Byzantine Fault-Tolerant Distributed Machine Learning using D-SGD and Norm-Based Comparative Gradient Elimination (CGE)

Nirupam Gupta
EPFL
Lausanne, Switzerland
nirupam115@gmail.com

Shuo Liu
Georgetown University
Washington, D.C., USA
sl1539@georgetown.edu

Nitin Vaidya
Georgetown University
Washington, D.C., USA
nv198@georgetown.edu

***Abstract*—This paper considers the Byzantine fault-tolerance problem in distributed stochastic gradient descent (D-SGD) method – a popular algorithm for distributed multi-agent machine learning. In this problem, each agent samples data points independently from a certain data-generating distribution. In the fault-free case, the D-SGD method allows all the agents to learn a mathematical model best fitting the data collectively sampled by all agents. We consider the case when a fraction of agents may be Byzantine faulty. Such faulty agents may not follow a prescribed algorithm correctly, and may render traditional D-SGD method ineffective by sharing arbitrary incorrect stochastic gradients. We propose a norm-based gradient-filter, named comparative gradient elimination (CGE), that robustifies the D-SGD method against Byzantine agents. We show that the CGE gradient-filter guarantees fault-tolerance against a bounded fraction of Byzantine agents under standard stochastic assumptions, and is *computationally simpler* compared to many existing gradient-filters such as multi-KRUM, geometric median-of-means, and the spectral filters. We empirically show, by simulating distributed learning on neural networks, that the fault-tolerance of CGE is comparable to that of existing gradient-filters. We also empirically show that *exponential averaging* of stochastic gradients improves the fault-tolerance of a generic gradient-filter.**

I. Introduction

The problem of distributed multi-agent learning or *federated* learning has gained significant attention in recent years [8], [15], [29], [33]. In this problem, there are multiple machines or *agents* in the system each sampling data points locally and independently. The goal is to design *distributed algorithms* that allow the agents to compute or *learn* a common mathematical model that optimally fits the data points collectively sampled by all the agents. Most prior works in distributed learning consider a fault-free setting wherein all the agents are free from faults and follow a prescribed algorithm honestly. However, in practical distributed systems, some agents may be faulty [1], [5], [11], [13], [17], [31], [32].

We consider a system with n agents where up to f agents are Byzantine faulty. The identity of the Byzantine agents is a priori unknown. Byzantine agents may collude and share arbitrary incorrect information with other non-faulty agents [21]. For instance, Byzantine agents may share information corresponding to *poisonous data points*; see [30] and references therein. In the presence of such faulty agents, a reasonable goal is to design a distributed algorithm that allows all the *non-faulty* agents to learn a mathematical model that optimally fits the data points only sampled by the non-faulty agents. A standard formulation of fault-tolerance in distributed learning is presented below.

Fault-tolerance in distributed learning: Each non-faulty agent samples data points independently and identically from a true data-generating distribution $\mathcal{D}$ over the m-dimensional real vector space $\mathbb{R}^m$. The non-faulty agents a priori fix a learning model Π, e.g., a neural network [7], characterized by d real-valued parameters compactly denoted by a vector $w \in \mathbb{R}^d$. For a given parameter vector w, each data point $z \in \mathbb{R}^m$ incurs a *loss* defined by a real-valued *loss function* $\ell : (w,\, z) \mapsto \mathbb{R}$. We define the *non-faulty expected loss function*:

$$Q(w) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}} \ell(w,\, z), \quad \forall w \in \mathbb{R}^d. \tag{1}$$

The goal of a Byzantine fault-tolerant distributed learning algorithm is to allow the non-faulty agents to compute an optimal learning parameter w^* that minimizes $Q(w)$, despite the presence of Byzantine faulty agents.

System architecture: We consider a synchronous server-based architecture shown in Fig. 1. The server is assumed trustworthy, but up to f agents may be Byzantine faulty. The trusted server helps solve the distributed learning problem in coordination with the agents.

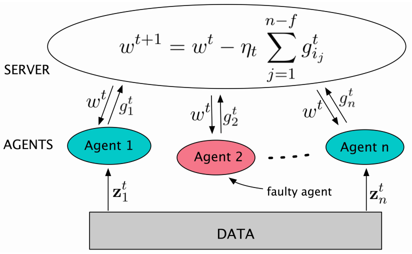


Fig. 1: *The system architecture.*

Research reported in this paper was sponsored in part by the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196, and by Fritz Fellowship from Georgetown University.

Fault-tolerance in the distributed stochastic gradient descent (D-SGD) method: We propose a fault-tolerance mechanism that confers fault-tolerance to the D-SGD method – a standard distributed machine learning algorithm [7]. The D-SGD method is an iterative algorithm wherein the server maintains an estimate of an optimal learning parameter, which is updated iteratively using *stochastic* gradients computed by the different agents using i.i.d. data points sampled from $\mathcal{D}$, as shown in Figure 1. In the fault-free setting, the D-SGD method converges to an optimal learning parameter, if the stochastic gradients have bounded variances [7]. However, the traditional D-SGD method is rendered ineffective in the presence of Byzantine faulty agents that may send arbitrary incorrect stochastic gradients to the server [5], [11], [17].

Our proposed fault-tolerance mechanism relies on a norm-based *gradient-filter*, named *comparative gradient elimination* (CGE), that mitigates the detrimental impact of arbitrary incorrect stochastic gradients to the correctness of the D-SGD method. In our CGE gradient-filter, in each iteration, the server eliminates f (out of n) received stochastic gradients with f largest Euclidean norms. The current estimate in each iteration is updated using the average of the $n - f$ remaining stochastic gradients. Details of our algorithm, and its fault-tolerance property are presented later in Section II. Please refer Figure 1 for an illustration.

In contrast to the other previously introduced applications of norm-based gradient elimination solving other unrelated problems, e.g., [25], [28], our CGE gradient-filter employs an *adaptive* threshold. Specifically, in our case, the threshold for eliminating stochastic gradients is not a constant but varies depending upon the norms of the non-faulty agents' stochastic gradients. This difference is critical for the fault-tolerance property of the CGE gradient-filter.

We also incorporate an *exponential averaging* scheme where the server uses exponentially weighted averages of the agents' stochastic gradients, instead of