatively describe the Byzantine resilience ability of an aggregation function.

Definition 1 (q-Byzantine Robust)**.** *For an aggregation function $Aggr(\cdot)$: $Aggr([\mathbf{h}_0, \dots, \mathbf{h}_{B-1}]) = \mathbf{G}$, where $\mathbf{G} = [G_1, \dots, G_d]^T$ and $\mathbf{h}_b = [h_{b1}, \dots, h_{bd}]^T, \forall b \in \{0, \dots, B-1\}$, we call $Aggr(\cdot)$ q-Byzantine Robust ($q \in \mathbb{Z}, 0 < q < B/2$), if it satisfies the following two properties:*
(a) $Aggr([\mathbf{h}_0 + \mathbf{h}', \dots, \mathbf{h}_{B-1} + \mathbf{h}']) = Aggr([\mathbf{h}_0, \dots, \mathbf{h}_{B-1}]) + \mathbf{h}'$, $\forall \mathbf{h}_0, \dots, \mathbf{h}_{B-1}, \mathbf{h}' \in \mathbb{R}^d$;
(b) $\min_{s\in\mathcal{S}}\{h_{sj}\} \leq G_j \leq \max_{s\in\mathcal{S}}\{h_{sj}\}$, $\forall j \in [d]$, $\forall \mathcal{S} \subset \{0, \dots, B-1\}$ with $|\mathcal{S}| = B - q$.

Intuitively, property (a) in Definition 1 says that if all candidate vectors $\mathbf{h}_i$ are added by a same vector $\mathbf{h}'$, the aggregated gradient will also be added by $\mathbf{h}'$. Property (b) says that for each coordinate j, the aggregated value G_j will be between the $(q+1)$-th smallest value and the $(q+1)$-th largest value among the j-th coordinates of all candidate vectors. Thus, the gradient aggregated by a q-BR function is insensitive to at least q outliers. We can find that the q-BR condition gets stronger when q increases. Namely, if $Aggr(\cdot)$ is q-BR, then for any $0 < q' < q$, $Aggr(\cdot)$ is also q'-BR.

Remark 2. *When $B > 1$, the mean function is not q-Byzantine Robust for any $q > 0$. We illustrate this with a one-dimension example. Let $h_0, \dots, h_{B-2} \in [0, 1]$ and $h_{B-1} = 10 \times B$. Then $\frac{1}{B}\sum_{b=0}^{B-1} h_b \geq \frac{h_{B-1}}{B} = 10$. Thus, the mean is larger than any of the first $B-1$ values.*

We show that the following two widely-used aggregation functions are both q-BR.

Definition 3 (Coordinate-wise median (Yin et al., 2018)). *For candidate vectors* $\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{B-1} \in \mathbb{R}^d$*,* $\mathbf{h}_b = [h_{b1}, h_{b2}, \ldots, h_{bd}]^T$*,* $\forall b = 0, \ldots, B-1$*.* Coordinate-wise median *is defined as:*

$$Med([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}]) = [Med(h_{\cdot 1}), \ldots, Med(h_{\cdot d})]^T,$$

where $Med(h_{\cdot j})$ *is the scalar median of the* j*-th coordinates,* $\forall j = 1, 2, \ldots, d$*.*

Definition 4 (Coordinate-wise q-trimmed-mean (Yin et al., 2018)). *For any positive interger* $q < B/2$ *and candidate vectors* $\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{B-1} \in \mathbb{R}^d$*,* $\mathbf{h}_b = [h_{b1}, h_{b2}, \ldots, h_{bd}]^T$*,* $\forall b = 0, \ldots, B-1$*.* Coordinate-wise q-trimmed-mean *is defined as:*

$$Trm([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}]) = [Trm(h_{\cdot 1}), \ldots, Trm(h_{\cdot d})]^T,$$

where $Trm(h_{\cdot j}) = \frac{1}{B-2q} \sum_{b \in \mathcal{M}_j} h_{bj}$ *is the scalar* q*-trimmed-mean.* $\mathcal{M}_j$ *is the subset of* $\{h_{bj}\}_{b=0}^{B-1}$ *obtained by removing the* q *largest elements and* q *smallest elements.*

In the following content, coordinate-wise median and coordinate-wise q-trimmed-mean are also called *median* and *trmean*, respectively.

Proposition 5. *Coordinate-wise* q*-trmean is* q*-BR. Coordinate-wise median is* $\lfloor \frac{B-1}{2} \rfloor$*-BR.*

Here, $\lfloor x \rfloor$ represents the maximum integer that is not larger than x. According to Proposition 5, both median and trmean are proper choices for aggregation function in BASGD. The proof can be found in Appendix B.

We also define another class of aggregation functions called $(\delta_{\max}, A_1, A_2)$-effective aggregation functions in Definition 6 and Definition 7. The definition of $(\delta_{\max}, A_1, A_2)$-effective aggregation function can be deemed as a bridge across synchronous Byzantine learning and asynchronous Byzantine learning.

Definition 6 (Stable aggregation function). *Aggregation function* $Aggr(\cdot)$ *is called* stable *provided that* $\forall \mathbf{h}_0, \ldots, \mathbf{h}_{B-1}$*,* $\tilde{\mathbf{h}}_0, \ldots, \tilde{\mathbf{h}}_{B-1} \in \mathbb{R}^d$*, we have:*

$$\left\| Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}]) - Aggr(\tilde{\mathbf{h}}_0, \ldots, \tilde{\mathbf{h}}_{B-1}) \right\| \leq \left(\sum_{b=0}^{B-1} \left\| \mathbf{h}_b - \tilde{\mathbf{h}}_b \right\|^2 \right)^{\frac{1}{2}}.$$

Definition 6 says that if $Aggr(\cdot)$ is a stable aggregation function, when there is a disturbance on buffers, the disturbance on the aggregated result by $Aggr(\cdot)$ will not be larger than the disturbance on buffers in L_2-norm.

Definition 7 (Effective aggregation function). *When the fraction of Byzantine workers is not larger than* $\delta_{\max}$*, stable aggregation function* $Aggr(\cdot)$ *is called a* $(\delta_{\max}, A_1, A_2)$*-effective aggregation function, provided that it satisfies the following two properties for all* $\mathbf{w}^t \in \mathbb{R}^d$ *in cases without delay (*$\tau_k^t = 0,\ \forall t = 0, 1, \ldots, T-1$*):*

(a) $\mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}_{syn}^t \mid \mathbf{w}^t] \geq \|\nabla F(\mathbf{w}^t)\|^2 - A_1$*;*
(b) $\mathbb{E}[\|\mathbf{G}_{syn}^t\|^2 \mid \mathbf{w}^t] \leq (A_2)^2$*;*

where $A_1, A_2 \in \mathbb{R}_+$ *are two non-negative constants,* $\mathbf{G}_{syn}^t$ *is the aggregated result of* $Aggr(\cdot)$ *at the* t*-th iteration in cases without delay.*

More specifically, $\mathbf{G}_{syn}^t$ can be the aggregated *gradient* or *momentum*. In the conference version of BASGD (Yang and Li, 2021), $\mathbf{G}_{syn}^t$ is the aggregated *gradient*. We change the statement to make it compatible with the BASGDm method (please refer to Section 3.4). The two properties in Definition 7 are for synchronous cases mainly because we would like to use this definition to extend existing theoretical results for synchronous Byzantine learning methods to those for asynchronous cases. Please see Section 4 for the detailed results.

For different aggregation functions, constants A_1 and A_2 may differ. A_1 and A_2 are related to loss function $F(\cdot)$, distribution of instances, buffer number B and maximum Byzantine worker fraction $\delta_{\max}$. Inequalities (a) and (b) in Definition 7 are two important properties in convergence proof of synchronous Byzantine learning methods. As revealed in (Yang et al., 2020), there are many existing aggregation rules for Byzantine learning. We find that most of them satisfy Definition 7. For example, Krum, median, and trimmed-mean have already been proved to satisfy these two properties (Blanchard et al., 2017; Yin et al., 2018). SignSGD (Bernstein et al., 2019) can be seen as a combination of 1-bit quantization and median aggregation, while median satisfies the properties in Definition 7.

The q-BR property in Definition 1 is relatively easy to check, while the definition of $(\delta_{\max}, A_1, A_2)$-effective aggregation allows us to extend existing theoretical results for synchronous cases to those for the asynchronous cases, which we mainly focus on in this work. Besides the two types of aggregation rules presented in Definition 1 and Definition 7, we also introduce the definition of $(\delta_{\max}, c)$-robust aggregation function in Definition 8.

Definition 8 ($(\delta_{\max}, c)$-robust aggregation function)**.** *Aggregation function $Aggr(\cdot)$ is called* $(\delta_{\max}, c)$-robust *provided that for any B independent random vectors $\mathbf{h}_0, \ldots, \mathbf{h}_{B-1} \in \mathbb{R}^d$ and any set $\mathcal{H} \subseteq \{0, 1, \ldots, B-1\}$ with $1 - \frac{|\mathcal{H}|}{B} = \delta \leq \delta_{\max}$ where $\delta_{\max} < \frac{1}{2}$, we have:*

$$\mathbb{E}\left\| Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}]) - \frac{1}{|\mathcal{H}|}\sum_{b \in \mathcal{H}} \mathbf{h}_b \right\|^2 \leq c\delta\rho^2,$$

where constant $\rho \geq 0$ satisfies that $\mathbb{E}\|\mathbf{h}_b - \mathbf{h}_{b'}\|^2 \leq \rho^2$ for any fixed $b, b' \in \mathcal{H}$.

Definition 8 has been used in previous works (Karimireddy et al., 2021) to theoretically prove that using momentum can enhance the resilience against Byzantine attacks for i.i.d. cases in synchronous BL. Moreover, it has been proved that the aggregation error $O(\delta\rho^2)$ is theoretically optimal (Karimireddy et al., 2021). We will also introduce momentum to BASGD in Section 3.4 and theoretically prove that using momentum can also enhance the resilience against Byzantine attacks for i.i.d. cases in asynchronous BL in Section 4.

Meanwhile, please note that too large B could lower the updating frequency and damage the performance, while too small B may harm the Byzantine resilience. Thus, a moderate B is usually preferred. From another perspective, the choice of B can be viewed as a trade-off between efficiency and Byzantine resilience. In practical applications, practitioners are suggested to first determine the maximum fraction of Byzantine workers $\delta_{\max}$ that the system can tolerate, and then set B to make the aggregation function resilient to up to a fraction of $\delta_{\max}$ Byzantine workers.

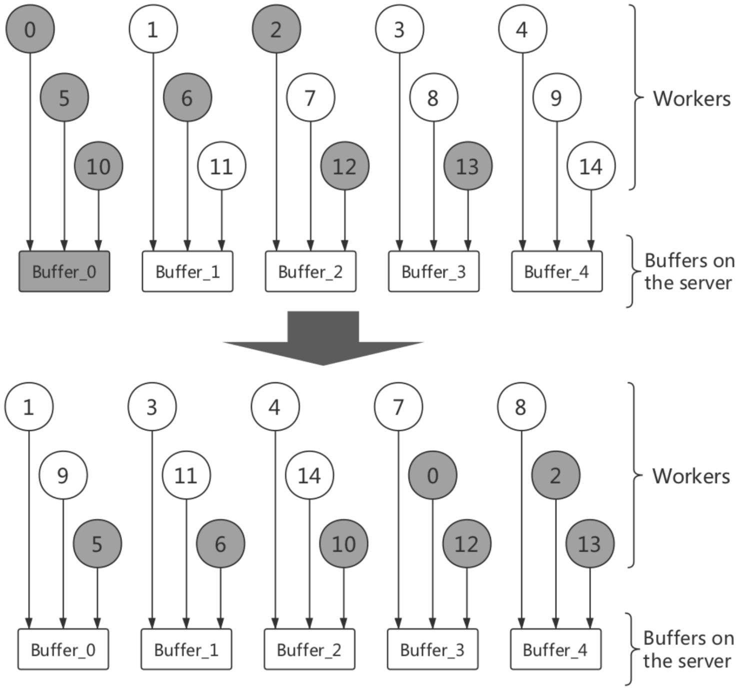


Figure 2: An example of buffer reassignment. The white circles represent active workers, and the grey circles represent unresponsive workers. Before reassignment, buffer_0 is a straggler. After reassignment, there is at least one active worker corresponding to each buffer.

3.3 Mapping Table

At each iteration of BASGD, buffer_b needs at least one gradient for aggregation. In the worst case, all the workers corresponding to buffer_b may be unresponsive. In this case, buffer_b will become the straggler, and slow down the whole learning process. To deal with this problem, we introduce the mapping table for buffer reassignment.

We call a worker active worker if it has responded at the current iteration. If the SGD step has not been executed for Δ seconds, the server immediately zeroes out stored gradients in all buffers, equally reassigns active workers to each buffer, and then continues the learning procedure. Hyper-parameter Δ is called reassignment interval. Figure 2 illustrates an example of reassignment. The grey circles represent unresponsive workers. After reassignment, there is at least one active worker corresponding to each buffer.

Specifically, we introduce a mapping table $\{\beta_s\}_{s=0}^{m-1}$ for buffer reassignment. Initially, $\beta_s = s$ ($\forall s = 0, 1, \ldots, m-1$). When reassigning buffers, the server only needs to modify the mapping table $\{\beta_s\}_{s=0}^{m-1}$, and then stores worker_s's gradients in buffer_$\{\beta_s \bmod B\}$, instead of buffer_$\{s \bmod B\}$. Please note that the server only needs to modify the mapping table for buffer reassignment, and there is no need to notify workers.

In addition, a timer is used on the server for indicating when to reassign buffers. The timer is started at the beginning of BASGD and is restarted immediately after each SGD step or buffer reassignment. When the timer exceeds Δ seconds, buffers will be zeroed out and then reassigned. Hyper-parameter Δ should be set properly. If Δ is too small, buffers will be zeroed out too frequently, which may slow down the learning process. If Δ is too large, straggler buffers could not be eliminated in time. In practical applications, practitioners can collect statistics about workers' time cost on computing gradients (or momentums), and then properly set Δ to make that more than B workers are usually able to finish the computation and send messages in Δ seconds.

3.4 Buffered Asynchronous SGD with Momentum

As previous works have revealed, history information can greatly help to resist Byzantine attacks (El-Mhamdi et al., 2021b; Allen-Zhu et al., 2020; Karimireddy et al., 2021). Therefore, we introduce momentum into BASGD and obtain the method called buffered asynchronous SGD with momentum (BASGDm). In BASGDm, the algorithm of the server is exactly the same as that in BASGD. The only difference is that each worker maintains a local momentum, and sends local momentums to the server instead of gradients. The detail of BASGDm is illustrated in Algorithm 2. With the benefit of momentum, BASGDm can achieve stronger Byzantine resilience. In particular, BASGDm has a significantly better empirical performance than BASGD, as we will show in Section 5.

Meanwhile, we have noticed that the work in (Nguyen et al., 2022) proposes the method FedBuff, which also adopts buffered asynchronous aggregation. However, the motivation of BASGD (BASGDm) and FedBuff significantly differ from each other. FedBuff mainly focuses on privacy preservation in federated learning while BASGD and BASGDm are for asynchronous Byzantine learning.

4. Convergence

In this section, we theoretically prove the convergence and resilience of BASGD and BASGDm against failure or attack. We will introduce four main theorems in this section. The first theorem is for BASGD with q-BR aggregation functions. The second and the third theorems are for BASGD and BASGDm with $(\delta_{\max}, A_1, A_2)$-effective aggregation functions, respectively. The last theorem is for BASGDm with $(\delta_{\max}, c)$-robust aggregation functions in i.i.d. cases. Furthermore, the last theorem also shows the effectiveness of using local momentum in asynchronous Byzantine learning.

Here we only present the results. Proof details are in Appendix B. We first make the following assumptions, which have been widely used in stochastic optimization.

Assumption 1 (Lower bound)**.** *Global loss function $F(\mathbf{w})$ is bounded below: $\exists F^* \in \mathbb{R}, F(\mathbf{w}) \geq F^*, \forall \mathbf{w} \in \mathbb{R}^d$.*

Assumption 2 (Bounded bias)**.** *For any loyal worker_k, it can use locally stored training instances to obtain an estimation of the global gradient with bounded bias κ: $\exists \kappa \in \mathbb{R}_+$, $\|\mathbb{E}_{i \sim \mathcal{D}_k}[\nabla f(\mathbf{w}; z_i)] - \nabla F(\mathbf{w})\| \leq \kappa,\ \forall \mathbf{w} \in \mathbb{R}^d$.*

Assumption 3 (Bounded gradient)**.** *Global loss function $F(\mathbf{w})$ has a bounded gradient: $\exists D \in \mathbb{R}_+$, $\|\nabla F(\mathbf{w})\| \leq D,\ \forall \mathbf{w} \in \mathbb{R}^d$.*

Algorithm 2 Buffered Asynchronous SGD with Momentum (BASGDm)

```
Server:
Input: learning rate $\eta$, momentum hyper-parameter $\mu$ ($0 \leq \mu < 1$),
  reassignment interval $\Delta$, buffer number $B$, aggregation function: $Aggr(\cdot)$;
Initialization: model parameter $\mathbf{w}^0$;
Set $\mathbf{h}_b \leftarrow \mathbf{0}$ and $N_b^0 \leftarrow 0$ for all $b = 0, \ldots, B-1$;
Initialize mapping table $\beta_s \leftarrow s$ ($s = 0, 1, \ldots, m-1$);
Send initial $\mathbf{w}^0$ to all workers;
Set $t \leftarrow 0$, and start the timer;
repeat
  Wait until receiving $\mathbf{u}$ from some worker_$s$;
  Choose buffer: $b \leftarrow \beta_s \ mod \ B$;
  Let $N_b^t \leftarrow N_b^t + 1$, and $\mathbf{h}_b \leftarrow \frac{(N_b^t - 1)\mathbf{h}_b + \mathbf{u}}{N_b^t}$;
  if $N_b^t > 0$ for each $b \in \{0, \ldots, B-1\}$ then
    Aggregate: $\mathbf{G}^t = Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}])$;
    Execute SGD step: $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta \cdot \mathbf{G}^t$;
    Zero out buffers: $\mathbf{h}_b \leftarrow \mathbf{0}$, $N_b^t \leftarrow 0$ ($b = 0, \ldots, B-1$);
    Set $t \leftarrow t + 1$, and restart the timer;
  end if
  if the timer has exceeded $\Delta$ seconds then
    Zero out buffers: $\mathbf{h}_b \leftarrow \mathbf{0}$, $N_b^t \leftarrow 0$ ($b = 0, \ldots, B-1$);
    Modify the mapping table $\{\beta_s\}_{s=0}^{m-1}$ for buffer reassignment, and restart the timer;
  end if
  Send the latest parameters back to worker_$s$, no matter whether an SGD step is executed
  or not.
until stop criterion is satisfied
Notify all workers to stop;

Worker_$k$:   ($k = 0, 1, ..., m-1$)
Initialization: initial momentum $\mathbf{u} \leftarrow \mathbf{0}$;
repeat
  Wait until receiving the latest parameter $\mathbf{w}$ from server;
  Randomly sample an index $i$ from $\mathcal{D}_k$;
  Compute stochastic gradient $\nabla f(\mathbf{w}; z_i)$;
  Update local momentum $\mathbf{u} \leftarrow \begin{cases} \nabla f(\mathbf{w}; z_i), & \text{at the first iteration;} \\ \mu \cdot \mathbf{u} + (1-\mu) \cdot \nabla f(\mathbf{w}; z_i), & \text{otherwise;} \end{cases}$
  Send $\mathbf{u}$ to server;
until receive server's notification to stop
```

Assumption 4 (Bounded variance). *For any loyal worker_k, the local stochastic gradient has a bounded variance:* $\exists \sigma \in \mathbb{R}_+$, $\mathbb{E}_{i \sim \mathcal{D}_k} ||\nabla f(\mathbf{w}; z_i) - \mathbb{E}_{i \sim \mathcal{D}_k}[\nabla f(\mathbf{w}; z_i)]||^2 \leq \sigma^2$, $\forall \mathbf{w} \in \mathbb{R}^d$.

Assumption 5 (L-smoothness). *Global loss function* $F(\mathbf{w})$ *is differentiable and* L*-smooth:* $||\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')|| \leq L||\mathbf{w} - \mathbf{w}'||$, $\forall \mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$.

Compared with the case of synchronous Byzantine learning, the threat of Byzantine attacks could be enlarged in asynchronous settings due to the bias caused by asynchrony. The bounded gradient assumption is mainly used to provide upper bounds for the bias of stochastic gradients (in BASGD) or local momentums (in BASGDm) caused by the combined effect of asynchrony and Byzantine attacks. Although $\nabla F(\mathbf{w})$ is bounded, Byzantine workers are allowed to send vectors with arbitrary values.

Then we first analyze the convergence of BASGD with q-BR aggregation functions. Let $N^{(t)}$ be the $(q+1)$-th smallest value in $\{N_b^t\}_{b\in\{0,\ldots,B-1\}}$, where N_b^t is the number of gradients (or momentums) stored in buffer_b at the t-th iteration. We use r to denote the total number of heavily delayed workers and Byzantine workers, and define the constant

$$\Theta_{B,q,r} = \frac{(B-r)\sqrt{B-r+1}}{\sqrt{(B-q-1)(q-r+1)}},$$

which will appear in Lemma 9 and Lemma 10.

Lemma 9. *If $Aggr(\cdot)$ is q-BR and the total number of heavily delayed workers and Byzantine workers is not larger than r ($r \leq q$), under Assumptions 3 and 4, we have:*

$$\mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \leq \Theta_{B,q,r} d \cdot (D^2 + \sigma^2/N^{(t)}).$$

Lemma 10. *If $Aggr(\cdot)$ is q-BR, and the total number of heavily delayed workers and Byzantine workers is not larger than r ($r \leq q$), under Assumptions 2, 3, 4 and 5, we have:*

$$||\mathbb{E}[\mathbf{G}^t - \nabla F(\mathbf{w}^t) \mid \mathbf{w}^t]|| \leq \Theta_{B,q,r} d(\tau_{max} L \cdot [\Theta_{B,q,r} d(D^2 + \sigma^2/N^{(t)})]^{\frac{1}{2}} + \sigma + \kappa).$$

Theorem 11. *Let $\tilde{D} = \frac{1}{T}\sum_{t=0}^{T-1}(D^2 + \sigma^2/N^{(t)})^{\frac{1}{2}}$. If the total number of Byzantine workers and heavily delayed workers at each iteration is not larger than r, $Aggr(\cdot)$ is q-BR where $q = r$, under Assumptions 1, 2, 3, 4 and 5, we have the following result for BASGD with learning rate $\eta = O(\frac{1}{L\sqrt{T}})$:*

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}[||\nabla F(\mathbf{w}^t)||^2]}{T} \leq O\left(\frac{L[F(\mathbf{w}^0) - F^*]}{T^{\frac{1}{2}}}\right) + O\left(\frac{2(1-\delta_{\max})rd\tilde{D}}{\delta_{\max}T^{\frac{1}{2}}}\right)$$
$$+ O\left(\frac{2(1-\delta_{\max})rDd\sigma}{\delta_{\max}} + \frac{2(1-\delta_{\max})rDd\kappa}{\delta_{\max}} + \frac{2\sqrt{2}(1-\delta_{\max})^{\frac{3}{2}} r^{\frac{3}{2}} LD\tilde{D}d^{\frac{3}{2}}\tau_{max}}{(\delta_{\max})^{\frac{3}{2}}}\right),$$

where $\delta_{\max} = \frac{q}{B}$.

Please note that the convergence rate of vanilla ASGD is $O(1/T^{\frac{1}{2}})$. Hence, Theorem 11 indicates that BASGD has a theoretical convergence rate as fast as vanilla ASGD, with an extra constant variance. The term $O(2(1-\delta_{\max})rDd\sigma/\delta_{\max})$ is caused by the aggregation function, which can be deemed as a sacrifice for Byzantine resilience. The term $O(2(1-\delta_{\max})rDd\kappa/\delta_{\max})$ is caused by the differences in training instances among different workers. In independent and identically distributed (i.i.d.) cases, $\kappa = 0$ and the term vanishes. The term $O(2\sqrt{2}(1-\delta_{\max})^{\frac{3}{2}} r^{\frac{3}{2}} LD\tilde{D}d^{\frac{3}{2}}\tau_{max}/\delta_{\max}^{\frac{3}{2}})$ is caused by the delay, and related to parameter τ_{max}. The term is also related to the buffer size since $\delta_{\max} = \frac{q}{B}$. When

N_b^t increases, $N^{(t)}$ may increase, and thus $\tilde{D}$ will decrease. Namely, a larger buffer size will result in smaller $\tilde{D}$. In addition, the factor $(1-\delta_{\max})r/\delta_{\max}$ or $(1-\delta_{\max})^{\frac{3}{2}}r^{\frac{3}{2}}/\delta_{\max}^{\frac{3}{2}}$ decreases as $\delta_{\max}$ increases, and increases as r increases.

Then, we present the convergence results for BASGD and BASGDm with $(\delta_{\max}, A_1, A_2)$-effective aggregation functions (please refer to Definition 7) in Theorem 12 and Theorem 13, respectively.

Theorem 12. *In BASGD, if the total number of Byzantine workers and heavily delayed workers at each iteration is not larger than r, $Aggr(\cdot)$ is a $(\delta_{\max}, A_1, A_2)$-effective aggregation function, $B = \lfloor r/\delta_{\max} \rfloor + 1$ and the learning rate $\eta = O(\frac{1}{\sqrt{LT}})$ satisfies that $2\eta^2 L^2 \tau_{max}^2 (B - r) < 1$, under Assumption 1, 3, 4 and 5, we have the following result for general asynchronous cases:*

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2]}{T} \leq O\left(\frac{L^{\frac{1}{2}}[F(\mathbf{w}^0) - F^*]}{T^{\frac{1}{2}}}\right) + O\left(\frac{(r - \delta_{\max} r + 1)^{\frac{1}{2}} L^{\frac{1}{2}} \tau_{max} D A_2}{\delta_{\max}^{\frac{1}{2}} T^{\frac{1}{2}}}\right)$$
$$+ O\left(\frac{L^{\frac{1}{2}}(A_2)^2}{T^{\frac{1}{2}}}\right) + O\left(\frac{(r - \delta_{\max} r + 1) L^{\frac{3}{2}} (A_2)^2 \tau_{max}^2}{\delta_{\max} T^{\frac{3}{2}}}\right) + A_1.$$

Theorem 12 indicates that if $Aggr(\cdot)$ makes a synchronous BL method converge (i.e., satisfies Definition 7), BASGD converges when using $Aggr(\cdot)$ as aggregation function. Hence, BASGD can also be seen as a technique of asynchronization. That is to say, new asynchronous methods can be obtained from synchronous ones when using BASGD. The factor $\frac{r-\delta_{\max}r+1}{\delta_{\max}}$ equals $(\frac{1-\delta_{\max}}{\delta_{\max}}r + \frac{1}{\delta_{\max}})$, which decreases as $\delta_{\max}$ increases and increases as r increases. The extra constant term A_1 is caused by gradient bias. When there is no Byzantine or heavily delayed workers ($r = 0$) and instances are i.i.d. across workers, letting $B = 1$ and $Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}]) = Aggr(\mathbf{h}_0) = \mathbf{h}_0$, BASGD degenerates to vanilla ASGD. In this case, there is no gradient bias ($A_1 = 0$), and BASGD has a convergence rate of $O(1/T^{\frac{1}{2}})$. Similarly, we have the following theoretical results for BASGDm.

Theorem 13. *In BASGDm, if the total number of Byzantine workers and heavily delayed workers at each iteration is not larger than r, $Aggr(\cdot)$ is a $(\delta_{\max}, A_1, A_2)$-effective aggregation function, $B = \lfloor r/\delta_{\max} \rfloor + 1$ and the learning rate $\eta = O(\frac{1}{\sqrt{LT}})$ satisfies that $2\eta^2 L^2 \tau_{max}^2 (1-\mu)^2 < 1$, under Assumption 1, 3, 4 and 5, we have the following result for general asynchronous cases:*

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2]}{T} \leq O\left(\frac{L^{\frac{1}{2}}[F(\mathbf{w}^0) - F^*]}{T^{\frac{1}{2}}}\right) + O\left(\frac{(r - \delta_{\max} r + 1)^{\frac{1}{2}} L^{\frac{1}{2}} \tau_{max} D A_2 (1-\mu)}{\delta_{\max}^{\frac{1}{2}} T^{\frac{1}{2}}}\right)$$
$$+ O\left(\frac{L^{\frac{1}{2}}(A_2)^2}{T^{\frac{1}{2}}}\right) + O\left(\frac{(r - \delta_{\max} r + 1) L^{\frac{3}{2}} (A_2)^2 \tau_{max}^2 (1-\mu)^2}{\delta_{\max} T^{\frac{3}{2}}}\right) + A_1.$$

Please note that when momentum hyper-parameter $\mu = 0$, BASGDm degenerates to BASGD. In this case, $1 - \mu = 1$, and Theorem 13 is exactly the same as Theorem 12. From this perspective, Theorem 13 can be deemed as a more general version of Theorem 12. In addition, we would also like to point out that the factor $(1-\mu)$ in Theorem 13 does not mean that a larger μ will lead to a tighter upper bound since constants A_1 and A_2 are dependent on momentum hyper-parameter μ. In fact, the influence of momentum hyper-parameter is a complex problem, which has been studied for decades (Qian, 1999). Since it is not the focus of this work, we are not going to further discuss this problem here.

In general cases, Theorem 12 and Theorem 13 guarantee BASGD and BASGDm to find a point such that the squared L_2-norm of its gradient is not larger than a positive number close to A_1 in expectation, respectively. Please note that Assumption 3 already guarantees that the gradient's squared L_2-norm is not larger than D^2. We introduce Proposition 14 to show that A_1 is guaranteed to be smaller than D^2 under a mild condition.

Proposition 14. *Assume that $Aggr(\cdot)$ is a $(\delta_{\max}, A_1, A_2)$-effective aggregation function, and $\mathbf{G}_{syn}^t$ is aggregated by $Aggr(\cdot)$ in synchronous setting. If $\mathbb{E}[\|\mathbf{G}_{syn}^t - \nabla F(\mathbf{w}^t)\| \mid \mathbf{w}^t] \leq D' < D$, $\forall \mathbf{w}^t \in \mathbb{R}^d$, we have $A_1 \leq D'D < D^2$.*

The aggregated result $\mathbf{G}_{syn}^t$ can be viewed as a robust estimator of $\nabla F(\mathbf{w}^t)$ used for updating. Since $\|\nabla F(\mathbf{w}^t)\| \leq D$, $\nabla F(\mathbf{w}^t)$ locates in a ball centered at the origin with radius D. $\mathbb{E}[\|\mathbf{G}_{syn}^t - \nabla F(\mathbf{w}^t)\| \mid \mathbf{w}^t] \leq D' < D$ means that the bias of $\mathbf{G}_{syn}^t$ is not larger than the radius D' ($D' < D$), which is a mild condition for $Aggr(\cdot)$.

In Theorem 11, Theorem 12 and Theorem 13, there exist constant variance terms, which will not decrease during the training. Recent works (Karimireddy et al., 2021) have shown that when the aggregation function is $(\delta_{\max}, c)$-robust (please refer to Definition 8) and the momentum hyper-parameter is properly set, synchronous Byzantine learning methods can reach the convergence rate of $O(1/T^{\frac{1}{2}})$ (without constant term) in i.i.d. cases where the bias $\kappa = 0$ in Assumption 2. We show that BASGDm can also achieve a similar convergence rate when $\kappa = 0$. The detailed results are presented in Theorem 15 as follows.

Theorem 15. *Let $\lambda = 1 - \mu$. When $Aggr(\cdot)$ is a $(\delta_{\max}, c)$-robust aggregation function, the total number of Byzantine workers and heavily delayed workers at each iteration is not larger than $B\delta_{\max}$, and learning rate $\eta \leq \frac{1}{L}$, under Assumption 1, 2, 3, 4 and 5, we have the following result for BASGDm when $\kappa = 0$ (i.i.d. case):*

$$\frac{\sum_{t=0}^{T-1} \mathbb{E}||\nabla F(\mathbf{w}^t)||^2}{T} \leq \frac{2[F(\mathbf{w}^0) - F^*]}{\eta T} + \frac{2(4c\delta + 1)(\tau_{max} + 1)\sigma^2}{T} + \zeta, \tag{2}$$

where δ is the fraction of Byzantine workers and heavily delayed workers together and $\zeta = 2(4c\delta + 1)\lambda^2\sigma^2 + 8(4c\delta + 1)^2 \left[4 - \lambda + 2\sqrt{4 - 2\lambda + 4\lambda^{-2}} + 2\lambda^{-2}\right] \eta^2 L^2 (\tau_{max} + 1)^2 D^2.$

The right-hand side (RHS) of inequality (2) depends on η and λ, where $\lambda = 1 - \mu$ and μ is the momentum hyper-parameter. Then we discuss the convergence results for different settings of λ.

Firstly, for BASGD without momentum, we have that $\mu = 0$ and $\lambda = 1 - \mu = 1$. In this case, for any η, the term ξ in the RHS of (2) is always larger than $2(4c\delta + 1)\sigma^2$ and thus cannot converge to 0. It is consistent with the results in previous works (Karimireddy et al., 2021) that the constant term is inevitable without using history information such as local momentum when there are Byzantine workers. As we will detailedly show in Proposition 16 below, a better choice for the momentum hyper-parameter in BASGDm is that $\lambda = \sqrt{\eta L}$.

Proposition 16. *Under the same conditions in Theorem 15, when* $\lambda = \sqrt{\eta L} \leq 1$ *and* $\eta = \min\left(\sqrt{\frac{F(\mathbf{w}^0)-F^*}{LT(4c\delta+1)[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]}}, \frac{1}{L}\right)$*, we have*

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}||\nabla F(\mathbf{w}^t)||^2}{T} \leq \frac{4L^{\frac{1}{2}}[F(\mathbf{w}^0)-F^*]^{\frac{1}{2}}(4c\delta+1)^{\frac{1}{2}}[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]^{\frac{1}{2}}}{T^{\frac{1}{2}}} + \frac{14L^{\frac{3}{4}}[F(\mathbf{w}^0)-F^*]^{\frac{3}{4}}(4c\delta+1)^{\frac{1}{2}}(\tau_{max}+1)^{\frac{1}{2}}D^{\frac{1}{2}}}{T^{\frac{3}{4}}} + \frac{6L[F(\mathbf{w}^0)-F^*]+2(4c\delta+1)(\tau_{max}+1)\sigma^2}{T}.$$

Proposition 16 shows that the ABL method BASGDm can converge to a stationary point with the convergence rate of $O(1/\sqrt{T})$, which is the same as that in SBL (Karimireddy et al., 2021). To the best of our knowledge, this is the first theoretical result indicating that using momentum can also enhance the resilience against Byzantine attacks in ABL.

In addition, we would like to point out that using local momentum on workers does not help to reduce the bias in non-i.i.d. cases. Theorem 15, Proposition 15 and the corresponding theoretical results in previous works (Karimireddy et al., 2021) for SBL are all based on the i.i.d. assumption. As far as we know, how to enhance Byzantine resilience for non-i.i.d. cases in asynchronous settings is still a challenging open problem. It is beyond the scope of this work and we leave it for future work.

Then we discuss the convergence rate with respect to L, T, and τ_{max}. Proposition 16 shows that BASGDm can achieve the convergence rate of $O(L^{\frac{1}{2}}/T^{\frac{1}{2}}) + O(L^{\frac{1}{2}}\tau_{max}/T^{\frac{1}{2}})$ in i.i.d. cases. Meanwhile, it is shown in previous works (Liu and Zhang, 2021) that vanilla ASGD can achieve the convergence rate of $O(L^{\frac{1}{2}}/T^{\frac{1}{2}}) + O(L\tau_{max}/T)$. Compared to vanilla ASGD, the convergence result of BASGDm is more sensitive to τ_{max}. We would like to point out that in ABL, the bias caused by asynchrony will leave more room for Byzantine attacks. Thus, it is reasonable that the convergence result of BASGDm under attacks is slightly looser than that of vanilla ASGD without attack. In addition, it remains uncertain whether the dependence on the staleness parameter τ_{max} in Proposition 16 is tight. To the best of our knowledge, there are almost no works revealing the tightness of τ_{max} in ABL.

Finally, we would like to summarize the theoretical results presented in this section. Theorem 11 is for BASGD with q-BR aggregation functions, the definition of which is relatively easy to check. Theorem 12 and Theorem 13 are for BASGD and BASGDm with $(\delta_{\max}, A_1, A_2)$-effective aggregation functions, respectively. The two theorems can be deemed as a bridge across ABL and SBL. The three theorems above (Theorem 11, Theorem 12 and Theorem 13) are for general non-i.i.d. cases, and constant error terms appear in the three theorems. As far as we know, it is still an open problem whether the constant error terms can be removed for ABL methods in general non-i.i.d. cases. Theorem 15 and Proposition 16 theoretically show the effectiveness of using local momentum for ABL methods in i.i.d. cases, which is consistent with previous works on SBL (Karimireddy et al., 2021).

5. Experiment

In this section, we empirically evaluate the performance of BASGD (BASGDm) and baselines in both image classification (IC) and natural language processing (NLP) applications. Our experiments are conducted on a distributed platform with dockers. Each docker is bound to an NVIDIA Tesla V100 (32G) GPU. We choose 30 dockers as workers and an extra docker as server[1]. All algorithms are implemented with PyTorch 1.3.

5.1 Experimental Setting

The performance of decentralized methods (El-Mhamdi et al., 2021a) will depend on the network topology, which makes it hard to conduct a fair comparison. Thus, we mainly consider methods under the PS framework in our experiments. Moreover, the server has no access to any training instances. The ABL methods Zeno++ (Xie et al., 2020b) and Sageflow (Park et al., 2021) need to store some instances on the server, and thus not applicable in the settings of this work. Hence, we compare BASGD (BASGDm) with baselines ASGD (ASGDm) and Kardam in our experiments. We set dampening function $\Lambda(\tau) = \frac{1}{1+\tau}$ for Kardam as suggested in (Damaskinos et al., 2018), and set momentum hyper-parameter $\mu = 0.9$ for BASGDm and ASGDm in each experiment.

Byzantine attacks. We will compare BASGD and BASGDm with baselines under the following different attack settings.

- No attack: In this setting, each worker will strictly follow the method, compute and send the gradient (or momentum) without error.

- Random disturbance attack (RD-attack): Byzantine workers with RD-attack will replace the true gradient (or momentum) $\mathbf{g}$ with $\tilde{\mathbf{g}}_{RD} = \mathbf{g} + \mathbf{g}_{rnd}$, where $\mathbf{g}_{rnd}$ is a random vector sampled from the normal distribution $\mathcal{N}(\mathbf{0}, \|\sigma_{atk}\mathbf{g}\|^2 \cdot \mathbf{I})$. Here, σ_{atk} is a parameter and $\mathbf{I}$ is the d-by-d identity matrix. We set $\sigma_{atk} = 0.2$ in our experiments. RD-attack can be seen as an accidental failure with expectation $\mathbf{0}$.

- Negative gradient attack (NG-attack): Byzantine workers with NG-attack will replace the true gradient (or momentum) $\mathbf{g}$ with $\tilde{\mathbf{g}}_{NG} = -k_{atk} \cdot \mathbf{g}$, where $k_{atk} \in \mathbb{R}_+$ is a parameter. We set $k_{atk} = 10$ in our experiments. NG-attack is a typical kind of malicious attack. In some previous works, this type of attack is also called bit-flipping attack (Xie et al., 2020b; Karimireddy et al., 2021).

- 'Fall of Empires' (FoE) attack (Xie et al., 2020a): Byzantine workers with FoE attack will replace the gradient (or momentum) $\mathbf{g}$ with $\tilde{\mathbf{g}}_{FoE} = -\frac{\epsilon}{|\mathcal{L}|}\sum_{i\in\mathcal{L}} \mathbf{g}_i$, where $\mathcal{L}$ is the index set of loyal workers and $\mathbf{g}_i$ is the gradient (or momentum) computed by the i-th worker at the same iteration. We set hyper-parameter $\epsilon = 6$ for FoE attack in the experiments of this work. FoE is a type of omniscient attack originally proposed in synchronous settings, which requires the gradients (or momentums) computed by loyal workers at the same iteration as omniscient knowledge. Thus, FoE cannot be directly adopted in asynchronous settings. To deal with this problem, we use the last sent gradient (or momentum) from each loyal worker as the omniscient knowledge for FoE.

[1] In the conference version (Yang and Li, 2021), we set 8 workers in the NLP experiment. To make the settings more consistent with that of the IC experiment, we also set the worker number to 30 for the NLP experiment in this journal version.

- 'A Little is Enough' (ALIE) attack (Baruch et al., 2019): Byzantine workers with ALIE attack will replace the gradient (or momentum) $\mathbf{g}$ with $\tilde{\mathbf{g}}_{ALIE}$, where $(\tilde{\mathbf{g}}_{ALIE})_j = mean_j - z^{max} \cdot std_j$. The sub-index $(\cdot)_j$ denotes the j-th coordinate of the vector. The scalars $mean_j$ and std_j are the mean and standard error of the j-th coordinate of loyal workers' gradients (or momentums) at the same iteration, respectively. $z^{max} = \Phi^{-1}(\frac{m - \lfloor m/2+1 \rfloor}{m-r})$, where $\Phi^{-1}(\cdot)$ is the inverse of the standard normal cumulative distribution function, m is the number of workers, and r is the number of Byzantine workers. ALIE is also a type of omniscient attack originally proposed in synchronous settings. Similarly, to make it compatible with asynchronous settings, we use the last sent gradient (or momentum) from each loyal worker as the omniscient knowledge for ALIE.

In real-world applications, it is usually hard to adopt the two types of omniscient attacks (FoE and ALIE) due to the lack of omniscient knowledge. However, we still compare the performance of different methods under these two attacks to evaluate resilience ability.

Aggregation rules. In the experiments, BASGD and BASGDm are evaluated with the following aggregation rules.

- Coordinate-wise q-trimmed-mean (trmean): Please refer to Definition 4.
- Coordinate-wise median (median): Please refer to Definition 3. Since median can be deemed as a special case of trmean, we only report the results of BASGD and BASGDm with median in the case of no attack[2].
- Geometric median (geoMed) (Chen et al., 2017): The geometric median of B vectors $\mathbf{h}_0, \ldots, \mathbf{h}_{B-1} \in \mathbb{R}^d$ is defined as:

$$\text{geoMed}([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}]) = \mathop{\arg\min}_{\mathbf{h} \in \mathbb{R}^d} \left\{ \sum_{b=0}^{B-1} \|\mathbf{h} - \mathbf{h}_b\|_2 \right\}. \tag{3}$$

  The optimization problem defined in the right-hand side of (3) has a unique solution when vectors $\{\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}\}$ do not lie in a line. However, geoMed usually does not have a closed-form solution. We use Weiszfeld's algorithm (Pillutla et al., 2019) to compute it and set the iteration number in Weiszfeld's algorithm to be 5.

- Centered clipping (CC) (Karimireddy et al., 2021): The CC aggregation result of vectors $\{\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}\}$ is given by the following iteration formula:

$$\mathbf{h}^{l+1} = \mathbf{h}^l + \frac{1}{B} \sum_{b=0}^{B-1} (\mathbf{h}_b - \mathbf{h}^l) \min\left(1, \frac{R}{\|\mathbf{h}_b - \mathbf{h}^l\|_2}\right). \tag{4}$$

  We set initial point $\mathbf{h}^0$ to be the last aggregation result for quicker convergence as suggested in (Karimireddy et al., 2021). The iteration number is set to be 5 in the IC task and 50 in the NLP task. Clipping size R is set to be 0.5.

[2] In the conference version (Yang and Li, 2021), we report the results of BASGD with median in all cases. In this journal version, we evaluate BASGD (BASGDm) with two more aggregation rules (geometric median and centered clipping). Due to limited space in each single figure, we do not report the results of BASGD (BASGDm) with median for better readability in this journal version. The performance of median is similar to that of other aggregation rules.

Table 1: Wall-clock-time of running 160 epochs for different methods (in seconds)

Method	ASGD	BASGDm ($B = 10$)			Kardam	
		w/ trmean	w/ geoMed	w/ CC	$\gamma = 2$	$\gamma = 10$
Wall-clock-time	1172.30	1191.01	1287.07	1289.32	1522.05	1535.22

To simulate an unstable network environment where asynchronous methods are usually preferred, each worker is manually set to have a delay, which is k_{del} times the computing time. The training set is randomly and equally distributed to different workers. For space saving, we will only present the average top-1 test accuracy (in IC) or average perplexity (in NLP) in this section. Average training loss w.r.t. epochs in the IC experiment can be found in Appendix C, which is consistent with the average top-1 test accuracy results presented in this section. Unless otherwise stated, for BASGD and BASGDm, the reassignment interval is set to be 1 second in the IC experiment and 5 seconds in the NLP experiment.

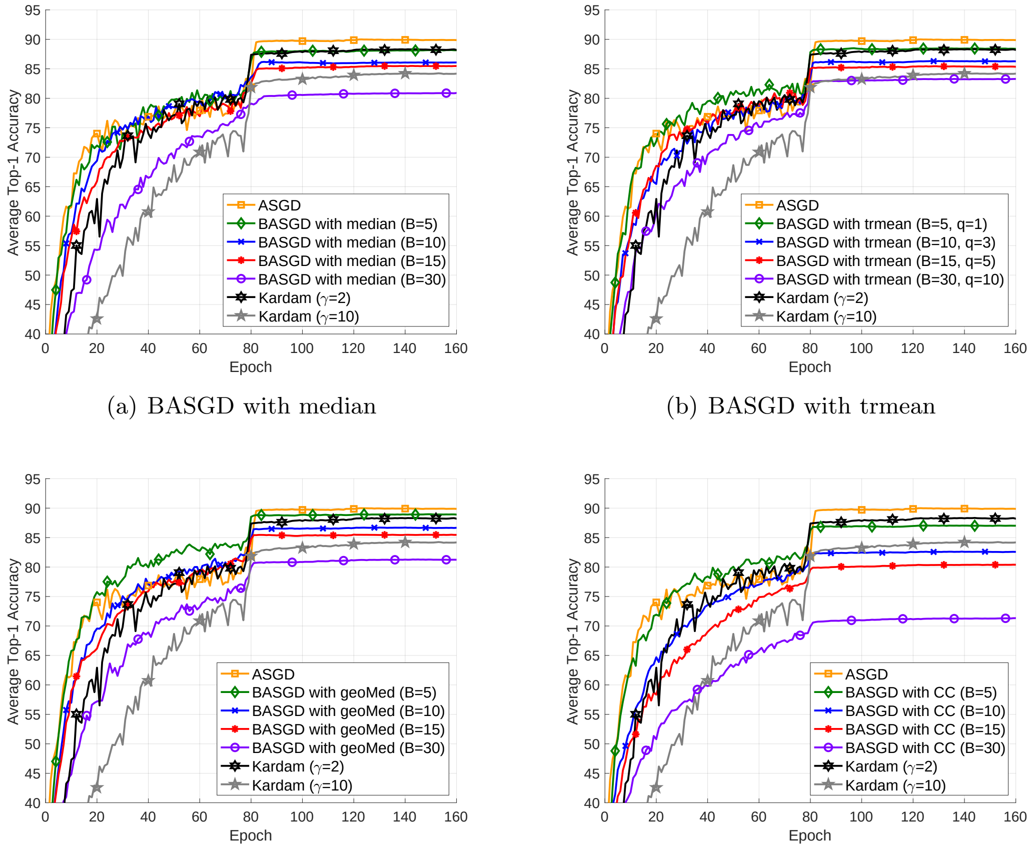
5.2 Image Classification Experiment

In this part, we will empirically compare the performance of BASGD (BASGDm) and existing asynchronous methods ASGD (ASGDm) and Kardam in image classification tasks. In the experiment, algorithms are evaluated on CIFAR-10 (Krizhevsky et al., 2009) with deep learning model ResNet-20 (He et al., 2016). Cross-entropy is used as the loss function. k_{del} is randomly sampled from truncated standard normal distribution within $[0, +\infty)$. As suggested in (He et al., 2016), learning rate η is set to 0.1 initially for each algorithm, and multiplied by 0.1 at the 80-th epoch and the 120-th epoch respectively. The weight decay is set to 10^{-4}. We run each algorithm for 160 epochs. The batch size is set to 25.

Firstly, we compare the performance of different methods when there are no Byzantine workers. Experimental results of BASGD and BASGDm are illustrated in Figure 3 and Figure 4, respectively. The solid line represents that the method does not use momentum while the dotted line represents that the method utilizes local momentum. ASGD (ASGDm) achieves the best performance. BASGD (BASGDm) ($B > 1$) and Kardam have similar convergence rates to ASGD (ASGDm), but both sacrifice a little accuracy. Furthermore, the performance of BASGD (BASGDm) gets worse when the buffer number B increases, which is consistent with the theoretical results. Please note that ASGD (ASGDm) is a degenerated case of BASGD (BASGDm) when $B = 1$ and $Aggr(\mathbf{h}_1) = \mathbf{h}_1$. Hence, BASGD (BASGDm) can achieve the same performance as ASGD (ASGDm) when there is no failure or attack. The wall clock time of running 160 epochs is reported in Table 1. The time cost of BASGDm is slightly larger than that of ASGD, while Kardam takes the most time.

Then, for each type of attack, we compare the performance of BASGD (BASGDm) and Kardam by conducting two experiments in which there are 3 and 6 Byzantine workers, respectively[3]. We respectively set 10 and 15 buffers for BASGD (BASGDm) in these two

[3] In the conference version (Yang and Li, 2021), we also report the experimental results of ASGD under attacks. However, due to limited space in figures, we do not report the results of ASGD and ASGDm in this journal version for better readability since ASGD and ASGDm are not Byzantine-resilient and achieve low accuracy under attack.



(a) BASGD with median

(b) BASGD with trmean

(c) BASGD with geoMed

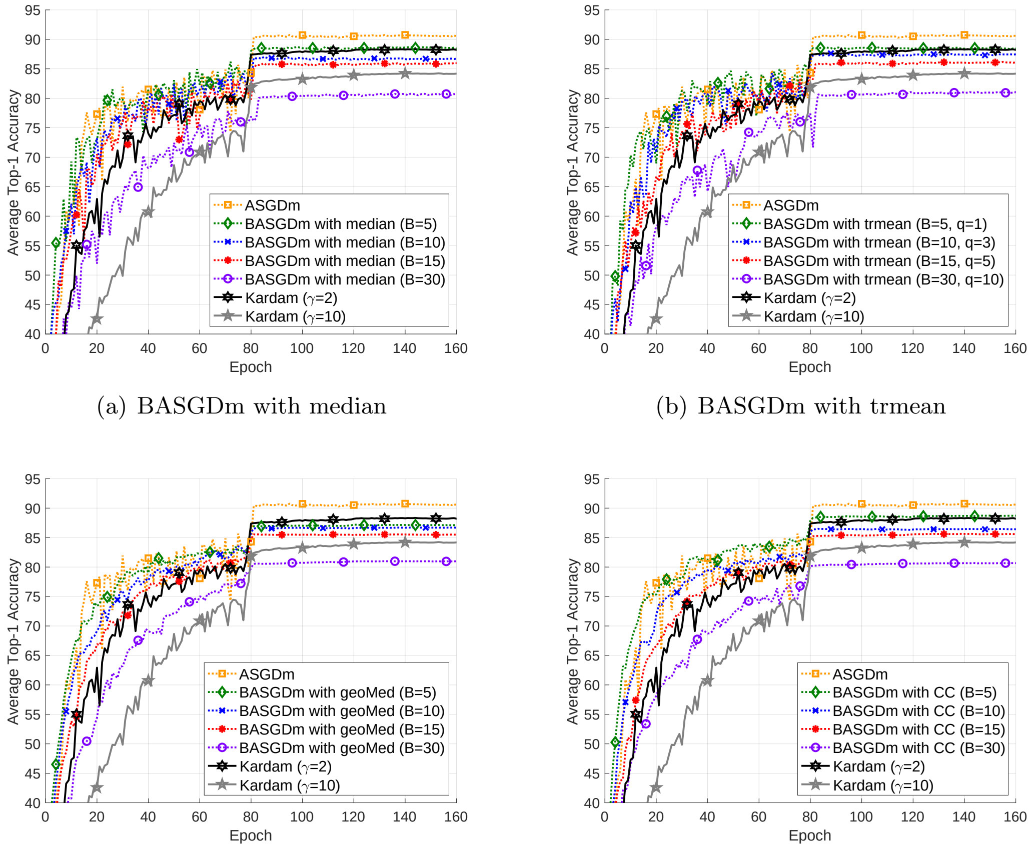
(d) BASGD with CC

Figure 3: Average top-1 test accuracy w.r.t. epochs of methods BASGD, ASGD, and Kardam when there are no Byzantine workers.

experiments. The experimental results of the methods under two types of non-omniscient attacks (RD-attack and NG-attack) are presented in Figure 5. We can find that BASGD (BASGDm) significantly outperforms Kardam under these two types of non-omniscient attacks.

Under the less harmful RD-attack, although Kardam still converges, it suffers a significant loss in accuracy. Under NG-attack, Kardam cannot converge even if we have tried different values of *assumed Byzantine worker number* for Kardam, which is denoted by the hyper-parameter γ in this paper. Hence, Kardam cannot resist these two types of attacks. On the contrary, BASGD still has a relatively good performance under these two types of attacks.

Moreover, we count the ratio of filtered gradients in Kardam, which is shown in Table 2. We can find that in order to filter Byzantine gradients, Kardam also filters approximately an equal ratio of loyal gradients. It explains why Kardam performs poorly under the attack.



(a) BASGDm with median

(b) BASGDm with trmean

(c) BASGDm with geoMed

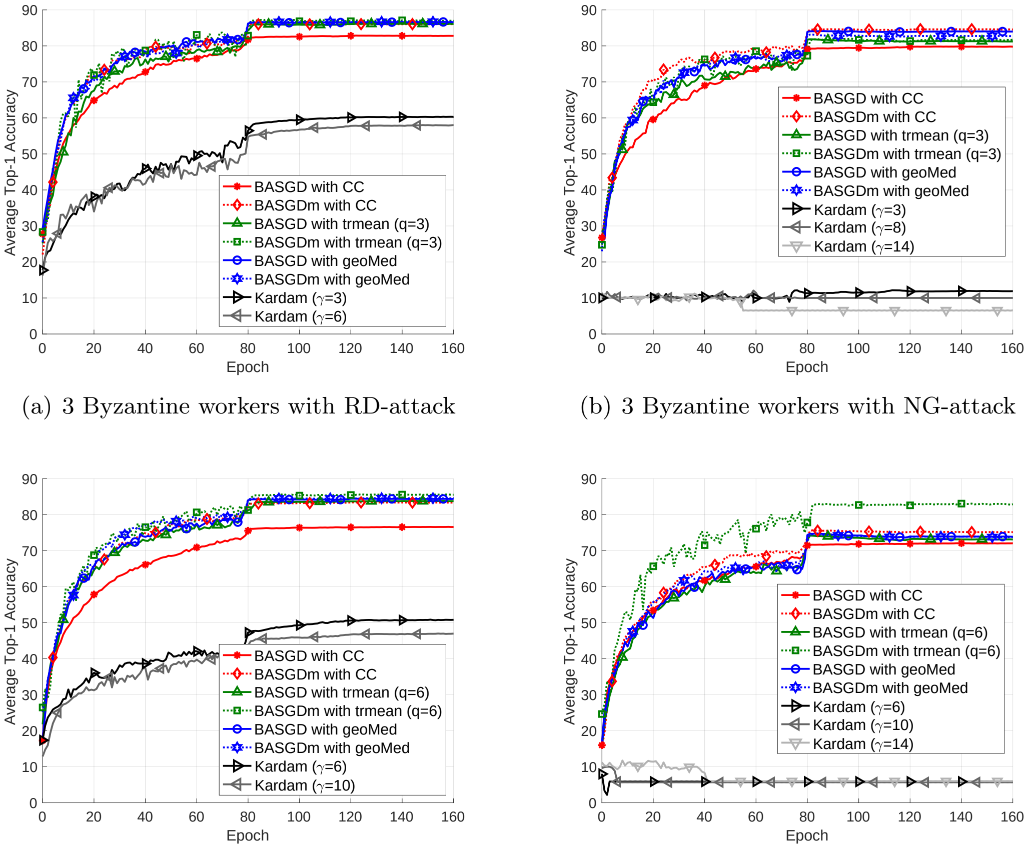
(d) BASGDm with CC

Figure 4: Average top-1 test accuracy w.r.t. epochs of methods BASGDm, ASGDm, and Kardam when there are no Byzantine workers.

Table 2: Filtered ratio in Kardam under NG-attack in IC task (3 Byzantine workers)

Term	By Frequency Filter	By Lipschitz Filter	In total
Loyal Grads ($\gamma = 3$)	10.15% (31202/307530)	40.97% (126000/307530)	51.12%
Byzt Grads ($\gamma = 3$)	10.77% (3681/34170)	40.31% (13773/34170)	51.08%
Loyal Grads ($\gamma = 8$)	28.28% (86957/307530)	28.26% (86893/307530)	56.53%
Byzt Grads ($\gamma = 8$)	28.38% (9699/34170)	28.06% (9588/34170)	56.44%
Loyal Grads ($\gamma = 14$)	85.13% (261789/307530)	3.94% (12117/307530)	89.07%
Byzt Grads ($\gamma = 14$)	84.83% (28985/34170)	4.26% (1455/34170)	89.08%

We also compare the performance of different methods under omniscient attacks (FoE attack and ALIE attack), the results of which are shown in Figure 6. BASGDm can



(a) 3 Byzantine workers with RD-attack

(b) 3 Byzantine workers with NG-attack

(c) 6 Byzantine workers with RD-attack

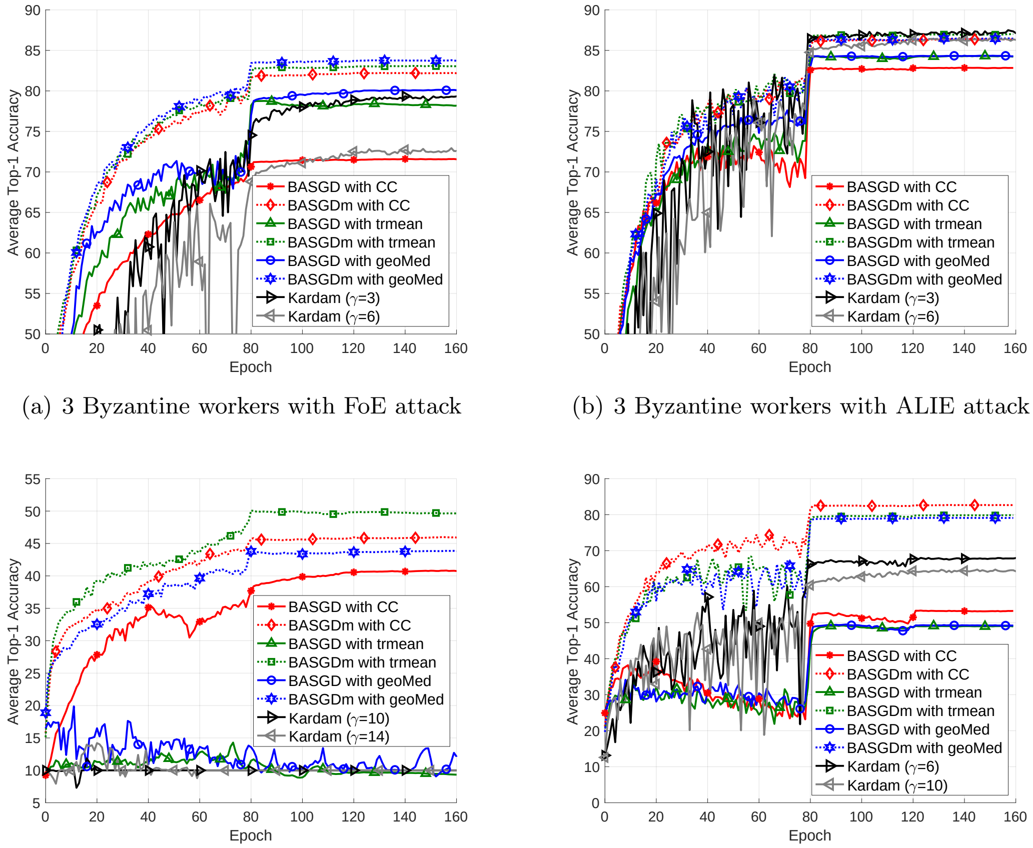
(d) 6 Byzantine workers with NG-attack

Figure 5: Average top-1 test accuracy w.r.t. epochs under non-omniscient attacks. $B = 10$ for BASGD (BASGDm) when there are 3 Byzantine workers and $B = 15$ for BASGD (BASGDm) when there are 6 Byzantine workers.

significantly outperform other methods in each case, except for the case of 3 Byzantine workers with ALIE attack. When there are 3 Byzantine workers with ALIE attack, all the methods have a comparable performance to each other. The main reason is that the Byzantine attack is not strong enough in this case. In addition, the performance of BASGDm is considerably better than BASGD. This reveals that using history information (such as momentum) can strengthen the resilience and improve the performance in Byzantine-resilient machine learning, which is consistent with previous works (Allen-Zhu et al., 2020; El-Mhamdi et al., 2021b; Karimireddy et al., 2021).

Moreover, although the performance of BASGDm with different aggregation rules (trmean, geoMed, and CC) slightly differ, all of them are better than those of BASGD and Kardam.

In addition, we evaluate the effect of the buffer reassignment interval Δ. Specifically, we will compare the performance of BASGDm when Δ is set to 0.01s, 0.1s, 1s, 10s, and



(a) 3 Byzantine workers with FoE attack

(b) 3 Byzantine workers with ALIE attack

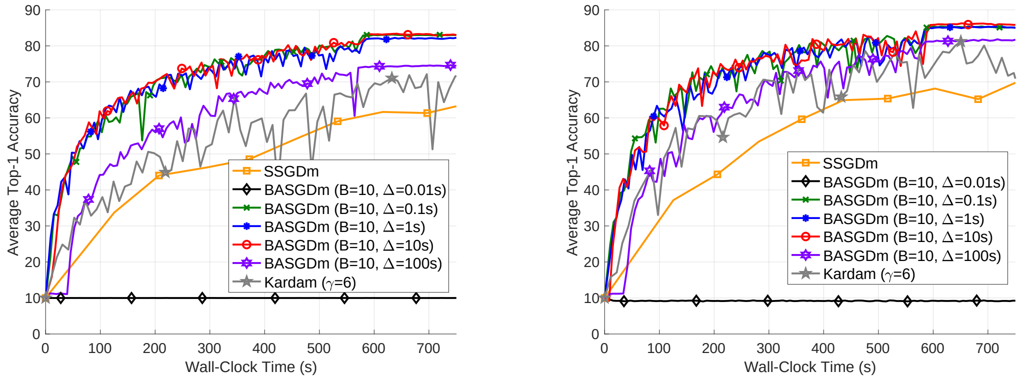
(c) 6 Byzantine workers with FoE attack

(d) 6 Byzantine workers with ALIE attack

Figure 6: Average top-1 test accuracy w.r.t. epochs under omniscient attacks. $B = 10$ for BASGD (BASGDm) when there are 3 Byzantine workers and $B = 15$ for BASGD (BASGDm) when there are 6 Byzantine workers.

100s. In this experiment, there are 3 Byzantine workers under omniscient attacks (FoE and ALIE). We set $B = 10$ for BASGDm. Since the buffer reassignment technique is used to deal with the straggler buffer in the extreme case, we set 3 extra workers to be stragglers, which take 10 times longer to finish local computation than the other workers. We also evaluate the performance of Kardam and synchronous SGD with momentum (SSGDm) in this setting. For both SSGDm and BASGDm, the aggregation rule is set to be trmean and the momentum hyper-parameter μ is set to 0.9. The average top-1 test accuracy w.r.t. wall-clock time of different methods is illustrated in Figure 7.

As we can see, the value of buffer reassignment interval Δ significantly affects the performance of BASGDm. When Δ is set too small ($\Delta = 0.01$s), buffers will be zeroed out too frequently and the global model updating is hardly executed on the server. When Δ is set too large ($\Delta = 100$s), the straggler buffer will not be promptly eliminated by buffer



(a) Under FoE attack

(b) Under ALIE attack

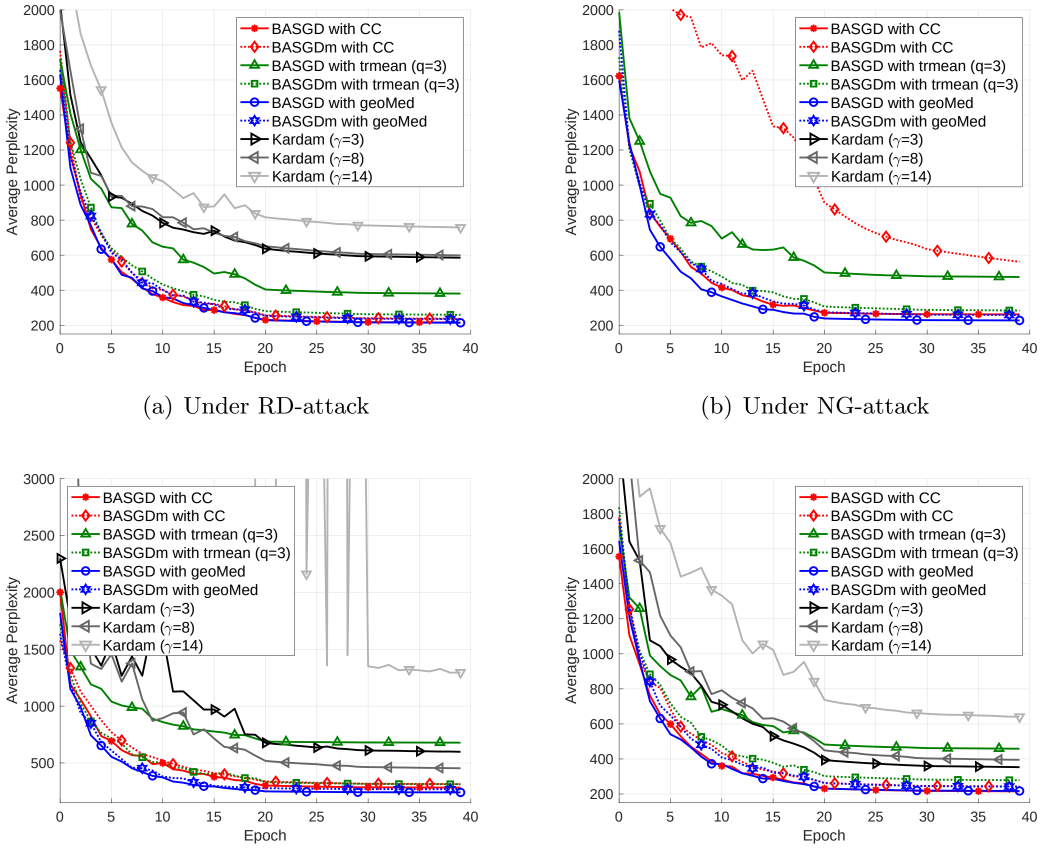
Figure 7: Average top-1 test accuracy w.r.t. wall-clock time when there are 3 Byzantine workers under omniscient attacks. In synchronous SGD with momentum (SSGDm) and BASGDm, the aggregation rules are set to be trmean. B is set to 10 for BASGDm.

reassignment. Meanwhile, BASGDm performs stably and outperforms SSGDm and Kardam when Δ ranges from 0.1s to 10s. The empirical results about hyper-parameter Δ is consistent with our discussion in Section 3.

5.3 Natural Language Processing Experiment

In this part, we will empirically compare the methods on natural language processing (NLP) tasks. In our NLP experiment, the methods are evaluated on the WikiText-2 dataset with an LSTM (Hochreiter and Schmidhuber, 1997) network. We only use the training set and test set, while the validation set is not used in our experiment. For LSTM, we adopt 2 layers with 100 units in each layer. Word embedding size is set to 100, and the sequence length is set to 35. Gradient clipping size is set to 0.25. Cross-entropy is used as the loss function. We run each algorithm for 40 epochs. Initial learning rate η is chosen from $\{1, 2, 5, 10, 20\}$ and is divided by 4 at the 21-st epoch and the 31-st epoch. The best result is adopted as the final one. k_{del} is randomly sampled from a standard exponential distribution. Similarly, each method is evaluated under RD-attack, NG-attack, FoE attack, and ALIE attack. The average perplexity is reported in Figure 8.

As illustrated in Figure 8(a) and Figure 8(b), under the two types of non-omniscient attacks (RD-attack and NG-attack), BASGD (BASGDm) can outperform Kardam, no matter which of the three aggregation rules is used. Moreover, the curves representing Kardam do not appear in Figure 8(b) because Kardam diverges under NG-attack and the perplexity explodes. We would also like to clarify that the performance of CC can get further improved by tuning the clipping size hyper-parameter more finely in different settings. However, it requires much computing cost and is beyond the scope of this work. Therefore, we fix clipping size $R = 0.5$, and this can already make BASGDm with CC outperform



(a) Under RD-attack

(b) Under NG-attack

(c) Under FoE attack

(d) Under ALIE attack

Figure 8: Average perplexity w.r.t. epochs (3 Byzantine workers, $B = 15$ for BASGD and BASGDm). In subfigure (b), the curves representing Kardam do not appear because Kardam diverges in this case and the average perplexity explodes.

Kardam. Theoretically, the best performance of CC can not be worse than geoMed since CC is equivalent to geoMed when the clipping size R is small enough (please see Appendix B.10 for the proof).

As illustrated in Figure 8(c) and Figure 8(d), under FoE attack and ALIE attack, BASGD can outperform Kardam except for the case of using trmean as aggregation rule. BASGD with trmean performs slightly worse than Kardam. A possible reason is that trmean is sensitive to model dimensions. On the contrary, by using momentum, BASGDm with any aggregation rule can always outperform Kardam.

The experimental results in this section have shown that BASGD (BASGDm) can outperform asynchronous Byzantine learning baselines under different settings. Moreover, BASGD (BASGDm) is compatible with various aggregation rules, such as trmean, geoMed, and CC. With the benefit of local momentum, BASGDm gets even stronger Byzantine resilience than BASGD, especially under the omniscient attacks FoE and ALIE.

6. Conclusion

In this paper, we propose a novel method called BASGD. To the best of our knowledge, BASGD is the first ABL method that can resist non-omniscient attacks without storing any instances on the server. Compared with those methods which need to store instances on the server, BASGD has a wider scope of application. An improved variant of BASGD, called BASGD with momentum (BASGDm), is further proposed by introducing local momentum into BASGD. As far as we know, BASGDm is the first ABL method that can resist the two omniscient attacks 'Fall of Empires' and 'A Little is Enough'. Both BASGD and BASGDm are compatible with various aggregation rules. Moreover, both BASGD and BASGDm are proved to be convergent and able to resist failure or attack. Empirical results show that our methods significantly outperform existing ABL baselines when there exists failure or attack on workers. Furthermore, both the theoretical results and the empirical results show the advantages of using local momentum in BASGDm.

Acknowledgments

This work is supported by National Key R&D Program of China (No. 2020YFA0713900), NSFC Project (No. 61921006, No. 62192783), and Fundamental Research Funds for the Central Universities (No. 020214380108).

Appendix A. Asynchronous SGD (ASGD)

One popular asynchronous method to solve the problem in (1) under the PS framework is ASGD (Dean et al., 2012), which is presented in Algorithm 3.

Algorithm 3 Asynchronous SGD (ASGD)

```
Server:
Initialization: initial parameter $\mathbf{w}^0$, learning rate $\eta$;
Send initial $\mathbf{w}^0$ to all workers;
for $t = 0$ to $t_{max} - 1$ do
  Wait until a new gradient $\mathbf{g}_k^t$ is received from arbitrary worker_$k$;
  Execute SGD step: $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta \cdot \mathbf{g}_k^t$;
  Send $\mathbf{w}^{t+1}$ back to worker_$k$;
end for
Notify all workers to stop;

Worker_$k$:   $(k = 0, 1, ..., m - 1)$
repeat
  Wait until receiving the latest parameter $\mathbf{w}$ from server;
  Randomly sample an index $i$ from $\mathcal{D}_k$ and compute $\nabla f(\mathbf{w}; z_i)$;
  Send $\nabla f(\mathbf{w}; z_i)$ to server;
until receive server's notification to stop
```

Appendix B. Proof Details

B.1 Proof of Proposition 5

Proof Firstly, we prove that coordinate-wise q-trimmed-mean is q-BR. It is not hard to check that trmean satisfies the property (a) in the definition of q-BR, then we prove that it also satisfies property (b). Without loss of generality, we assume $h_{1j}, \ldots, h_{Bj}$ are already in descending order. By definition, $Trm(h_{\cdot j})$ is the average value of $\mathcal{M}_j$, which is obtained by removing q largest values and q smallest values of $\{h_{ij}\}_{i=1}^B$. Therefore,

$$h_{(q+1)j} = \max_{x \in \mathcal{M}_j} \{x\} \geq Trm(h_{\cdot j}) \geq \min_{x \in \mathcal{M}_j} \{x\} = h_{(n-q)j}.$$

For any $\mathcal{S} \subset \{0, \ldots, B-1\}$ with $|\mathcal{S}| = B - q$, by Pigeonhole Principle, $\mathcal{S}$ includes at least one of $h_{1j}, \ldots, h_{(q+1)j}$, and includes at least one of $h_{(n-q)j}, \ldots, h_{Bj}$. Therefore,

$$\max_{s \in \mathcal{S}} \{h_{sj}\} \geq h_{(q+1)j}; \qquad \min_{s \in \mathcal{S}} \{h_{sj}\} \leq h_{(n-q)j}.$$

Combining these two inequalities, we have:

$$\max_{s \in \mathcal{S}} \{h_{sj}\} \geq Trm(h_{\cdot j}) \geq \min_{s \in \mathcal{S}} \{h_{sj}\}.$$

Thus, coordinate-wise q-trimmed-mean is q-BR. By definition, coordinate-wise median can be seen as $\lfloor \frac{B-1}{2} \rfloor$-trimmed-mean, and thus is $\lfloor \frac{B-1}{2} \rfloor$-BR. ■

B.2 Proof of Lemma 9

To begin with, we will introduce a lemma to estimate the ordered statistics.

Lemma 17. *$X_1, \dots, X_M$ are non-negative, independent and identically distributed (i.i.d.) random variables sampled from distribution $\mathcal{D}$, and have limited expectation $\mathbb{E}[X]$. Denote the K-th largest value in $\{X_1, \dots, X_M\}$ as $X_{(K)}$, then $\mathbb{E}[X_{(K)}] \le C_{M,K} \cdot \mathbb{E}[X]$, where*

$$C_{M,K} = \begin{cases} M, & K = 1; \\ \frac{M!(K-1)^{K-1}(M-K)^{M-K}}{(K-1)!(M-K)!(M-1)^{M-1}}, & 1 < K < \frac{M}{2}. \end{cases}$$

Proof Denote the Probability Density Function (PDF) and Cumulative Density Function (CDF) of $\mathcal{D}$ as $p(x)$ and $P(x)$, respectively. Then the PDF of $X_{(K)}$ is:

$$p_{(K)}(x) = \frac{M!}{(K-1)!(M-K)!}[1-P(x)]^{K-1}P(x)^{M-K}p(x).$$

Thus,

$$\begin{aligned}
\mathbb{E}[X_{(K)}] =& \int_0^{+\infty} x \cdot p_{(K)}(x)dx \\
=& \int_0^{+\infty} \left[\frac{M!}{(K-1)!(M-K)!} \cdot [1-P(x)]^{K-1}P(x)^{M-K}\right] \cdot xp(x)dx \\
\overset{(a)}{\le}& \int_0^{+\infty} \left[\frac{M!}{(K-1)!(M-K)!} \cdot \frac{(K-1)^{K-1}(M-K)^{M-K}}{(M-1)^{M-1}}\right] \cdot xp(x)dx \\
=& \frac{M!(K-1)^{K-1}(M-K)^{M-K}}{(K-1)!(M-K)!(M-1)^{M-1}} \cdot \mathbb{E}[X].
\end{aligned}$$

Inequality (a) is derived based on $[1-P(x)]^{K-1}P(x)^{M-K} \le \frac{(K-1)^{K-1}(M-K)^{M-K}}{(M-1)^{M-1}}$, which is obtained by the following process:
Let $\theta(x) = (1-x)^{K-1}x^{M-K},\ x \in [0,1]$.
Then $\theta'(x) = (1-x)^{K-2}x^{M-K-1}[(M-K)(1-x) - (K-1)x]$.
Let $\theta'(x) = 0$. Solving the equation, we obtain $x = \frac{M-K}{M-1}$, 0 or 1.
Also, we have $\theta(0) = \theta(1) = 0$, and $\theta(\frac{M-K}{M-1}) = \frac{(K-1)^{K-1}(M-K)^{M-K}}{(M-1)^{M-1}}$.
Then we have $\max_{x\in[0,1]} \theta(x) = \theta(\frac{M-K}{M-1}) = \frac{(K-1)^{K-1}(M-K)^{M-K}}{(M-1)^{M-1}}$.
Thus, $[1-P(x)]^{K-1}P(x)^{M-K} = \theta(P(x)) \le \frac{(K-1)^{K-1}(M-K)^{M-K}}{(M-1)^{M-1}}$. ■

Proposition 18. *$\forall B, q, r \in \mathbb{Z}_+, 0 \le r \le q < \frac{B}{2}$,*

$$C_{B-r,q-r+1} \le \Theta_{B,q,r} = \frac{(B-r)\sqrt{B-r+1}}{\sqrt{(B-q-1)(q-r+1)}}.$$

Proof By Stirling's approximation, we have:

$$\sqrt{2\pi n} \cdot n^n e^{-n} \le n! \le e\sqrt{n} \cdot n^n e^{-n},\ \forall n \in \mathbb{Z}_+.$$

Therefore,

$$\sqrt{2\pi n} \cdot e^{-n} \leq \frac{n!}{n^n} \leq e\sqrt{n} \cdot e^{-n}, \ \forall n \in \mathbb{Z}_+. \tag{5}$$

By definition of $C_{M,k}$,

$$\begin{aligned} C_{M,K} =& \frac{M!(K-1)^{K-1}(M-K)^{M-K}}{(K-1)!(M-K)!(M-1)^{M-1}} \\ =& M \cdot \frac{(M-1)!}{(M-1)^{M-1}} \cdot \frac{(K-1)^{K-1}}{(K-1)!} \cdot \frac{(M-K)^{M-K}}{(M-K)!} \\ \leq& M \cdot [e\sqrt{M-1} \cdot e^{-(M-1)}] \cdot \frac{e^{K-1}}{\sqrt{2\pi(K-1)}} \cdot \frac{e^{M-K}}{\sqrt{2\pi(M-K)}} \\ =& \frac{e}{2\pi} \cdot \frac{M\sqrt{M-1}}{\sqrt{(M-K)(K-1)}}, \end{aligned}$$

where the inequality uses Inequality (5).
Case (i). When $r < q$,

$$\begin{aligned} C_{B-r,q-r+1} \leq& \frac{e}{2\pi} \cdot \frac{(B-r)\sqrt{B-r-1}}{\sqrt{(B-q-1)(q-r)}} \\ \leq& \frac{(B-r)\sqrt{B-r+1}}{\sqrt{(B-q-1)(q-r+1)}}. \end{aligned}$$

Case (ii). When $r = q$, by definition of $C_{M,K}$, we have:

$$C_{B-r,q-r+1} = C_{B-q,1} = B - q = \frac{(B-r)\sqrt{B-r+1}}{\sqrt{(B-q-1)(q-r+1)}}.$$

In conclusion, when $r \leq q$, we have:

$$C_{B-r,q-r+1} \leq \frac{(B-r)\sqrt{B-r+1}}{\sqrt{(B-q-1)(q-r+1)}}.$$

■

When B and q are fixed, the upper bound of $C_{B-r,q-r+1}$ will increase when r (the number of Byzantine workers) increases. Namely, the upper bound will be larger if there are more Byzantine workers. When B and r are fixed, q measures the Byzantine Robust degree of aggregation function $Aggr(\cdot)$. The factor $[(B-q-1)(q-r)]^{-\frac{1}{2}}$ is monotonically decreasing with respect to q, when $q < \frac{B-1+r}{2}$. Since $r \leq q < \frac{B}{2}$, the upper bound will decrease when q increases. Also, $B - q$ decreases when q increases. Namely, the upper bound will be smaller if $Aggr(\cdot)$ has a stronger q-BR property.

In the worst case ($q = r$), the upper bound of $C_{B-r,q-r+1}$ is linear to B. Even in the best case ($r = 0, q = \lfloor \frac{B-1}{2} \rfloor$), the denominator is about $\frac{B}{2}$ and the upper bound of $C_{B-r,q-r+1}$ is linear to $\sqrt{B}$. Thus, larger B might result in larger error. Hence, the buffer number is not supposed to be set too large.

Now we prove Lemma 9.

Proof

$$\begin{aligned}
&\mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t]\\
=&\mathbb{E}[||Aggr([\mathbf{h}_0,\ldots,\mathbf{h}_{B-1}])||^2 \mid \mathbf{w}^t]\\
=&\sum_{j=1}^{d}\mathbb{E}[Aggr([\mathbf{h}_0,\ldots,\mathbf{h}_{B-1}])_j^2 \mid \mathbf{w}^t],
\end{aligned}$$

where $Aggr([\mathbf{h}_0,\ldots,\mathbf{h}_{B-1}])_j$ represents the j-th coordinate of the aggregated gradient.

We use $\mathcal{H}^t$ to denote the credible buffer index set, which is composed of the index of buffers, where the stored gradients are all from loyal workers.

For each $b \in \mathcal{H}^t$, $\mathbf{h}_b$ has stored N_b^t gradients at iteration t: $\mathbf{g}_1,\ldots,\mathbf{g}_{N_b^t}$, and we have:

$$\mathbf{h}_b = \frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\mathbf{g}_l.$$

Then,

$$\begin{aligned}
\mathbb{E}[\|\mathbf{h}_b\|^2|\mathbf{w}^t] =&\mathbb{E}[\|\mathbf{h}_b - \mathbb{E}[\mathbf{h}_b|\mathbf{w}^t]\|^2|\mathbf{w}^t] + \|\mathbb{E}[\mathbf{h}_b|\mathbf{w}^t]\|^2\\
=&\mathbb{E}\left[\left\|\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}(\mathbf{g}_l - \mathbb{E}[\mathbf{g}_l|\mathbf{w}^t])\right\|^2\middle|\mathbf{w}^t\right] + \left\|\mathbb{E}\left[\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\mathbf{g}_l\middle|\mathbf{w}^t\right]\right\|^2\\
\overset{(a)}{\le}&\frac{\sigma^2}{N_b^t} + \left\|\mathbb{E}\left[\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\mathbf{g}_l\middle|\mathbf{w}^t\right]\right\|^2\\
=&\frac{\sigma^2}{N_b^t} + \frac{1}{(N_b^t)^2}\left\|\sum_{l=1}^{N_b^t}\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\right\|^2\\
\overset{(b)}{\le}&\frac{\sigma^2}{N_b^t} + \frac{1}{(N_b^t)^2}\cdot N_b^t\cdot\sum_{l=1}^{N_b^t}\|\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\|^2\\
\overset{(c)}{\le}&\frac{\sigma^2}{N_b^t} + D^2.
\end{aligned}$$

Inequality (a) is derived based on Assumption 4 and the fact that $\mathbf{g}_i$ is mutually uncorrelated. Inequality (b) is derived by the following process:

$$\begin{aligned}
\left\|\sum_{l=1}^{N_b^t}\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\right\|^2 =&\sum_{l=1}^{N_b^t}\|\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\|^2 + \sum_{1\le l<l'\le N_b^t}2\cdot\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]^T\mathbb{E}[\mathbf{g}_l'|\mathbf{w}^t]\\
\le&\sum_{l=1}^{N_b^t}\|\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\|^2 + \sum_{1\le l<l'\le N_b^t}(\|\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\|^2 + \|\mathbb{E}[\mathbf{g}_l'|\mathbf{w}^t]\|^2)
\end{aligned}$$

$$
\begin{aligned}
=&\sum_{l=1}^{N_b^t} \|\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\|^2 + (N_b^t - 1)\cdot \sum_{l=1}^{N_b^t} \|\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\|^2 \\
=&N_b^t \cdot \sum_{i=l}^{N_b^t} \|\mathbb{E}[\mathbf{g}_l|\mathbf{w}^t]\|^2.
\end{aligned}
$$

Inequality (c) is derived based on Assumption 3.

Because there are no more than r Byzantine workers at iteration t, no more than r buffers contain Byzantine gradient. Thus, the credible buffer index set $\mathcal{H}^t$ has at least $(B-r)$ elements. In case that $\mathcal{H}^t$ has more than $(B-r)$ elements, we take the indices of the smallest $(B-q)$ elements in $\{h_{bj}\}_{b\in\mathcal{H}^t}$ to compose $\mathcal{H}_j^t$, and we have $|\mathcal{H}_j^t| = B-q$.

Note that $Aggr(\cdot)$ is q-BR, and by definition we have:

$$
\min_{b\in\mathcal{H}_j^t}\{h_{bj}\} \leq Aggr([\mathbf{h}_0,\ldots,\mathbf{h}_{B-1}])_j \leq \max_{b\in\mathcal{H}_j^t}\{h_{bj}\}.
$$

Therefore,

$$
\sum_{j=1}^{d}\mathbb{E}[Aggr([\mathbf{h}_0,\ldots,\mathbf{h}_{B-1}])_j^2|\mathbf{w}^t] \leq \sum_{j=1}^{d}\mathbb{E}[\max_{b\in\mathcal{H}_j^t}\{h_{bj}^2\}|\mathbf{w}^t].
$$

There are $(B-r)$ credible buffers, and we choose the smallest $(B-q)$ buffers to compose $\mathcal{H}_j^t$. Therefore, for all $b\in\mathcal{H}_j^t$, h_{bj} is not larger than the $(q-r+1)$-th largest one in $\{h_{bj}\}_{b\in\mathcal{H}^t}$. Let $N^{(t)}$ be the $(q+1)$-th smallest value in $\{N_b^t\}_{b\in\{0,\ldots,B-1\}}$. Using Lemma 17, we have:

$$
\begin{aligned}
\mathbb{E}[\max_{b\in\mathcal{H}_j^t}\{h_{bj}^2\}|\mathbf{w}^t] \leq&\mathbb{E}[\max_{b\in\mathcal{H}_j^t}\{\|\mathbf{h}_b\|^2\}|\mathbf{w}^t] \\
\leq&\mathbb{E}[\max_{b\in\mathcal{H}_j^t}\{D^2 + \frac{\sigma^2}{N_b^t}\}|\mathbf{w}^t] \\
=&C_{B-r,q-r+1}\cdot(D^2 + \frac{\sigma^2}{N^{(t)}}).
\end{aligned}
$$

Thus,

$$
\mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \ \leq\ \sum_{j=1}^{d}\mathbb{E}[\max_{b\in\mathcal{H}_j^t}\{h_{bj}^2\}|\mathbf{w}^t] \ \leq C_{B-r,q-r+1}d\cdot(D^2 + \frac{\sigma^2}{N^{(t)}}).
$$

By Proposition 18, we have:

$$
\mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \leq d\cdot\frac{(B-r)\sqrt{B-r+1}}{\sqrt{(B-q-1)(q-r+1)}}\cdot(D^2 + \frac{\sigma^2}{N^{(t)}}).
$$

■

B.3 Proof of Lemma 10

Proof

$$\begin{aligned}
&\mathbb{E}[\mathbf{G}^t - \nabla F(\mathbf{w}^t) \mid \mathbf{w}^t] \\
=&\mathbb{E}[Aggr([\mathbf{h}_0, \dots, \mathbf{h}_{B-1}]) - \nabla F(\mathbf{w}^t) \mid \mathbf{w}^t] \\
=&\mathbb{E}[Aggr([\mathbf{h}_0 - \nabla F(\mathbf{w}^t), \dots, \mathbf{h}_{B-1} - \nabla F(\mathbf{w}^t)]) \mid \mathbf{w}^t], \qquad (6)
\end{aligned}$$

where the second equation is derived based on Property (b) in the definition of q-BR.

For each $b \in \mathcal{H}^t$, $\mathbf{h}_b$ has stored N_b^t gradients at iteration t: $\mathbf{g}_1, \dots, \mathbf{g}_{N_b^t}$, and we have:

$$\mathbf{h}_b - \nabla F(\mathbf{w}^t) = \frac{1}{N_b^t}\sum_{k=1}^{N_b^t} \mathbf{g}_i - \nabla F(\mathbf{w}^t) = \frac{1}{N_b^t}\sum_{k=1}^{N_b^t} [\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \nabla F(\mathbf{w}^t)],$$

where $0 \le t - t_k \le \tau_{max}$, $\forall k = 1, 2, \dots, N_b^t$.

Taking expectations on both sides, we have:

$$\begin{aligned}
&\mathbb{E}[||\mathbf{h}_b - \nabla F(\mathbf{w}^t)|| \; |\mathbf{w}^t] \\
=&\mathbb{E}[||\frac{1}{N_b^t}\sum_{k=1}^{N_b^t}(\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \nabla F(\mathbf{w}^t))|| \; |\mathbf{w}^t] \\
\le&\frac{1}{N_b^t}\sum_{k=1}^{N_b^t}\mathbb{E}[||\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \nabla F(\mathbf{w}^t)|| \; |\mathbf{w}^t] \\
\overset{(a)}{\le}&\frac{1}{N_b^t}\sum_{k=1}^{N_b^t}\{\mathbb{E}[||\nabla F(\mathbf{w}^{t_k}) - \nabla F(\mathbf{w}^t)|| \; |\mathbf{w}^t] \\
&\qquad + \mathbb{E}[||\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \mathbb{E}[\nabla f(\mathbf{w}^{t_k}; z_{i_k})]|| \; |\mathbf{w}^t] \\
&\qquad + \mathbb{E}[||\mathbb{E}[\nabla f(\mathbf{w}^{t_k}; z_{i_k})] - \nabla F(\mathbf{w}^{t_k})|| \; |\mathbf{w}^t]\},
\end{aligned}$$

where (a) is derived based on Triangle Inequality.

The first part:

$$\begin{aligned}
&\mathbb{E}[||\nabla F(\mathbf{w}^{t_k}) - \nabla F(\mathbf{w}^t)|| \; |\mathbf{w}^t] \\
\overset{(b)}{\le}&L \cdot \mathbb{E}[||\mathbf{w}^{t_k} - \mathbf{w}^t|| \; |\mathbf{w}^t] \\
=&L \cdot \mathbb{E}[||\sum_{t'=t_k}^{t-1} \mathbf{G}^{t'}|| \; |\mathbf{w}^t] \\
\le&\sum_{t'=t_k}^{t-1} L \cdot \mathbb{E}[||\mathbf{G}^{t'}|| \; |\mathbf{w}^t] \\
=&\sum_{t'=t_k}^{t-1} L \cdot \sqrt{\mathbb{E}[||\mathbf{G}^{t'}|| \; |\mathbf{w}^t]^2}
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{t'=t_k}^{t-1} L \cdot \sqrt{\mathbb{E}[||\mathbf{G}^{t'}||^2 \ |\mathbf{w}^t]} \\
&\overset{(c)}{\leq} \sum_{t'=t_k}^{t-1} L \cdot \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})} \\
&\overset{(d)}{\leq} \tau_{max} L \cdot \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})},
\end{aligned}
$$

where (b) is derived based on Assumption 5, (c) is derived based on Lemma 9, and (d) is derived based on $t - t_k \leq \tau_{max}$.

The second part:

$$
\begin{aligned}
&\mathbb{E}[||\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \mathbb{E}[\nabla f(\mathbf{w}^{t_k}; z_{i_k})]|| \ |\mathbf{w}^t] \\
=&\sqrt{\mathbb{E}[||\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \mathbb{E}[\nabla f(\mathbf{w}^{t_k}; z_{i_k})]|| \ |\mathbf{w}^t]^2} \\
\leq&\sqrt{\mathbb{E}[||\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \mathbb{E}[\nabla f(\mathbf{w}^{t_k}; z_{i_k})]||^2 \ |\mathbf{w}^t]} \\
\overset{(e)}{\leq}&\sigma,
\end{aligned}
$$

where (e) is derived based on Assumption 4.

By Assumption 2, we have the following estimation for the third part:

$$
\mathbb{E}[||\mathbb{E}[\nabla f(\mathbf{w}^{t_k}; z_{i_k})] - \nabla F(\mathbf{w}^{t_k})|| \ |\mathbf{w}^t] \leq \kappa.
$$

Therefore,

$$
\begin{aligned}
&\mathbb{E}[||\mathbf{h}_b - \nabla F(\mathbf{w}^t)|| \ |\mathbf{w}^t] \\
\leq&\frac{1}{N_b^t} \sum_{k=1}^{N_b^t} (\tau_{max} L \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})} + \sigma + \kappa) \\
=&\tau_{max} L \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})} + \sigma + \kappa. \qquad (7)
\end{aligned}
$$

Similar to the proof of Lemma 9, $\forall j \in [d]$, we have:

$$
\begin{aligned}
&\min_{b \in \mathcal{H}_j^t} \{h_{bj} - \nabla F(\mathbf{w}^t)_j\} \\
\leq&Aggr([\mathbf{h}_0 - \nabla F(\mathbf{w}^t), \ldots, \mathbf{h}_{B-1} - \nabla F(\mathbf{w}^t)])_j \\
\leq&\max_{b \in \mathcal{H}_j^t} \{h_{bj} - \nabla F(\mathbf{w}^t)_j\},
\end{aligned}
$$

where $\mathcal{H}_j^t$ is composed by the indices of the smallest $(B-q)$ elements in $\{h_{bj} - \nabla F(\mathbf{w}^t)_j\}_{b \in \mathcal{H}^t}$. Therefore,

$$
\begin{aligned}
&||\mathbb{E}[Aggr([\mathbf{h}_0 - \nabla F(\mathbf{w}^t), \ldots, \mathbf{h}_{B-1} - \nabla F(\mathbf{w}^t)]) \mid \mathbf{w}^t]|| \\
\leq&\sum_{j=1}^{d} ||\mathbb{E}[Aggr([\mathbf{h}_0 - \nabla F(\mathbf{w}^t), \ldots, \mathbf{h}_{B-1} - \nabla F(\mathbf{w}^t)])_j \mid \mathbf{w}^t]||
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{j=1}^{d} \mathbb{E}[||Aggr([\mathbf{h}_0 - \nabla F(\mathbf{w}^t), \dots, \mathbf{h}_{B-1} - \nabla F(\mathbf{w}^t)])_j|| \mid \mathbf{w}^t] \\
&\overset{(f)}{\leq} \sum_{j=1}^{d} \mathbb{E}[\max_{b\in\mathcal{H}_j^t} ||h_{bj} - \nabla F(\mathbf{w}^t)_j|| \mid \mathbf{w}^t] \\
&\overset{(g)}{\leq} \sum_{j=1}^{d} C_{B-r,q-r+1}\mathbb{E}[||h_{bj} - \nabla F(\mathbf{w}^t)_j|| \, |\mathbf{w}^t] \\
&\leq \sum_{j=1}^{d} C_{B-r,q-r+1}\mathbb{E}[||\mathbf{h}_b - \nabla F(\mathbf{w}^t)|| \, |\mathbf{w}^t] \\
&\overset{(h)}{\leq} \sum_{j=1}^{d} C_{B-r,q-r+1} \cdot \left(\tau_{max} L \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})} + \sigma + \kappa\right) \\
&= C_{B-r,q-r+1} d \cdot \left(\tau_{max} L \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})} + \sigma + \kappa\right), \qquad (8)
\end{aligned}
$$

where (f) is derived based on definition of q-BR, (g) is derived based on Lemma 17, and (h) is derived based on Inequality (7). Combining Equation (6) and Inequality (8), we obtain:

$$
||\mathbb{E}[\mathbf{G}^t - \nabla F(\mathbf{w}^t) \mid \mathbf{w}^t]|| \leq C_{B-r,q-r+1} d \cdot \left(\tau_{max} L \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})} + \sigma + \kappa\right).
$$

By Proposition (18), we have:

$$
\begin{aligned}
||\mathbb{E}[\mathbf{G}^t - \nabla F(\mathbf{w}^t) \mid \mathbf{w}^t]|| \leq & \frac{d(B-r)\sqrt{B-r+1}}{\sqrt{(B-q-1)(q-r+1)}} \\
& \cdot \left(\tau_{max} L \sqrt{d \frac{(B-r)\sqrt{B-r+1}}{\sqrt{(B-q-1)(q-r+1)}} \cdot (D^2 + \sigma^2/N^{(t)})} + \sigma + \kappa\right).
\end{aligned}
$$

■

B.4 Proof of Theorem 11

Proof

$$
\begin{aligned}
\mathbb{E}[F(\mathbf{w}^{t+1}) \mid \mathbf{w}^t] =& \mathbb{E}[F(\mathbf{w}^t - \eta \cdot \mathbf{G}^t) \mid \mathbf{w}^t] \\
\overset{(a)}{\leq} & \mathbb{E}[F(\mathbf{w}^t) - \eta \cdot \nabla F(\mathbf{w}^t)^T \mathbf{G}^t + \frac{L}{2}\eta^2 ||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \\
=& F(\mathbf{w}^t) - \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}^t \mid \mathbf{w}^t] + \frac{\eta^2 L}{2} \mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \\
=& F(\mathbf{w}^t) - \eta \cdot \nabla F(\mathbf{w}^t)^T \mathbb{E}[\mathbf{G}^t \mid \mathbf{w}^t] + \frac{\eta^2 L}{2} \mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \\
=& F(\mathbf{w}^t) - \eta \cdot \nabla F(\mathbf{w}^t)^T \nabla F(\mathbf{w}^t) + \frac{\eta^2 L}{2} \mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t]
\end{aligned}
$$

$$
\begin{aligned}
&- \eta \cdot \nabla F(\mathbf{w}^t)^T \mathbb{E}[\mathbf{G}^t - \nabla F(\mathbf{w}^t) \mid \mathbf{w}^t] \\
\leq & F(\mathbf{w}^t) - \eta \cdot ||\nabla F(\mathbf{w}^t)||^2 + \frac{\eta^2 L}{2}\mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \\
&+ \eta \cdot ||\nabla F(\mathbf{w}^t)|| \cdot ||\mathbb{E}[\mathbf{G}^t - \nabla F(\mathbf{w}^t) \mid \mathbf{w}^t]||,
\end{aligned}
$$

where (a) is derived based on Assumption 5. Using Lemma 9 and Lemma 10, we have:

$$
\begin{aligned}
&\mathbb{E}[F(\mathbf{w}^{t+1}) \mid \mathbf{w}^t] \\
\leq & F(\mathbf{w}^t) - \eta \cdot ||\nabla F(\mathbf{w}^t)||^2 + \frac{\eta^2 L}{2} C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)}) \\
&+ \eta \cdot C_{B-r,q-r+1} d \cdot (\tau_{max} L \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})} + \sigma + \kappa) \cdot ||\nabla F(\mathbf{w}^t)||.
\end{aligned}
$$

Also, by Assumption 3, $||\nabla F(\mathbf{w}^t)|| \leq D$. Taking total expectation and using that $||\nabla F(\mathbf{w}^t)|| \leq D$, we have:

$$
\begin{aligned}
\mathbb{E}[F(\mathbf{w}^{t+1})] \leq & \mathbb{E}[F(\mathbf{w}^t)] - \eta \cdot \mathbb{E}[||\nabla F(\mathbf{w}^t)||^2] + \frac{\eta^2 L}{2} C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)}) \\
&+ \eta \cdot C_{B-r,q-r+1} D d (\tau_{max} L \sqrt{C_{B-r,q-r+1} d \cdot (D^2 + \sigma^2/N^{(t)})} + \sigma + \kappa).
\end{aligned}
$$

Let $\tilde{D} = \frac{1}{T}\sum_{t=0}^{T-1} \sqrt{D^2 + \sigma^2/N^{(t)}}$. By telescoping, we have:

$$
\begin{aligned}
\eta \cdot \sum_{t=0}^{T-1} \mathbb{E}[||\nabla F(\mathbf{w}^t)||^2] \leq & \{F(\mathbf{w}^0) - \mathbb{E}[F(\mathbf{w}^T)]\} + \eta^2 T \cdot \frac{L}{2} C_{B-r,q-r+1} d \cdot \frac{1}{T}\sum_{t=0}^{T-1}(D^2 + \sigma^2/N^{(t)}) \\
&+ \eta T \cdot C_{B-r,q-r+1} D d (\tau_{max} L \tilde{D} \sqrt{C_{B-r,q-r+1} d} + \sigma + \kappa).
\end{aligned}
$$

Note that $\mathbb{E}[F(\mathbf{w}^T)] \geq F^*$, and let $\eta = O\left(\frac{1}{L\sqrt{T}}\right)$:

$$
\begin{aligned}
\frac{\sum_{t=0}^{T-1} \mathbb{E}[||\nabla F(\mathbf{w}^t)||^2]}{T} \leq & O\left(\frac{L[F(\mathbf{w}^0) - F^*]}{\sqrt{T}}\right) + O\left(\frac{C_{B-r,q-r+1}\tilde{D} d}{\sqrt{T}}\right) \\
&+ O\left(C_{B-r,q-r+1} D d \cdot (\tau_{max} L \tilde{D} \sqrt{C_{B-r,q-r+1} d} + \sigma + \kappa)\right).
\end{aligned}
$$

Let $\delta_{\max} = q/B$. When $q = r$, we have

$$
\begin{aligned}
C_{B-r,q-r+1} \leq & \frac{(q/\delta_{\max} - r)\sqrt{q/\delta_{\max} - r + 1}}{\sqrt{(q/\delta_{\max} - q - 1)(q - r + 1)}} \\
= & \sqrt{\frac{r/\delta_{\max} - r + 1}{r/\delta_{\max} - r - 1}} \cdot \frac{(1 - \delta_{\max}) r}{\delta_{\max}} \leq \frac{2(1 - \delta_{\max}) r}{\delta_{\max}}.
\end{aligned}
$$

Thus,

$$
\frac{\sum_{t=0}^{T-1} \mathbb{E}[||\nabla F(\mathbf{w}^t)||^2]}{T} \leq O\left(\frac{L[F(\mathbf{w}^0) - F^*]}{T^{\frac{1}{2}}}\right) + O\left(\frac{2(1 - \delta_{\max}) r d \tilde{D}}{\delta_{\max} T^{\frac{1}{2}}}\right)
$$

$$+O\left(\frac{2(1-\delta_{\max})rDd\sigma}{\delta_{\max}}+\frac{2(1-\delta_{\max})rDd\kappa}{\delta_{\max}}+\frac{2\sqrt{2}(1-\delta_{\max})^{\frac{3}{2}}r^{\frac{3}{2}}LD\tilde{D}d^{\frac{3}{2}}\tau_{max}}{(\delta_{\max})^{\frac{3}{2}}}\right).$$

■

B.5 Proof of Theorem 12

Proof Let $\mathbf{h}'_b$ be the value of the b-th buffer, if all received loyal gradients were computed based on $\mathbf{w}^t$. Since $\mathbf{G}^t = Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}])$, we have:

$$\begin{aligned}
&\mathbb{E}[F(\mathbf{w}^{t+1}) \mid \mathbf{w}^t] \\
=&\mathbb{E}[F(\mathbf{w}^t - \eta \cdot \mathbf{G}^t) \mid \mathbf{w}^t] \\
\overset{(a)}{\le}&\mathbb{E}[F(\mathbf{w}^t) - \eta \cdot \nabla F(\mathbf{w}^t)^T\mathbf{G}^t + \frac{L}{2}\eta^2||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \\
=&F(\mathbf{w}^t) - \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T\mathbf{G}^t \mid \mathbf{w}^t] + \frac{\eta^2 L}{2}\mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t],
\end{aligned} \tag{9}$$

where (a) is derived based on Assumption 5.

Firstly, we estimate the value of $\mathbb{E}[\nabla F(\mathbf{w}^t)^T\mathbf{G}^t \mid \mathbf{w}^t]$.

Since there are at most r Byzantine workers, at most r buffers may contain Byzantine gradients. Without loss of generality, suppose only the first r buffers may contain Byzantine gradients.

Let $\mathbf{G}^t_{syn} = Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{r-1}, \mathbf{h}'_r, \ldots, \mathbf{h}'_{B-1}])$, where $\mathbf{h}_0, \ldots, \mathbf{h}_{r-1}$ may contain Byzantine gradients and be arbitrary value, and $\mathbf{h}'_r, \ldots, \mathbf{h}'_{B-1}$ each stores loyal gradients computed based on $\mathbf{w}^t$. Thus,

$$\mathbb{E}[\nabla F(\mathbf{w}^t)^T\mathbf{G}^t_{syn} \mid \mathbf{w}^t] \ge \|\nabla F(\mathbf{w}^t)\|^2 - A_1, \tag{10}$$

$$\mathbb{E}[\|\mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] \le (A_2)^2. \tag{11}$$

Let $\alpha = 2\eta^2L^2\tau_{max}^2(B-r) < 1$.

We claim that

$$\mathbb{E}[\|\mathbf{G}^t - \mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] \le (\frac{1}{2}\alpha^{t+1} + \frac{\alpha}{1-\alpha}) \cdot (A_2)^2,$$

and

$$\mathbb{E}[\|\mathbf{G}^t\|^2 \mid \mathbf{w}^t] \le (\alpha^{t+1} + \frac{2}{1-\alpha}) \cdot (A_2)^2.$$

Now we prove it by induction on t.

Step 1. When $t=0$, all gradients are computed according to $\mathbf{w}^0$, and we have $\mathbf{G}^0 = \mathbf{G}^0_{syn}$. Thus,

$$\mathbb{E}[\|\mathbf{G}^0 - \mathbf{G}^0_{syn}\|^2 \mid \mathbf{w}^0] = 0 \le (\frac{1}{2}\alpha^1 + \frac{\alpha}{1-\alpha}) \cdot (A_2)^2,$$

$$\mathbb{E}[\|\mathbf{G}^0\|^2 \mid \mathbf{w}^0] = \mathbb{E}[\|\mathbf{G}^0_{syn}\|^2 \mid \mathbf{w}^0] \le (A_2)^2 \le (\alpha^1 + \frac{2}{1-\alpha}) \cdot (A_2)^2.$$

Step 2. If

$$\mathbb{E}[\|\mathbf{G}^{t'} - \mathbf{G}_{syn}^{t'}\|^2 \mid \mathbf{w}^{t'}] \leq (\frac{1}{2}\alpha^{t'+1} + \frac{\alpha}{1-\alpha}) \cdot (A_2)^2,$$
$$\mathbb{E}[\|\mathbf{G}^{t'}\|^2 \mid \mathbf{w}^{t'}] \leq (\alpha^{t'+1} + \frac{2}{1-\alpha}) \cdot (A_2)^2,$$

holds for all $t' = 0, 1, \ldots, t-1$ (induction hypothesis), then:

$$\begin{aligned}
&\mathbb{E}[\|\mathbf{G}^t - \mathbf{G}_{syn}^t\|^2 \mid \mathbf{w}^t] \\
=&\mathbb{E}[\|Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{r-1}, \mathbf{h}_r, \ldots, \mathbf{h}_{B-1}]) - Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{r-1}, \mathbf{h}'_r, \ldots, \mathbf{h}'_{B-1}])\|^2 \mid \mathbf{w}^t] \\
\overset{(b)}{\leq}&\mathbb{E}[\sum_{b=r}^{B-1} \|\mathbf{h}_b - \mathbf{h}'_b\|^2 \mid \mathbf{w}^t] \\
=&\sum_{b=r}^{B-1} \mathbb{E}[\|\frac{1}{N_b^t}\sum_{k=1}^{N_b^t}(\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \nabla f(\mathbf{w}^t; z_{i_k}))\|^2 \mid \mathbf{w}^t] \\
\overset{(c)}{\leq}&\sum_{b=r}^{B-1} \mathbb{E}[\frac{1}{N_b^t}\sum_{k=1}^{N_b^t}\|\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \nabla f(\mathbf{w}^t; z_{i_k})\|^2 \mid \mathbf{w}^t] \\
\overset{(d)}{\leq}&\sum_{b=r}^{B-1} \mathbb{E}[\frac{1}{N_b^t}\sum_{k=1}^{N_b^t} L^2\|\mathbf{w}^{t_k} - \mathbf{w}^t\|^2 \mid \mathbf{w}^t] \\
=&\sum_{b=r}^{B-1} \frac{L^2}{N_b^t}\sum_{k=1}^{N_b^t} \mathbb{E}[\|\mathbf{w}^{t_k} - \mathbf{w}^t\|^2 \mid \mathbf{w}^t] \\
=&\sum_{b=r}^{B-1} \frac{L^2}{N_b^t}\sum_{k=1}^{N_b^t} \mathbb{E}[\|\sum_{t'=t_k}^{t-1} \eta \cdot \mathbf{G}^{t'}\|^2 \mid \mathbf{w}^t] \\
\overset{(e)}{\leq}&\sum_{b=r}^{B-1} \frac{\eta^2 L^2}{N_b^t}\sum_{k=1}^{N_b^t} \mathbb{E}[(t-t_k)\sum_{t'=t_k}^{t-1} \|\mathbf{G}^{t'}\|^2 \mid \mathbf{w}^t] \\
\overset{(f)}{\leq}&\sum_{b=r}^{B-1} \frac{\eta^2 L^2}{N_b^t}\sum_{k=1}^{N_b^t} [(t-t_k)\sum_{t'=t_k}^{t-1} (\alpha^{t'+1} + \frac{2}{1-\alpha}) \cdot (A_2)^2] \\
\leq&\sum_{b=r}^{B-1} \frac{\eta^2 L^2}{N_b^t}\sum_{k=1}^{N_b^t} [(t-t_k)\sum_{t'=t_k}^{t-1} (\alpha^{t} + \frac{2}{1-\alpha}) \cdot (A_2)^2] \\
\overset{(g)}{\leq}&\sum_{b=r}^{B-1} (\eta^2 L^2 \tau_{max}^2) \cdot (\alpha^t + \frac{2}{1-\alpha}) \cdot (A_2)^2 \\
=&(\eta^2 L^2 (B-r) \tau_{max}^2) \cdot (\alpha^t + \frac{2}{1-\alpha}) \cdot (A_2)^2 \\
\overset{(h)}{=}&\frac{1}{2}\alpha \cdot (\alpha^t + \frac{2}{1-\alpha}) \cdot (A_2)^2 \\
=&(\frac{1}{2}\alpha^{t+1} + \frac{\alpha}{1-\alpha}) \cdot (A_2)^2,
\end{aligned} \tag{12}$$

where (b) is derived based on the definition of stable aggregation function, (c) is derived based on Cauchy's Inequality, (d) is derived based on Assumption 5, (e) is also derived based on Cauchy's Inequality, (f) is derived based on the induction hypothesis, (g) is derived based on that $t - t_k \leq \tau_{max}$, and (h) is derived based on that $\alpha = 2\eta^2 L^2 \tau_{max}^2 (B - r)$.

Therefore,

$$\begin{aligned}\mathbb{E}[\|\mathbf{G}^t\|^2 \mid \mathbf{w}^t] =& \mathbb{E}[||\mathbf{G}^t_{syn} + (\mathbf{G}^t - \mathbf{G}^t_{syn})||^2 \mid \mathbf{w}^t] \\ \overset{(i)}{\leq}& 2 \cdot \mathbb{E}[\|\mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] + 2 \cdot \mathbb{E}[||\mathbf{G}^t - \mathbf{G}^t_{syn}||^2 \mid \mathbf{w}^t] \\ \overset{(j)}{\leq}& 2 \cdot (A_2)^2 + 2 \cdot \mathbb{E}[||\mathbf{G}^t - \mathbf{G}^t_{syn}||^2 \mid \mathbf{w}^t] \\ \overset{(k)}{\leq}& 2 \cdot (A_2)^2 + 2 \cdot (\frac{1}{2}\alpha^{t+1} + \frac{\alpha}{1-\alpha}) \cdot (A_2)^2 \\ =& (\alpha^{t+1} + \frac{2}{1-\alpha}) \cdot (A_2)^2, \end{aligned} \tag{13}$$

where (i) is derived based on that $\|\mathbf{x} + \mathbf{y}\|^2 \leq 2\|\mathbf{x}\|^2 + 2\|\mathbf{y}\|^2,\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, (j) is derived by the definition of $(\delta_{\max}, A_1, A_2)$-effective aggregation function, and (k) is derived based on Inequality (12).

By Inequality (12) and (13), the claimed property also holds for $t' = t$.

In conclusion, for all $t = 0, 1, \ldots, T-1$, we have:

$$\mathbb{E}[\|\mathbf{G}^t - \mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] \leq (\frac{1}{2}\alpha^{t+1} + \frac{\alpha}{1-\alpha}) \cdot (A_2)^2, \tag{14}$$

and

$$\mathbb{E}[\|\mathbf{G}^t\|^2 \mid \mathbf{w}^t] \leq (\alpha^{t+1} + \frac{2}{1-\alpha}) \cdot (A_2)^2. \tag{15}$$

Also, $\mathbb{E}[\|\mathbf{G}^t\| \mid \mathbf{w}^t]^2 + Var[\|\mathbf{G}^t\| \mid \mathbf{w}^t] = \mathbb{E}[\|\mathbf{G}^t\|^2 \mid \mathbf{w}^t]$. Therefore,

$$\mathbb{E}[\|\mathbf{G}^t\| \mid \mathbf{w}^t] = \sqrt{\mathbb{E}[\|\mathbf{G}^t\| \mid \mathbf{w}^t]^2} \leq \sqrt{\alpha^{t+1} + \frac{2}{1-\alpha}} \cdot A_2. \tag{16}$$

We have:

$$\begin{aligned} & \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}^t \mid \mathbf{w}^t] \\ =& \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}^t_{syn} \mid \mathbf{w}^t] + \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T (\mathbf{G}^t - \mathbf{G}^t_{syn}) \mid \mathbf{w}^t] \\ \overset{(l)}{\geq}& \eta \cdot (\|\nabla F(\mathbf{w}^t)\|^2 - A_1) + \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T (\mathbf{G}^t - \mathbf{G}^t_{syn}) \mid \mathbf{w}^t] \\ \geq& \eta \cdot \|\nabla F(\mathbf{w}^t)\|^2 - \eta \cdot A_1 - \eta \cdot \|\nabla F(\mathbf{w}^t)\| \cdot \|\mathbb{E}[(\mathbf{G}^t - \mathbf{G}^t_{syn}) \mid \mathbf{w}^t]\| \\ \overset{(m)}{\geq}& \eta \cdot \|\nabla F(\mathbf{w}^t)\|^2 - \eta \cdot A_1 - \eta \cdot D \cdot \|\mathbb{E}[(\mathbf{G}^t - \mathbf{G}^t_{syn}) \mid \mathbf{w}^t]\| \\ \overset{(n)}{\geq}& \eta \cdot \|\nabla F(\mathbf{w}^t)\|^2 - \eta \cdot A_1 - \eta \cdot D \cdot \sqrt{\frac{1}{2}\alpha^{t+1} + \frac{\alpha}{1-\alpha}} \cdot A_2, \end{aligned} \tag{17}$$

where (l) is derived based on the definition of $(\delta_{\max}, A_1, A_2)$-effective aggregation function, (m) is derived by Assumption 3, and (n) is derived based on Inequality (14).

Combining Inequalities (9), (15), (17) and taking total expectation, we have:

$$\begin{aligned}\mathbb{E}[F(\mathbf{w}^{t+1})] \leq& \mathbb{E}[F(\mathbf{w}^t)] - \eta \cdot \mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2] \\ &+ \eta \cdot A_1 + \eta \cdot D\sqrt{\frac{1}{2}\alpha^{t+1} + \frac{\alpha}{1-\alpha}} \cdot A_2 + \frac{1}{2}\eta^2 L(\alpha^{t+1} + \frac{2}{1-\alpha}) \cdot (A_2)^2.\end{aligned}$$

By telescoping, we have:

$$\begin{aligned}\eta \cdot \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2] \leq& \{F(\mathbf{w}^0) - \mathbb{E}[F(\mathbf{w}^T)]\} + \frac{1}{2}\eta^2 TL(\alpha + \frac{2}{1-\alpha}) \cdot (A_2)^2 \\ &+ \eta T A_1 + \eta T D \cdot \sqrt{\frac{1}{2}\alpha + \frac{\alpha}{1-\alpha}} \cdot A_2.\end{aligned}$$

Divide both sides of the equation by ηT, and let $\eta = O(\frac{1}{\sqrt{LT}})$:

$$\begin{aligned}&\frac{\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2]}{T} \\ \leq& \frac{\{F(\mathbf{w}^0) - \mathbb{E}[F(\mathbf{w}^T)]\}}{\eta T} + \frac{1}{2}\eta L(\alpha + \frac{2}{1-\alpha}) \cdot (A_2)^2 + A_1 + D \cdot \sqrt{\frac{1}{2}\alpha + \frac{\alpha}{1-\alpha}} \cdot A_2 \\ \leq& \frac{\sqrt{L}[F(\mathbf{w}^0) - F^*]}{\sqrt{T}} + \frac{\sqrt{L}(\frac{1}{2}\alpha + \frac{1}{1-\alpha}) \cdot (A_2)^2}{\sqrt{T}} + A_1 + \alpha^{\frac{1}{2}}[\frac{3-\alpha}{2(1-\alpha)}]^{\frac{1}{2}} \cdot DA_2.\end{aligned}$$

Since $\eta = O(\frac{1}{\sqrt{LT}})$ and $B = \lfloor r/\delta_{\max} \rfloor + 1$, we have that

$$\alpha = 2\eta^2 L^2 \tau_{max}^2 (B - r) = O\left(\frac{L\tau_{max}^2 (r - \delta_{\max} r + 1)}{\delta_{\max} T}\right).$$

Finally, it is obtained that:

$$\begin{aligned}\frac{\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2]}{T} \leq& O\left(\frac{\sqrt{L} \cdot [F(\mathbf{w}^0) - F^*]}{\sqrt{T}}\right) + O\left(\frac{\sqrt{L}(A_2)^2(1+\alpha)}{\sqrt{T}}\right) \\ &+ O\left(\alpha^{\frac{1}{2}} D A_2\right) + A_1 \\ =& O\left(\frac{L^{\frac{1}{2}}[F(\mathbf{w}^0) - F^*]}{T^{\frac{1}{2}}}\right) + O\left(\frac{(r - \delta_{\max} r + 1)^{\frac{1}{2}} L^{\frac{1}{2}} \tau_{max} D A_2}{\delta_{\max}^{\frac{1}{2}} T^{\frac{1}{2}}}\right) \\ &+ O\left(\frac{L^{\frac{1}{2}}(A_2)^2}{T^{\frac{1}{2}}}\right) + O\left(\frac{(r - \delta_{\max} r + 1) L^{\frac{3}{2}} (A_2)^2 \tau_{max}^2}{\delta_{\max} T^{\frac{3}{2}}}\right) + A_1.\end{aligned}$$

■

B.6 Proof of Theorem 13

Proof The proof of this theorem is similar to that of Theorem 12. The main differences are the choices of the values α (in Theorem 12) and $\tilde{\alpha}$ (here in Theorem 13). For more readability, we still present the detailed proof processes here.

Let $\mathbf{h}'_b$ be the value of the b-th buffer, if all received loyal gradients were computed based on $\mathbf{w}^t$. Since $\mathbf{G}^t = Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}])$, we have:

$$\begin{aligned}
&\mathbb{E}[F(\mathbf{w}^{t+1}) \mid \mathbf{w}^t] \\
=&\mathbb{E}[F(\mathbf{w}^t - \eta \cdot \mathbf{G}^t) \mid \mathbf{w}^t] \\
\overset{(a)}{\leq}&\mathbb{E}[F(\mathbf{w}^t) - \eta \cdot \nabla F(\mathbf{w}^t)^T \mathbf{G}^t + \frac{L}{2}\eta^2 ||\mathbf{G}^t||^2 \mid \mathbf{w}^t] \\
=&F(\mathbf{w}^t) - \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}^t \mid \mathbf{w}^t] + \frac{\eta^2 L}{2}\mathbb{E}[||\mathbf{G}^t||^2 \mid \mathbf{w}^t], \qquad (18)
\end{aligned}$$

where (a) is derived based on Assumption 5.

Firstly, we estimate the value of $\mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}^t \mid \mathbf{w}^t]$.

Since there are at most r Byzantine workers, at most r buffers may contain Byzantine gradients. Without loss of generality, suppose only the first r buffers may contain Byzantine gradients.

Let $\mathbf{G}^t_{syn} = Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{r-1}, \mathbf{h}'_r, \ldots, \mathbf{h}'_{B-1}])$, where $\mathbf{h}_1, \ldots, \mathbf{h}_r$ may contain Byzantine gradients and be arbitrary value, and $\mathbf{h}'_r, \ldots, \mathbf{h}'_{B-1}$ each stores loyal gradients computed based on $\mathbf{w}^t$. Thus,

$$\mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}^t_{syn} \mid \mathbf{w}^t] \geq \|\nabla F(\mathbf{w}^t)\|^2 - A_1, \qquad (19)$$

$$\mathbb{E}[\|\mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] \leq (A_2)^2. \qquad (20)$$

Let $\tilde{\alpha} = 2\eta^2 L^2 \tau_{max}^2 (1-\mu)^2 (B-r) < 1$.

We claim that

$$\mathbb{E}[\|\mathbf{G}^t - \mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] \leq (\frac{1}{2}\tilde{\alpha}^{t+1} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}) \cdot (A_2)^2,$$

and

$$\mathbb{E}[\|\mathbf{G}^t\|^2 \mid \mathbf{w}^t] \leq (\tilde{\alpha}^{t+1} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2.$$

Now we prove it by induction on t.

Step 1. When $t = 0$, all gradients are computed according to $\mathbf{w}^0$, and we have $\mathbf{G}^0 = \mathbf{G}^0_{syn}$. Thus,

$$\mathbb{E}[\|\mathbf{G}^0 - \mathbf{G}^0_{syn}\|^2 \mid \mathbf{w}^0] = 0 \leq (\frac{1}{2}\tilde{\alpha}^1 + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}) \cdot (A_2)^2,$$

$$\mathbb{E}[\|\mathbf{G}^0\|^2 \mid \mathbf{w}^0] = \mathbb{E}[\|\mathbf{G}^0_{syn}\|^2 \mid \mathbf{w}^0] \leq (A_2)^2 \leq (\tilde{\alpha}^1 + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2.$$

Step 2. If

$$\mathbb{E}[\|\mathbf{G}^{t'} - \mathbf{G}^{t'}_{syn}\|^2 \mid \mathbf{w}^{t'}] \leq (\frac{1}{2}\tilde{\alpha}^{t'+1} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}) \cdot (A_2)^2,$$

$$\mathbb{E}[\|\mathbf{G}^{t'}\|^2 \mid \mathbf{w}^{t'}] \leq (\tilde{\alpha}^{t'+1} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2,$$

holds for all $t' = 0, 1, \ldots, t-1$ (induction hypothesis), then:

$$\begin{aligned}
&\mathbb{E}[\|\mathbf{G}^t - \mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] \\
=&\mathbb{E}[\|Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{r-1}, \mathbf{h}_r, \ldots, \mathbf{h}_{B-1}]) - Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{r-1}, \mathbf{h}'_r, \ldots, \mathbf{h}'_{B-1}])\|^2 \mid \mathbf{w}^t] \\
\overset{(b)}{\leq}&\mathbb{E}[\sum_{b=r}^{B-1} \|\mathbf{h}_b - \mathbf{h}'_b\|^2 \mid \mathbf{w}^t] \\
\overset{(c)}{=}&\sum_{b=r}^{B-1} \mathbb{E}[\|\frac{1}{N_b^t}\sum_{k=1}^{N_b^t}(1-\mu)(\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \nabla f(\mathbf{w}^t; z_{i_k}))\|^2 \mid \mathbf{w}^t] \\
\overset{(d)}{\leq}&\sum_{b=r}^{B-1} \mathbb{E}[\frac{1}{N_b^t}\sum_{k=1}^{N_b^t} \|(1-\mu)(\nabla f(\mathbf{w}^{t_k}; z_{i_k}) - \nabla f(\mathbf{w}^t; z_{i_k}))\|^2 \mid \mathbf{w}^t] \\
\overset{(e)}{\leq}&\sum_{b=r}^{B-1} \mathbb{E}[\frac{1}{N_b^t}\sum_{k=1}^{N_b^t}(1-\mu)^2 L^2\|\mathbf{w}^{t_k} - \mathbf{w}^t\|^2 \mid \mathbf{w}^t] \\
=&\sum_{b=r}^{B-1} \frac{L^2(1-\mu)^2}{N_b^t}\sum_{k=1}^{N_b^t}\mathbb{E}[\|\mathbf{w}^{t_k} - \mathbf{w}^t\|^2 \mid \mathbf{w}^t] \\
=&\sum_{b=r}^{B-1} \frac{L^2(1-\mu)^2}{N_b^t}\sum_{k=1}^{N_b^t}\mathbb{E}[\|\sum_{t'=t_k}^{t-1} \eta \cdot \mathbf{G}^{t'}\|^2 \mid \mathbf{w}^t] \\
\overset{(f)}{\leq}&\sum_{b=r}^{B-1} \frac{\eta^2 L^2(1-\mu)^2}{N_b^t}\sum_{k=1}^{N_b^t}\mathbb{E}[(t-t_k)\sum_{t'=t_k}^{t-1} \|\mathbf{G}^{t'}\|^2 \mid \mathbf{w}^t] \\
\overset{(g)}{\leq}&\sum_{b=r}^{B-1} \frac{\eta^2 L^2(1-\mu)^2}{N_b^t}\sum_{k=1}^{N_b^t}[(t-t_k)\sum_{t'=t_k}^{t-1} (\tilde{\alpha}^{t'+1} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2] \\
\leq&\sum_{b=r}^{B-1} \frac{\eta^2 L^2(1-\mu)^2}{N_b^t}\sum_{k=1}^{N_b^t}[(t-t_k)\sum_{t'=t_k}^{t-1} (\tilde{\alpha}^{t} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2] \\
\overset{(h)}{\leq}&\sum_{b=r}^{B-1}(\eta^2 L^2(1-\mu)^2\tau_{max}^2) \cdot (\tilde{\alpha}^t + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2 \\
=&(\eta^2 L^2(1-\mu)^2\tau_{max}^2(B-r)) \cdot (\tilde{\alpha}^t + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2 \\
\overset{(i)}{=}&\frac{1}{2}\tilde{\alpha} \cdot (\tilde{\alpha}^t + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2 \\
=&(\frac{1}{2}\tilde{\alpha}^{t+1} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}) \cdot (A_2)^2,
\end{aligned} \tag{21}$$

where (b) is derived based on the definition of stable aggregation function, (c) is derived based on the worker momentum updating formula $\mathbf{u} \leftarrow \mu \cdot \mathbf{u} + (1-\mu) \cdot \nabla f(\mathbf{w}; z_i)$, (d) is derived based on Cauchy's Inequality, (e) is derived based on Assumption 5, (f) is also derived based on Cauchy's Inequality, (g) is derived based on the induction hypothesis, (h) is derived based on that $t - t_k \le \tau_{max}$, and (i) is derived based on that $\tilde{\alpha} = 2\eta^2 L^2 \tau_{max}^2 (1-\mu)^2 (B-r)$.

Therefore,

$$\begin{aligned}
\mathbb{E}[\|\mathbf{G}^t\|^2 \mid \mathbf{w}^t] =& \mathbb{E}[||\mathbf{G}^t_{syn} + (\mathbf{G}^t - \mathbf{G}^t_{syn})||^2 \mid \mathbf{w}^t] \\
\overset{(j)}{\le}& 2 \cdot \mathbb{E}[\|\mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] + 2 \cdot \mathbb{E}[||\mathbf{G}^t - \mathbf{G}^t_{syn}||^2 \mid \mathbf{w}^t] \\
\overset{(k)}{\le}& 2 \cdot (A_2)^2 + 2 \cdot \mathbb{E}[||\mathbf{G}^t - \mathbf{G}^t_{syn}||^2 \mid \mathbf{w}^t] \\
\overset{(l)}{\le}& 2 \cdot (A_2)^2 + 2 \cdot (\frac{1}{2}\tilde{\alpha}^{t+1} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}) \cdot (A_2)^2 \\
=& (\tilde{\alpha}^{t+1} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2,
\end{aligned} \tag{22}$$

where (j) is derived based on that $\|\mathbf{x}+\mathbf{y}\|^2 \le 2\|\mathbf{x}\|^2 + 2\|\mathbf{y}\|^2,\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, (k) is derived by the definition of $(\delta_{\max}, A_1, A_2)$-effective aggregation function, and (l) is derived based on Inequality (21).

By Inequality (21) and (22), the claimed property also holds for $t' = t$.

In conclusion, for all $t = 0, 1, \ldots, T-1$, we have:

$$\mathbb{E}[||\mathbf{G}^t - \mathbf{G}^t_{syn}\|^2 \mid \mathbf{w}^t] \le (\frac{1}{2}\tilde{\alpha}^{t+1} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}) \cdot (A_2)^2, \tag{23}$$

and

$$\mathbb{E}[\|\mathbf{G}^t\|^2 \mid \mathbf{w}^t] \le (\tilde{\alpha}^{t+1} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2. \tag{24}$$

Also, $\mathbb{E}[\|\mathbf{G}^t\| \mid \mathbf{w}^t]^2 + Var[\|\mathbf{G}^t\| \mid \mathbf{w}^t] = \mathbb{E}[\|\mathbf{G}^t\|^2 \mid \mathbf{w}^t]$. Therefore,

$$\mathbb{E}[\|\mathbf{G}^t\| \mid \mathbf{w}^t] = \sqrt{\mathbb{E}[\|\mathbf{G}^t\| \mid \mathbf{w}^t]^2} \le \sqrt{\tilde{\alpha}^{t+1} + \frac{2}{1-\tilde{\alpha}}} \cdot A_2. \tag{25}$$

We have:

$$\begin{aligned}
& \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}^t \mid \mathbf{w}^t] \\
=& \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}^t_{syn} \mid \mathbf{w}^t] + \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T (\mathbf{G}^t - \mathbf{G}^t_{syn}) \mid \mathbf{w}^t] \\
\overset{(m)}{\ge}& \eta \cdot (\|\nabla F(\mathbf{w}^t)\|^2 - A_1) + \eta \cdot \mathbb{E}[\nabla F(\mathbf{w}^t)^T (\mathbf{G}^t - \mathbf{G}^t_{syn}) \mid \mathbf{w}^t] \\
\ge& \eta \cdot \|\nabla F(\mathbf{w}^t)\|^2 - \eta \cdot A_1 - \eta \cdot \|\nabla F(\mathbf{w}^t)\| \cdot \|\mathbb{E}[(\mathbf{G}^t - \mathbf{G}^t_{syn}) \mid \mathbf{w}^t]\| \\
\overset{(n)}{\ge}& \eta \cdot \|\nabla F(\mathbf{w}^t)\|^2 - \eta \cdot A_1 - \eta \cdot D \cdot \|\mathbb{E}[(\mathbf{G}^t - \mathbf{G}^t_{syn}) \mid \mathbf{w}^t]\| \\
\overset{(p)}{\ge}& \eta \cdot \|\nabla F(\mathbf{w}^t)\|^2 - \eta \cdot A_1 - \eta \cdot D \cdot \sqrt{\frac{1}{2}\tilde{\alpha}^{t+1} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}} \cdot A_2,
\end{aligned} \tag{26}$$

where (m) is derived based on the definition of $(\delta_{\max}, A_1, A_2)$-effective aggregation function, (n) is derived by Assumption 3, and (p) is derived based on Inequality (23).

Combining Inequalities (18), (24), (26) and taking total expectation, we have:

$$\begin{aligned}\mathbb{E}[F(\mathbf{w}^{t+1})] \leq & \mathbb{E}[F(\mathbf{w}^t)] - \eta \cdot \mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2] \\ & + \eta \cdot A_1 + \eta \cdot D\sqrt{\frac{1}{2}\tilde{\alpha}^{t+1} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}} \cdot A_2 + \frac{1}{2}\eta^2 L(\tilde{\alpha}^{t+1} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2.\end{aligned}$$

By telescoping, we have:

$$\begin{aligned}\eta \cdot \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2] \leq & \{F(\mathbf{w}^0) - \mathbb{E}[F(\mathbf{w}^T)]\} + \frac{1}{2}\eta^2 TL(\tilde{\alpha} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2 \\ & + \eta T A_1 + \eta T D \cdot \sqrt{\frac{1}{2}\tilde{\alpha} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}} \cdot A_2.\end{aligned}$$

Divide both sides of the equation by ηT, and let $\eta = O(\frac{1}{\sqrt{LT}})$:

$$\begin{aligned}& \frac{\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2]}{T} \\ \leq & \frac{\{F(\mathbf{w}^0) - \mathbb{E}[F(\mathbf{w}^T)]\}}{\eta T} + \frac{1}{2}\eta L(\tilde{\alpha} + \frac{2}{1-\tilde{\alpha}}) \cdot (A_2)^2 + A_1 + D \cdot \sqrt{\frac{1}{2}\tilde{\alpha} + \frac{\tilde{\alpha}}{1-\tilde{\alpha}}} \cdot A_2 \\ \leq & \frac{\sqrt{L}[F(\mathbf{w}^0) - F^*]}{\sqrt{T}} + \frac{\sqrt{L}(\frac{1}{2}\tilde{\alpha} + \frac{1}{1-\tilde{\alpha}}) \cdot (A_2)^2}{\sqrt{T}} + A_1 + \tilde{\alpha}^{\frac{1}{2}}[\frac{3-\tilde{\alpha}}{2(1-\tilde{\alpha})}]^{\frac{1}{2}} \cdot DA_2.\end{aligned}$$

Since $\eta = O(\frac{1}{\sqrt{LT}})$ and $B = \lfloor r/\delta_{\max} \rfloor + 1$, we have that

$$\tilde{\alpha} = 2\eta^2 L^2 \tau_{max}^2 (1-\mu)^2 (B - r) = O\left(\frac{L\tau_{max}^2(1-\mu)^2(r - \delta_{\max} r + 1)}{\delta_{\max} T}\right).$$

Finally, it is obtained that:

$$\begin{aligned}& \frac{\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2]}{T} \\ \leq & \; O\left(\frac{\sqrt{L} \cdot [F(\mathbf{w}^0) - F^*]}{\sqrt{T}}\right) + O\left(\frac{\sqrt{L}(A_2)^2(1+\tilde{\alpha})}{\sqrt{T}}\right) \\ & + O\left(\tilde{\alpha}^{\frac{1}{2}} DA_2\right) + A_1 \\ = & \; O\left(\frac{L^{\frac{1}{2}}[F(\mathbf{w}^0) - F^*]}{T^{\frac{1}{2}}}\right) + O\left(\frac{(r - \delta_{\max} r + 1)^{\frac{1}{2}} L^{\frac{1}{2}} \tau_{max} D A_2 (1-\mu)}{\delta_{\max}^{\frac{1}{2}} T^{\frac{1}{2}}}\right) \\ & + O\left(\frac{L^{\frac{1}{2}}(A_2)^2}{T^{\frac{1}{2}}}\right) + O\left(\frac{(r - \delta_{\max} r + 1) L^{\frac{3}{2}} (A_2)^2 \tau_{max}^2 (1-\mu)^2}{\delta_{\max} T^{\frac{3}{2}}}\right) + A_1.\end{aligned}$$

■

B.7 Proof of Proposition 14

Proof Under the condition that $\forall \mathbf{w}^t \in \mathbb{R}^d,\ \mathbb{E}[\|\mathbf{G}_{syn}^t - \nabla F(\mathbf{w}^t)\| \mid \mathbf{w}^t] \leq D' < D$, we have:

$$\begin{aligned}
&\mathbb{E}[\nabla F(\mathbf{w}^t)^T \mathbf{G}_{syn}^t \mid \mathbf{w}^t] \\
=& \mathbb{E}[\nabla F(\mathbf{w}^t)^T\ [\nabla F(\mathbf{w}^t) + (\mathbf{G}_{syn}^t - \nabla F(\mathbf{w}^t))] \mid \mathbf{w}^t] \\
=& \|\nabla F(\mathbf{w}^t)\|^2 + \mathbb{E}[\nabla F(\mathbf{w}^t)^T (\mathbf{G}_{syn}^t - \nabla F(\mathbf{w}^t)) \mid \mathbf{w}^t] \\
\geq& \|\nabla F(\mathbf{w}^t)\|^2 - \|\nabla F(\mathbf{w}^t)\| \cdot \mathbb{E}[\|\mathbf{G}_{syn}^t - \nabla F(\mathbf{w}^t)\| \mid \mathbf{w}^t] \\
\geq& \|\nabla F(\mathbf{w}^t)\|^2 - DD'.
\end{aligned}$$

Combining with the property (a) of $(\delta_{\max}, A_1, A_2)$-effective aggregation function, we have $A_1 \leq DD' < D^2$.

■

B.8 Proof of Theorem 15

Proof At the end of the t-th iteration, the server updates the global model by letting $\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \cdot \mathbf{G}^t$, where $\mathbf{G}^t = Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}])$. Please note that $\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}$ may differ for different t's. However, when it does not cause confusion, we will omit the superscript t for more readability. Let $\mathcal{H}^t = \{b \mid \mathbf{h}_b$ does not contain Byzantine values$\}$ denote the set of credible buffer at the t-th iteration. Since there are at most r Byzantine workers, we have $|\mathcal{H}^t| \geq B - r$. Let

$$\bar{\mathbf{G}}^t = \frac{1}{|\mathcal{H}^t|} \sum_{b \in \mathcal{H}^t} \mathbf{h}_b$$

denote the mean value of credible buffers. At the beginning, we introduce the following lemma, which provides an upper bound for the difference between received local momentums and the global gradient.

Lemma 19. *Under the same conditions in Theorem 15, for each local momentum* $\mathbf{u}$ *stored in any credible buffer* $\mathbf{h}_b$ $(b \in \mathcal{H}^t)$ *at the t-th iteration, we have:*

$$\left\|\mathbb{E}[\mathbf{u} - \nabla F(\mathbf{w}^t)]\right\|^2 \leq 4(4c\delta + 1)\left[4 - \lambda + 2\sqrt{4 - 2\lambda + 4\lambda^{-2}} + 2\lambda^{-2}\right] \eta^2 L^2 (\tau_{max} + 1)^2 D^2. \tag{27}$$

Proof In the proof below, all the mentioned local momentums and stochastic gradients are from a loyal worker_k. For simplicity, we will omit the worker ID 'k' when there is no confusion. Let t_v $(v = 0, 1, \ldots)$ denote the iteration numbers corresponding to the model parameters that worker_k received during the training process. Thus, we have

$$0 = t_0 \leq t_1 \leq \ldots \leq t_v \leq \ldots \qquad \text{and} \qquad t_v - t_{v-1} \leq \tau_{max} + 1.$$

Therefore, the global model parameter is $\mathbf{w}^{t_v}$ when worker_k sends local momentums for the v-th time. Let $\mathbf{u}_{k,v}$ denote the v-th sent momentum from worker_k. We are left to provide an upper bound for $\|\mathbb{E}[\mathbf{u}_{k,v}] - \mathbb{E}[\nabla F(\mathbf{w}^{t_v})]\|^2$. Using Assumption 2, Assumption 5 and the i.i.d.-ness, it is obtained that

$$\begin{aligned}
&\left\|\mathbb{E}[\mathbf{u}_{k,v} - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \\
=\ &\left\|\mu \cdot \mathbb{E}[\mathbf{u}_{k,v-1}] + (1-\mu) \cdot \mathbb{E}[\nabla f(\mathbf{w}^{t_{v-1}}; z_{i_k})] - \mathbb{E}[\nabla F(\mathbf{w}^{t_v})]\right\|^2 \\
=\ &\left\|\mu \cdot \mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_v})] + (1-\mu) \cdot \mathbb{E}[\nabla F(\mathbf{w}^{t_{v-1}}) - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \\
=\ &\left\|(1-\lambda) \cdot \mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_v})] + \lambda \cdot \mathbb{E}[\nabla F(\mathbf{w}^{t_{v-1}}) - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \\
\leq\ &(1-\lambda)^2 \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_v})]\right\|^2 + \lambda^2 \left\|\mathbb{E}[\nabla F(\mathbf{w}^{t_{v-1}}) - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \\
&+ 2\lambda(1-\lambda) \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_v})]\right\| \cdot \left\|\mathbb{E}[\nabla F(\mathbf{w}^{t_{v-1}}) - \nabla F(\mathbf{w}^{t_v})]\right\|. \qquad (28)
\end{aligned}$$

By using Assumption 3, the L_2-norm of each received momentum from loyal workers is bounded by D. Thus, for any $b, b' \in \mathcal{H}$, $\|\mathbf{h}_b - \mathbf{h}_{b'}\|^2 \leq 4D^2$. Since $t_v - t_{v-1} \leq \tau_{max}$, we have:

$$\begin{aligned}
\mathbb{E}\|\mathbf{w}^{t_{v-1}} - \mathbf{w}^{t_v}\|^2 = \mathbb{E}\left\|\sum_{t=t_{v-1}}^{t_v-1} \eta \cdot \mathbf{G}^t\right\|^2 &\leq 2\eta^2\mathbb{E}\left\|\sum_{t=t_{v-1}}^{t_v-1} (\mathbf{G}^t - \bar{\mathbf{G}}^t)\right\|^2 + 2\eta^2\mathbb{E}\left\|\sum_{t=t_{v-1}}^{t_v-1} \bar{\mathbf{G}}^t\right\|^2 \\
&\leq 2\eta^2(\tau_{max}+1)^2 c\delta(4D^2) + 2\eta^2(\tau_{max}+1)^2 D^2 \\
&= 2(4c\delta+1)\eta^2(\tau_{max}+1)^2 D^2. \qquad (29)
\end{aligned}$$

The first term in the right-hand side (RHS) of inequality (28) is bounded by

$$\begin{aligned}
&(1-\lambda)^2 \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \\
=&(1-\lambda)^2 \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_{v-1}})] + \mathbb{E}[\nabla F(\mathbf{w}^{t_{v-1}}) - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \\
\leq&(1-\lambda)(1+\frac{\lambda}{2}) \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_{v-1}})]\right\|^2 + (1-\lambda)(1+\frac{2}{\lambda}) \left\|\mathbb{E}[\nabla F(\mathbf{w}^{t_{v-1}}) - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \\
\leq&(1-\lambda)(1+\frac{\lambda}{2}) \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_{v-1}})]\right\|^2 + (1-\lambda)(1+\frac{2}{\lambda})\mathbb{E} \left\|\nabla F(\mathbf{w}^{t_{v-1}}) - \nabla F(\mathbf{w}^{t_v})\right\|^2 \\
\leq&(1-\frac{\lambda}{2}) \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_{v-1}})]\right\|^2 + \frac{2L^2}{\lambda}\mathbb{E} \left\|\mathbf{w}^{t_{v-1}} - \mathbf{w}^{t_v}\right\|^2 \\
\leq&(1-\frac{\lambda}{2}) \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_{v-1}})]\right\|^2 + \frac{4(4c\delta+1)\eta^2 L^2(\tau_{max}+1)^2 D^2}{\lambda}. \qquad (30)
\end{aligned}$$

The second term in the RHS of inequality (28) is bounded by

$$\lambda^2 \left\|\mathbb{E}[\nabla F(\mathbf{w}^{t_{v-1}}) - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \leq \lambda^2 L^2 \mathbb{E}\left\|\mathbf{w}^{t_{v-1}} - \mathbf{w}^{t_v}\right\|^2 \leq 2(4c\delta+1)\lambda^2\eta^2 L^2(\tau_{max}+1)^2 D^2. \qquad (31)$$

Let constants $Q = 2(4c\delta+1)\eta^2 L^2(\tau_{max}+1)^2 D^2$. Substituting (30) and (31) into (28), we have

$$\begin{aligned}
\left\|\mathbb{E}[\mathbf{u}_{k,v} - \nabla F(\mathbf{w}^{t_v})]\right\|^2 \leq\ &(1-\frac{\lambda}{2}) \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_{v-1}})]\right\|^2 + \frac{2}{\lambda}Q + \lambda^2 Q \\
&+ 2\lambda Q^{\frac{1}{2}} \left((1-\frac{\lambda}{2}) \left\|\mathbb{E}[\mathbf{u}_{k,v-1} - \nabla F(\mathbf{w}^{t_{v-1}})]\right\|^2 + \frac{2}{\lambda}Q\right)^{\frac{1}{2}}. \qquad (32)
\end{aligned}$$

Let $\xi_v = \|\mathbb{E}[\mathbf{u}_{k,v} - \nabla F(\mathbf{w}^{t_v})]\| \geq 0$, we have

$$(\xi_v)^2 \leq \left(1-\frac{\lambda}{2}\right)(\xi_{v-1})^2 + \frac{2}{\lambda}Q + \lambda^2 Q + 2\lambda\sqrt{\left(1-\frac{\lambda}{2}\right)Q(\xi_{v-1})^2 + \frac{2}{\lambda}Q^2}. \tag{33}$$

By mathematical induction on v, then we prove that

$$\xi_v \leq \left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)\sqrt{Q}, \qquad \forall v \in \mathbb{N}_+. \tag{34}$$

Step 1. For $v=1$, using Inequality (29), Assumption 2 and Assumption 5, we have

$$\begin{aligned}\xi_1 = \|\mathbb{E}[\mathbf{u}_{k,1} - \nabla F(\mathbf{w}^{t_1})]\| &= \|\mathbb{E}[\nabla f(\mathbf{w}^{t_0}; z_{i_k}) - \nabla F(\mathbf{w}^{t_1})]\| \\ &= \|\mathbb{E}[\nabla F(\mathbf{w}^{t_0}) - \nabla F(\mathbf{w}^{t_1})]\| \\ &\leq L \cdot \mathbb{E}\|\mathbf{w}^{t_0} - \mathbf{w}^{t_1}\| \\ &\leq \sqrt{2(4c\delta+1)\eta^2 L^2 (\tau_{max}+1)^2 D^2} \\ &= \sqrt{Q} \qquad \leq \left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)\sqrt{Q}.\end{aligned} \tag{35}$$

Step 2. Suppose that $\xi_v \leq \left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)\sqrt{Q}$ holds for v. Then for $v+1$, we have

$$\begin{aligned}(\xi_{v+1})^2 &\leq \left(1-\frac{\lambda}{2}\right)(\xi_v)^2 + \frac{2}{\lambda}Q + \lambda^2 Q + 2\lambda\sqrt{\left(1-\frac{\lambda}{2}\right)Q(\xi_v)^2 + \frac{2}{\lambda}Q^2} \\ &= \left(\sqrt{\left(1-\frac{\lambda}{2}\right)(\xi_v)^2 + \frac{2}{\lambda}Q} + \lambda\sqrt{Q}\right)^2.\end{aligned} \tag{36}$$

Therefore,

$$\begin{aligned}\xi_{v+1} \leq& \sqrt{\left(1-\frac{\lambda}{2}\right)(\xi_v)^2 + \frac{2}{\lambda}Q} + \lambda\sqrt{Q} \\ \leq& \sqrt{\left(1-\frac{\lambda}{2}\right)\left(\left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)\sqrt{Q}\right)^2 + \frac{2}{\lambda}Q} + \sqrt{\lambda^2 Q} \\ =& \left[\sqrt{\left(1-\frac{\lambda}{2}\right)\left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)^2 + \frac{2}{\lambda}} + \lambda\right] \cdot \sqrt{Q}.\end{aligned} \tag{37}$$

Since

$$\sqrt{\left(1-\frac{\lambda}{2}\right)\left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)^2 + \frac{2}{\lambda}} + \lambda$$

$$
\begin{aligned}
=&\sqrt{\left(1-\frac{\lambda}{2}\right)\left(4+\frac{4\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}+\frac{4}{\lambda^2}+4-2\lambda\right)+\frac{2}{\lambda}+\lambda}\\
=&\sqrt{\left(8+\frac{4\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}+\frac{4}{\lambda^2}-2\lambda\right)-\left(4\lambda+2\sqrt{4+(4-2\lambda)\lambda^2}+\frac{2}{\lambda}-\lambda^2\right)+\frac{2}{\lambda}+\lambda}\\
=&\sqrt{4+\left(\frac{4}{\lambda^2}+4-2\lambda\right)+\lambda^2+\frac{4\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}-2\sqrt{4+(4-2\lambda)\lambda^2}-4\lambda+\lambda}\\
=&\sqrt{2^2+\left(\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)^2+\lambda^2+\frac{4\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}-\frac{2\lambda\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}-4\lambda+\lambda}\\
=&\sqrt{\left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}-\lambda\right)^2+\lambda}\\
=&2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}-\lambda+\lambda\\
=&2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}, \qquad (38)
\end{aligned}
$$

we have

$$\xi_{v+1}\leq\left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)\sqrt{Q}. \qquad (39)$$

It indicates that the induction hypothesis also holds for $v+1$. Consequently, for any positive integer v, we have

$$\xi_{v}\leq\left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)\sqrt{Q}. \qquad (40)$$

Recall that $\xi_v=\|\mathbb{E}[\mathbf{u}_{k,v}-\nabla F(\mathbf{w}^{t_v})]\|$ and finally it is obtained that

$$
\begin{aligned}
\|\mathbb{E}[\mathbf{u}_{k,v}-\nabla F(\mathbf{w}^{t_v})]\|^2&\leq\left(2+\frac{\sqrt{4+(4-2\lambda)\lambda^2}}{\lambda}\right)^2 Q\\
&=4(4c\delta+1)\left[4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2}\right]\eta^2L^2(\tau_{max}+1)^2D^2. \qquad (41)
\end{aligned}
$$

■

We have finished the proof of Lemma 19 and we will continue to prove Theorem 15.

$$
\begin{aligned}
&\mathbb{E}[F(\mathbf{w}^{t+1})\mid\mathbf{w}^t]\\
=&\mathbb{E}[F(\mathbf{w}^t-\eta\cdot\mathbf{G}^t)\mid\mathbf{w}^t]\\
\overset{(a)}{\leq}&\mathbb{E}\left[F(\mathbf{w}^t)-\eta\cdot\nabla F(\mathbf{w}^t)^T\mathbf{G}^t+\frac{L}{2}\eta^2\|\mathbf{G}^t\|^2\;\middle|\;\mathbf{w}^t\right]\\
=&\mathbb{E}\left[F(\mathbf{w}^t)-\eta\cdot\frac{1}{2}\Big(\|\nabla F(\mathbf{w}^t)\|^2+\|\mathbf{G}^t\|^2-\|\nabla F(\mathbf{w}^t)-\mathbf{G}^t\|^2\Big)+\frac{L}{2}\eta^2\|\mathbf{G}^t\|^2\;\middle|\;\mathbf{w}^t\right]
\end{aligned}
$$

$$\begin{aligned}
=&\mathbb{E}\left[F(\mathbf{w}^t) - \frac{\eta}{2}\|\nabla F(\mathbf{w}^t)\|^2 + \frac{\eta}{2}\|\nabla F(\mathbf{w}^t) - \mathbf{G}^t\|^2 - \frac{\eta(1-\eta L)}{2}||\mathbf{G}^t||^2 \;\Big|\; \mathbf{w}^t\right]\\
\overset{(b)}{\le}& F(\mathbf{w}^t) - \frac{\eta}{2}\|\nabla F(\mathbf{w}^t)\|^2 + \frac{\eta}{2}\cdot\mathbb{E}\Big[\|\mathbf{G}^t - \nabla F(\mathbf{w}^t)\|^2\Big|\mathbf{w}^t\Big]\\
\overset{(c)}{\le}& F(\mathbf{w}^t) - \frac{\eta}{2}\|\nabla F(\mathbf{w}^t)\|^2 + \eta\cdot\mathbb{E}\Big[\|\mathbf{G}^t - \bar{\mathbf{G}}^t\|^2\Big|\mathbf{w}^t\Big] + \eta\cdot\mathbb{E}\Big[\|\bar{\mathbf{G}}^t - \nabla F(\mathbf{w}^t)\|^2\Big|\mathbf{w}^t\Big],
\end{aligned} \tag{42}$$

where (a) is derived based on Assumption 5, (b) is derived based on that $\eta \le \frac{1}{L}$, and (c) is derived based on that $\|\mathbf{x}+\mathbf{y}\|^2 \le 2\|\mathbf{x}\|^2 + 2\|\mathbf{y}\|^2$. Take total expectation on both sides and it is obtained that

$$\mathbb{E}[F(\mathbf{w}^{t+1})] \le \mathbb{E}[F(\mathbf{w}^t)] - \frac{\eta}{2}\mathbb{E}\left\|\nabla F(\mathbf{w}^t)\right\|^2 + \eta\cdot\mathbb{E}\left\|\mathbf{G}^t - \bar{\mathbf{G}}^t\right\|^2 + \eta\cdot\mathbb{E}\left\|\bar{\mathbf{G}}^t - \nabla F(\mathbf{w}^t)\right\|^2. \tag{43}$$

Since the fraction of Byzantine workers is not larger than $\frac{B\delta}{n}$, there are at most $B\delta$ Byzantine workers. Thus, there are at most $B\delta$ buffers that are not credible. It indicates that the fraction of credible buffers equals or is larger than $1-\delta$. Since $Aggr(\cdot)$ is a $(\delta_{\max}, c)$-robust aggregation function and $\mathbf{G}^t = Aggr([\mathbf{h}_0, \ldots, \mathbf{h}_{B-1}])$, we have

$$\begin{aligned}
\mathbb{E}\left\|\mathbf{G}^t - \bar{\mathbf{G}}^t\right\|^2 \le& c\delta\cdot\max_{b,b'\in H^t}\mathbb{E}\left\|\mathbf{h}_b - \mathbf{h}_{b'}\right\|^2\\
\le& c\delta\cdot\max_{b,b'\in H^t}\left(2\mathbb{E}\left\|\mathbf{h}_b - \nabla F(\mathbf{w}^t)\right\|^2 + 2\mathbb{E}\left\|\mathbf{h}_{b'} - \nabla F(\mathbf{w}^t)\right\|^2\right)\\
\le& 4c\delta\cdot\max_{b\in H^t}\mathbb{E}\left\|\mathbf{h}_b - \nabla F(\mathbf{w}^t)\right\|^2.
\end{aligned} \tag{44}$$

In addition, since $\bar{\mathbf{G}}^t = \frac{1}{|\mathcal{H}^t|}\sum_{b\in\mathcal{H}^t}\mathbf{h}_b$, we have

$$\mathbb{E}\|\bar{\mathbf{G}}^t - \nabla F(\mathbf{w}^t)\|^2 \le \frac{1}{|\mathcal{H}^t|}\sum_{b\in\mathcal{H}^t}\mathbb{E}\left\|\mathbf{h}_b - \nabla F(\mathbf{w}^t)\right\|^2 \le \max_{b\in H^t}\mathbb{E}\left\|\mathbf{h}_b - \nabla F(\mathbf{w}^t)\right\|^2. \tag{45}$$

Therefore,

$$\mathbb{E}[F(\mathbf{w}^{t+1})] \le \mathbb{E}[F(\mathbf{w}^t)] - \frac{\eta}{2}\mathbb{E}\|\nabla F(\mathbf{w}^t)\|^2 + \eta(4c\delta+1)\cdot\max_{b\in H^t}\mathbb{E}\left\|\mathbf{h}_b - \nabla F(\mathbf{w}^t)\right\|^2. \tag{46}$$

For any $b \in \mathcal{H}^t$, let $\mathbf{u}_l$ $(l = 1, 2, \ldots, N_b^t)$ denote the momentums received in $\mathbf{h}_b$.

$$\begin{aligned}
&\mathbb{E}\|\mathbf{h}_b - \nabla F(\mathbf{w}^t)\|^2\\
=&\mathbb{E}\left\|\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\mathbf{u}_l - \nabla F(\mathbf{w}^t)\right\|^2\\
=&\mathbb{E}\left\|\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\mathbf{u}_l - \mathbb{E}\left[\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\mathbf{u}_l\right]\right\|^2 + \left\|\mathbb{E}\left[\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\mathbf{u}_l\right] - \nabla F(\mathbf{w}^t)\right\|^2\\
\le&\mathbb{E}\left\|\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}(\mathbf{u}_l - \mathbb{E}[\mathbf{u}_l])\right\|^2 + \frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\left\|\mathbb{E}[\mathbf{u}_l] - \nabla F(\mathbf{w}^t)\right\|^2.
\end{aligned} \tag{47}$$

$\forall l \in \{1, 2, \dots, N_b^t\}$, suppose that $\mathbf{u}_l$ is the momentum received from worker_k for the v-th time. When $t > \tau_{max}$, we have $v > 1$. Thus, $\mathbf{u}_l = \mathbf{u}_{k,v} = (1-\lambda)\mathbf{u}_{k,v-1} + \lambda \nabla f(\mathbf{w}^{t_{v-1}}; z_{i_k})$ where $\mathbf{u}_{k,v-1}$ denotes the stored momentum on worker_k before $\nabla f(\mathbf{w}^{t_{v-1}}; z_{i_k})$ is computed and $t - \tau_{max} \le t_{v-1} \le t$. By using Assumption 4, we have

$$\mathbb{E}\|\mathbf{u}_l - \mathbb{E}[\mathbf{u}_l]\|^2 = \lambda^2 \cdot \mathbb{E}\left\|\nabla f(\mathbf{w}^{t_{v-1}}; z_{i_k}) - \mathbb{E}[\nabla f(\mathbf{w}^{t_{v-1}}; z_{i_k})]\right\|^2 \le \lambda^2\sigma^2. \tag{48}$$

When $0 \le t \le \tau_{max}$, it is uncertain whether $v = 1$ or not. If $v > 1$, it is already obtained that $\mathbb{E}\|\mathbf{u}_l - \mathbb{E}[\mathbf{u}_l]\|^2 \le \lambda^2\sigma^2 \le \sigma^2$. If $v = 1$, $\mathbf{u}_l = \mathbf{u}_{k,v} = \nabla f(\mathbf{w}^{t_{v-1}}; z_{i_k})$. Thus,

$$\mathbb{E}\|\mathbf{u}_l - \mathbb{E}[\mathbf{u}_l]\|^2 = \mathbb{E}\left\|\nabla f(\mathbf{w}^{t_{v-1}}; z_{i_k}) - \mathbb{E}[\nabla f(\mathbf{w}^{t_{v-1}}; z_{i_k})]\right\|^2 \le \sigma^2. \tag{49}$$

In summary, when $0 \le t \le \tau_{max}$, we have $\mathbb{E}\|\mathbf{u}_l - \mathbb{E}[\mathbf{u}_l]\|^2 \le \sigma^2$. Therefore,

$$\mathbb{E}\left\|\frac{1}{N_b^t}\sum_{l=1}^{N_b^t}(\mathbf{u}_l - \mathbb{E}[\mathbf{u}_l])\right\|^2 \le \frac{1}{N_b^t}\sum_{l=1}^{N_b^t}\mathbb{E}\|\mathbf{u}_l - \mathbb{E}[\mathbf{u}_l]\|^2 \le \begin{cases}\sigma^2, & \text{if } 0 \le t \le \tau_{max};\\ \lambda^2\sigma^2, & \text{if } t > \tau_{max}.\end{cases} \tag{50}$$

Meanwhile, by Lemma 19, we have

$$\left\|\mathbb{E}[\mathbf{u}_l - \nabla F(\mathbf{w}^t)]\right\|^2 \le 4(4c\delta+1)\left[4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2}\right]\eta^2L^2(\tau_{max}+1)^2D^2. \tag{51}$$

Consequently,

$$\begin{aligned}&\mathbb{E}\|\mathbf{h}_b - \nabla F(\mathbf{w}^t)\|^2\\ \le&\begin{cases}\sigma^2 + 4(4c\delta+1)(4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2})\eta^2L^2(\tau_{max}+1)^2D^2, & t \le \tau_{max};\\ \lambda^2\sigma^2 + 4(4c\delta+1)(4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2})\eta^2L^2(\tau_{max}+1)^2D^2, & t > \tau_{max}.\end{cases}\end{aligned} \tag{52}$$

Substituting it into (46), it is obtained that if $0 \le t \le \tau_{max}$,

$$\begin{aligned}\mathbb{E}[F(\mathbf{w}^{t+1})] \le& \mathbb{E}[F(\mathbf{w}^t)] - \frac{\eta}{2}\mathbb{E}\|\nabla F(\mathbf{w}^t)\|^2 + \eta(4c\delta+1)\sigma^2\\ &+ 4\eta(4c\delta+1)^2\left[4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2}\right]\eta^2L^2(\tau_{max}+1)^2D^2;\end{aligned} \tag{53}$$

and that if $t > \tau_{max}$,

$$\begin{aligned}\mathbb{E}[F(\mathbf{w}^{t+1})] \le& \mathbb{E}[F(\mathbf{w}^t)] - \frac{\eta}{2}\mathbb{E}\|\nabla F(\mathbf{w}^t)\|^2 + \eta(4c\delta+1)\lambda^2\sigma^2\\ &+ 4\eta(4c\delta+1)^2\left[4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2}\right]\eta^2L^2(\tau_{max}+1)^2D^2.\end{aligned} \tag{54}$$

By taking summation over t, it is obtained that when $T > \tau_{max}$,

$$\mathbb{E}[F(\mathbf{w}^T)] \le F(\mathbf{w}^0) - \frac{\eta}{2}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\mathbf{w}^t)\|^2 + \eta(4c\delta+1)\sigma^2(\tau_{max}+1+\lambda^2(T-\tau_{max}-1))$$

$$+ 4\eta T(4c\delta+1)^2 \left[4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2}\right]\eta^2 L^2(\tau_{max}+1)^2 D^2. \tag{55}$$

Finally, by using $\mathbb{E}[F(\mathbf{w}^T)] \geq F^*$ and $T-\tau_{max}-1 < T$, we have

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}||\nabla F(\mathbf{w}^t)||^2}{T} \leq \frac{2[F(\mathbf{w}^0)-F^*]}{\eta T} + \frac{2(4c\delta+1)(\tau_{max}+1)\sigma^2}{T} + \zeta, \tag{56}$$

where
$\zeta = 2(4c\delta+1)\lambda^2\sigma^2 + 8(4c\delta+1)^2\left[4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2}\right]\eta^2L^2(\tau_{max}+1)^2D^2$. ■

B.9 Proof of Proposition 16

Proof Substituting $\lambda = \sqrt{\eta L}$ and $\eta \leq \sqrt{\frac{F(\mathbf{w}^0)-F^*}{LT(4c\delta+1)[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]}}$ into (2), it is obtained that

$$\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}||\nabla F(\mathbf{w}^t)||^2}{T}\\
\leq\ & \frac{2[F(\mathbf{w}^0)-F^*]}{\eta T} + \frac{2(4c\delta+1)(\tau_{max}+1)\sigma^2}{T} + 2(4c\delta+1)\lambda^2\sigma^2\\
&+ 8(4c\delta+1)^2\left[4-\lambda+2\sqrt{4-2\lambda+4\lambda^{-2}}+2\lambda^{-2}\right]\eta^2L^2(\tau_{max}+1)^2D^2\\
\leq\ & \frac{2[F(\mathbf{w}^0)-F^*]}{\eta T} + \frac{2(4c\delta+1)(\tau_{max}+1)\sigma^2}{T} + 2(4c\delta+1)\lambda^2\sigma^2\\
&+ 8(4c\delta+1)^2\left(4+8\lambda^{-1}+2\lambda^{-2}\right)\eta^2L^2(\tau_{max}+1)^2D^2\\
=\ & \frac{2[F(\mathbf{w}^0)-F^*]}{\eta T} + 2(4c\delta+1)[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]\eta L + \frac{2(4c\delta+1)(\tau_{max}+1)\sigma^2}{T}\\
&+ 64(4c\delta+1)^2(\tau_{max}+1)^2D^2\eta^{\frac{3}{2}}L^{\frac{3}{2}} + 32(4c\delta+1)^2(\tau_{max}+1)^2D^2\eta^2L^2\\
\leq\ & \max\left(\frac{2L^{\frac{1}{2}}[F(\mathbf{w}^0)-F^*]^{\frac{1}{2}}(4c\delta+1)^{\frac{1}{2}}[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]^{\frac{1}{2}}}{T^{\frac{1}{2}}}, \frac{2L[F(\mathbf{w}^0)-F^*]}{T}\right)\\
&+ \frac{2L^{\frac{1}{2}}[F(\mathbf{w}^0)-F^*]^{\frac{1}{2}}(4c\delta+1)^{\frac{1}{2}}[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]^{\frac{1}{2}}}{T^{\frac{1}{2}}}\\
&+ \frac{2(4c\delta+1)(\tau_{max}+1)\sigma^2}{T}\\
&+ 64(4c\delta+1)^2(\tau_{max}+1)^2D^2\left(\frac{L[F(\mathbf{w}^0)-F^*]}{T(4c\delta+1)[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]}\right)^{\frac{3}{4}}\\
&+ 32(4c\delta+1)^2(\tau_{max}+1)^2D^2\left(\frac{L[F(\mathbf{w}^0)-F^*]}{T(4c\delta+1)[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]}\right)\\
\leq\ & \frac{4L^{\frac{1}{2}}[F(\mathbf{w}^0)-F^*]^{\frac{1}{2}}(4c\delta+1)^{\frac{1}{2}}[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2D^2]^{\frac{1}{2}}}{T^{\frac{1}{2}}}
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{L^{\frac{3}{4}}[F(\mathbf{w}^0) - F^*]^{\frac{3}{4}} \frac{64(4c\delta+1)^{\frac{5}{4}}(\tau_{max}+1)^2 D^2}{[\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2 D^2]^{\frac{3}{4}}}}{T^{\frac{3}{4}}} \\
&+ \frac{L[F(\mathbf{w}^0) - F^*]\left(2 + \frac{32(4c\delta+1)(\tau_{max}+1)^2 D^2}{\sigma^2+8(4c\delta+1)(\tau_{max}+1)^2 D^2}\right) + 2(4c\delta+1)(\tau_{max}+1)\sigma^2}{T} \\
\leq\ & \frac{4L^{\frac{1}{2}}[F(\mathbf{w}^0) - F^*]^{\frac{1}{2}}(4c\delta+1)^{\frac{1}{2}}[\sigma^2 + 8(4c\delta+1)(\tau_{max}+1)^2 D^2]^{\frac{1}{2}}}{T^{\frac{1}{2}}} \\
&+ \frac{14L^{\frac{3}{4}}[F(\mathbf{w}^0) - F^*]^{\frac{3}{4}}(4c\delta+1)^{\frac{1}{2}}(\tau_{max}+1)^{\frac{1}{2}} D^{\frac{1}{2}}}{T^{\frac{3}{4}}} \\
&+ \frac{6L[F(\mathbf{w}^0) - F^*] + 2(4c\delta+1)(\tau_{max}+1)\sigma^2}{T}.
\end{aligned}
$$

■

B.10 Relation between Geometric Median and Centered Clipping

Corollary 20. *Aggregation rule centered clipping (CC) is equivalent to geometric median (geoMed) when clipping size $R \to 0^+$.*

Proof The definition of CC is given by:

$$
\mathbf{h}^{l+1} = \mathbf{h}^l + \frac{1}{B}\sum_{b=0}^{B-1}(\mathbf{h}_b - \mathbf{h}^l)\min\left(1, \frac{R}{\|\mathbf{h}_b - \mathbf{h}^l\|_2}\right). \tag{57}
$$

When CC converges to $\mathbf{h}_{CC}^*$, it means that

$$
\mathbf{h}_{CC}^* = \mathbf{h}_{CC}^* + \frac{1}{B}\sum_{b=0}^{B-1}(\mathbf{h}_b - \mathbf{h}_{CC}^*)\min\left(1, \frac{R}{\|\mathbf{h}_b - \mathbf{h}_{CC}^*\|_2}\right). \tag{58}
$$

Thus, we have:

$$
\sum_{b=0}^{B-1}(\mathbf{h}_b - \mathbf{h}_{CC}^*)\min\left(1, \frac{R}{\|\mathbf{h}_b - \mathbf{h}_{CC}^*\|_2}\right) = \mathbf{0}. \tag{59}
$$

When $\forall b \in \{0, \ldots, B-1\},\ R \leq \|\mathbf{h}_b - \mathbf{h}_{CC}^*\|_2$ (since $R \to 0^+$), we have

$$
\min\left(1, \frac{R}{\|\mathbf{h}_b - \mathbf{h}_{CC}^*\|_2}\right) = \frac{R}{\|\mathbf{h}_b - \mathbf{h}_{CC}^*\|_2}. \tag{60}
$$

Therefore,

$$
R \cdot \sum_{b=0}^{B-1}\frac{(\mathbf{h}_b - \mathbf{h}_{CC}^*)}{\|\mathbf{h}_b - \mathbf{h}_{CC}^*\|_2} = \mathbf{0}. \tag{61}
$$

Namely,

$$
R \cdot \left[\nabla\left(\sum_{b=0}^{B-1}\|\mathbf{h} - \mathbf{h}_b\|_2\right)\right]\Bigg|_{\mathbf{h}=\mathbf{h}_{CC}^*} = \mathbf{0}. \tag{62}
$$

Considering that the function $\sum_{b=0}^{B-1} \|\mathbf{h} - \mathbf{h}_b\|_2$ is convex, we have:

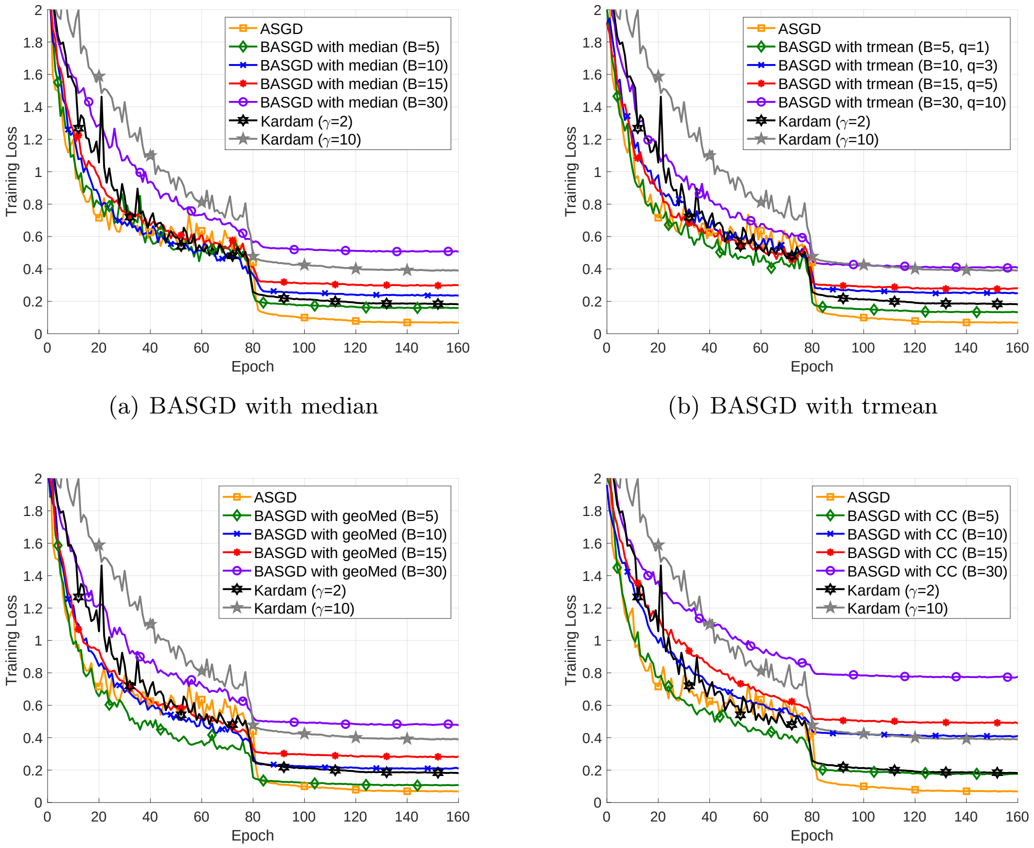
$$\mathbf{h}^*_{CC} = \arg\min_{\mathbf{h}\in\mathbb{R}^d} \left\{ \sum_{b=0}^{B-1} \|\mathbf{h} - \mathbf{h}_b\|_2 \right\} = \text{geoMed}([\mathbf{h}_0, \dots, \mathbf{h}_{B-1}]). \tag{63}$$

■

Meanwhile, we have to point out that although CC is theoretically equivalent to geoMed when R is small enough, R is not supposed to be set too small in practical applications. Too small R will slow the convergence rate of CC.

Appendix C. More Experimental Results

Figure 9-10, Figure 11, and Figure 12 illustrate the average training loss w.r.t. epochs when under no attack, non-omniscient attacks, and omniscient attacks in the image classification task. Please note that in Figure 11 and Figure 12, some curves do not appear because the value of the loss function is extremely large due to the Byzantine attack. γ is the hyper-parameter about the assumed number of Byzantine workers in Kardam. The experimental results further support the conclusions of this work.



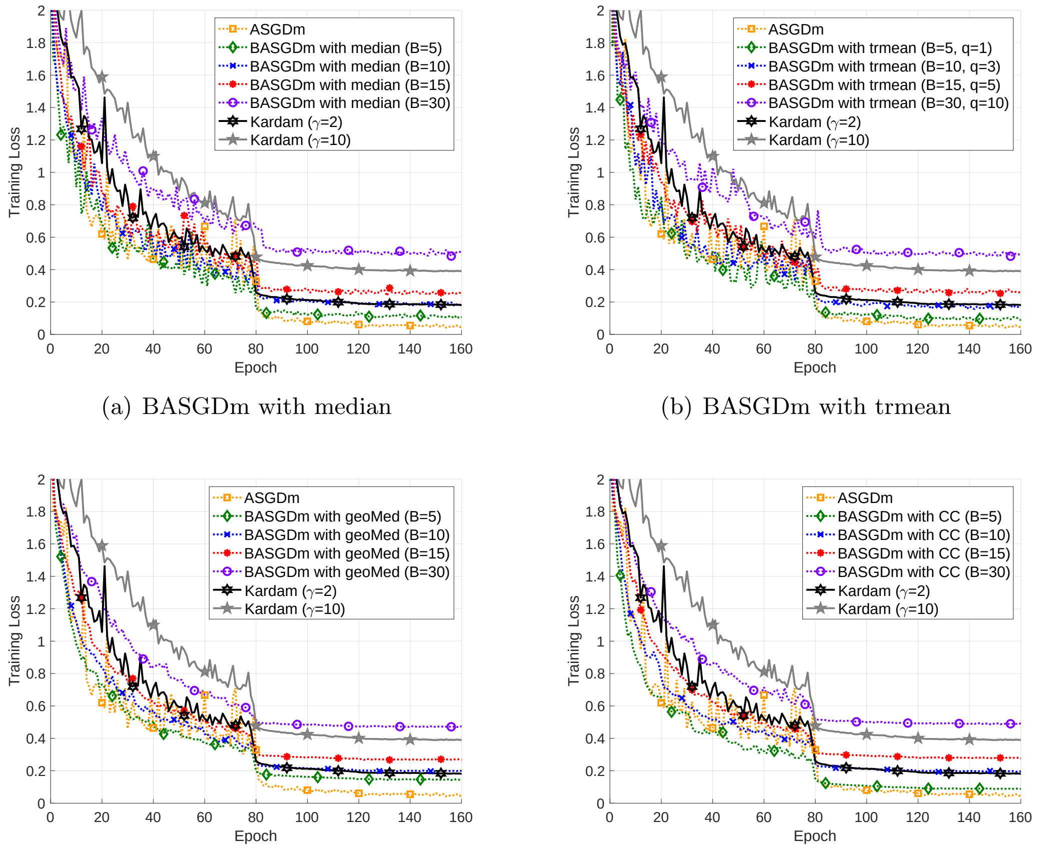
(a) BASGD with median

(b) BASGD with trmean

(c) BASGD with geoMed

(d) BASGD with CC

Figure 9: Average training loss w.r.t. epochs of methods BASGD, ASGD, and Kardam when there are no Byzantine workers.



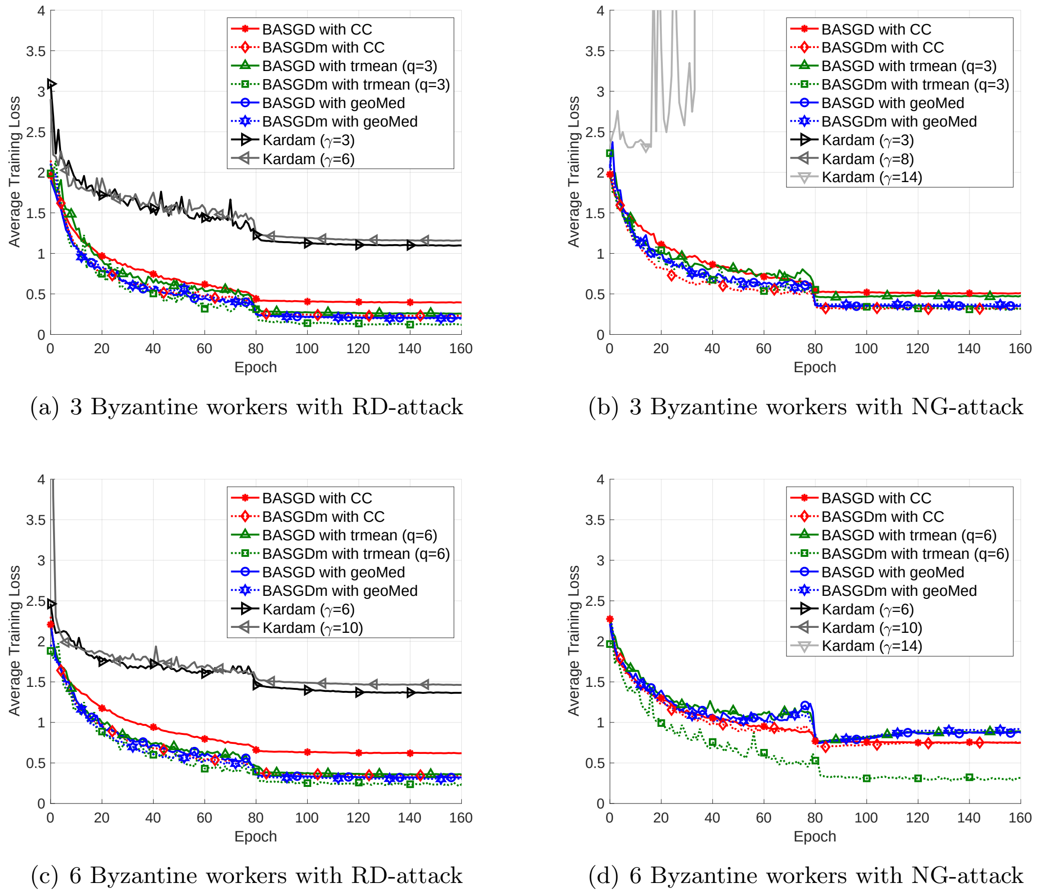
(a) BASGDm with median

(b) BASGDm with trmean

(c) BASGDm with geoMed

(d) BASGDm with CC

Figure 10: Average training loss w.r.t. epochs of methods BASGDm, ASGDm, and Kardam when there are no Byzantine workers.



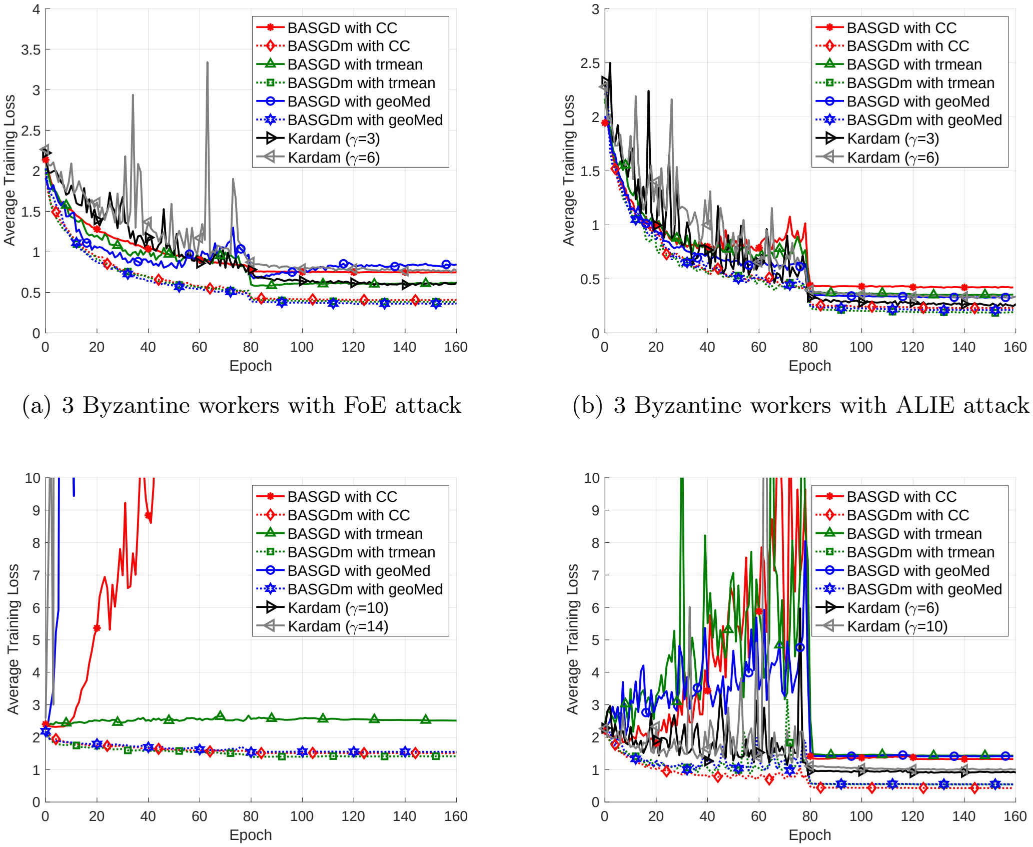
(a) 3 Byzantine workers with RD-attack

(b) 3 Byzantine workers with NG-attack

(c) 6 Byzantine workers with RD-attack

(d) 6 Byzantine workers with NG-attack

Figure 11: Average training loss w.r.t. epochs under non-omniscient attacks. $B = 10$ for BASGD (BASGDm) when there are 3 Byzantine workers and $B = 15$ for BASGD (BASGDm) when there are 6 Byzantine workers. Some curves do not appear in the figure, because the value of loss function is extremely large.



(a) 3 Byzantine workers with FoE attack

(b) 3 Byzantine workers with ALIE attack

(c) 6 Byzantine workers with FoE attack

(d) 6 Byzantine workers with ALIE attack

Figure 12: Average training loss w.r.t. epochs under omniscient attacks. $B = 10$ for BASGD (BASGDm) when there are 3 Byzantine workers and $B = 15$ for BASGD (BASGDm) when there are 6 Byzantine workers. Some curves do not appear in the figure, because the value of loss function is extremely large.

References

Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 4613–4623, 2018.

Zeyuan Allen-Zhu, Faeze Ebrahimian, Jerry Li, and Dan Alistarh. Byzantine-resilient non-convex stochastic gradient descent. *arXiv preprint arXiv:2012.14368*, 2020.

Mahmoud Assran, Arda Aytekin, Hamid Reza Feyzmahdavian, Mikael Johansson, and Michael G Rabbat. Advances in asynchronous parallel and distributed optimization. *Proceedings of the IEEE*, 108(11):2013–2031, 2020.

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, pages 8635–8645, 2019.

Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signSGD with majority vote is communication efficient and fault tolerant. In *Proceedings of the International Conference on Learning Representations*, 2019.

Dimitri Bertsekas, P Tsitsiklis, and N John. Parallel and distributed computation: Numeral methods. Prentice-Hall Inc., 1989.

Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, pages 119–129, 2017.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the International Conference on Computational Statistics*, pages 177–186. Springer, 2010.

Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Draco: Byzantine-resilient distributed training via redundant gradients. In *Proceedings of the International Conference on Machine Learning*, pages 903–912, 2018.

Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

Georgios Damaskinos, Rachid Guerraoui, Rhicheek Patra, Mahsa Taziki, et al. Asynchronous Byzantine machine learning (the case of SGD). In *Proceedings of the International Conference on Machine Learning*, pages 1145–1154, 2018.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

Ilias Diakonikolas and Daniel M Kane. Recent advances in algorithmic high-dimensional robust statistics. *arXiv preprint arXiv:1911.05911*, 2019.

Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *Proceedings of the International Conference on Machine Learning*, pages 999–1008, 2017.

Xiaomin Duan, Huafei Sun, and Linyu Peng. Application of gradient descent algorithms based on geodesic distances. *Science China Information Sciences*, 63:1–11, 2020.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

El-Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, Byzantine, heterogeneous, asynchronous and nonconvex learning). In *Advances in Neural Information Processing Systems*, pages 25044–25057, 2021a.

El-Mahdi El-Mhamdi, Rachid Guerraoui, and Sébastien Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *Proceedings of the International Conference on Learning Representations*, 2021b.

Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Trading redundancy for communication: Speeding up distributed SGD for non-convex optimization. In *Proceedings of the International Conference on Machine Learning*, pages 2545–2554, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Martin Jaggi, Virginia Smith, Martin Takác, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pages 3068–3076, 2014.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for Byzantine robust optimization. In *Proceedings of the International Conference on Machine Learning*, pages 5311–5319, 2021.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *Proceedings of the International Conference on Learning Representations*, 2022.

Jakub Konevcnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv:1610.05492*, 2016.

Konstantinos Konstantinidis and Aditya Ramamoorthy. Byzshield: An efficient and robust system for distributed training. *Proceedings of Machine Learning and Systems*, 3:812–828, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. In *Concurrency: the works of leslie lamport*, pages 203–226, 2019.

Jason D Lee, Qihang Lin, Tengyu Ma, and Tianbao Yang. Distributed stochastic variance reduced gradient methods by sampling extra data with replacement. *The Journal of Machine Learning Research*, 18(1):4404–4446, 2017.

Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1544–1551, 2019.

Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, pages 19–27, 2014.

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

Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method. In *Advances in Neural Information Processing Systems*, pages 3059–3067, 2014.

Ji Liu and Ce Zhang. Distributed learning systems with first-order methods. *arXiv preprint arXiv:2104.05245*, 2021.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael Jordan, Peter Richtárik, and Martin Takác. Adding vs. averaging in distributed primal-dual optimization. In *Proceedings of the International Conference on Machine Learning*, pages 1973–1982, 2015.

John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 3581–3607, 2022.

Matthew Nokleby, Haroon Raja, and Waheed U Bajwa. Scaling-up distributed processing of data streams for machine learning. *arXiv preprint arXiv:2005.08854*, 2020.

Jungwuk Park, Dong-Jun Han, Minseok Choi, and Jaekyun Moon. Sageflow: Robust federated learning against both stragglers and adversaries. In *Advances in Neural Information Processing Systems*, pages 840–851, 2021.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

Shashank Rajput, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Detox: A redundancy-based framework for faster and more robust gradient aggregation. In *Advances in Neural Information Processing Systems*, pages 10320–10330, 2019.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

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 *Annual Conference of the International Speech Communication Association*, 2014.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the International Conference on Machine Learning*, pages 1000–1008, 2014.

Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

Jy-yong Sohn, Dong-Jun Han, Beongjun Choi, and Jaekyun Moon. Election coding for distributed learning: Protecting signsgd against Byzantine attacks. In *Advances in Neural Information Processing Systems*, pages 14615–14625, 2020.

Shizhao Sun, Wei Chen, Jiang Bian, Xiaoguang Liu, and Tie-Yan Liu. Slim-dp: a multi-agent system for communication-efficient distributed deep learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 721–729, 2018.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Advances in Neural Information Processing Systems*, pages 16070–16084, 2020.

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.

Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B Giannakis. Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks. *IEEE Transactions on Signal Processing*, 68:4583–4596, 2020.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *Proceedings of the International Conference on Machine Learning*, pages 6893–6901, 2019.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking Byzantine-tolerant sgd by inner product manipulation. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 261–270, 2020a.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno++: Robust fully asynchronous SGD. In *Proceedings of the International Conference on Machine Learning*, pages 10495–10503, 2020b.

Tianbao Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pages 629–637, 2013.

Yi-Rui Yang and Wu-Jun Li. BASGD: Buffered asynchronous SGD for Byzantine learning. In *Proceedings of the International Conference on Machine Learning*, pages 11751–11761, 2021.

Zhixiong Yang, Arpita Gang, and Waheed U Bajwa. Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the Byzantine threat model. *IEEE Signal Processing Magazine*, 37(3):146–159, 2020.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the International Conference on Machine Learning*, pages 5650–5659, 2018.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Defending against saddle point attack in Byzantine-robust distributed learning. In *Proceedings of the International Conference on Machine Learning*, pages 7074–7084, 2019.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *Proceedings of the International Conference on Machine Learning*, pages 7184–7193, 2019a.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5693–5700, 2019b.

Lijun Zhang, Mehrdad Mahdavi, and Rong Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, pages 980–988, 2013.

Ruiliang Zhang and James Kwok. Asynchronous distributed admm for consensus optimization. In *Proceedings of the International Conference on Machine Learning*, pages 1701–1709, 2014.

Shen-Yi Zhao, Ru Xiang, Ying-Hao Shi, Peng Gao, and Wu-Jun Li. SCOPE: scalable composite optimization for learning on spark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2928–2934, 2017.

Shen-Yi Zhao, Gong-Duo Zhang, Ming-Wei Li, and Wu-Jun Li. Proximal SCOPE for distributed sparse learning. In *Advances in Neural Information Processing Systems*, pages 6551–6560, 2018.

Shen-Yi Zhao, Yin-Peng Xie, and Wu-Jun Li. On the convergence and improvement of stochastic normalized gradient descent. *Science China Information Sciences*, 64:1–13, 2021.

Shuxin Zheng, Qi Meng, Taifeng Wang, Wei Chen, Nenghai Yu, Zhi-Ming Ma, and Tie-Yan Liu. Asynchronous stochastic gradient descent with delay compensation. In *Proceedings of the International Conference on Machine Learning*, pages 4120–4129, 2017.

Yi Zhou, Yingbin Liang, Yaoliang Yu, Wei Dai, and Eric P Xing. Distributed proximal gradient algorithm for partially asynchronous computer clusters. *The Journal of Machine Learning Research*, 19(1):733–764, 2018.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2010.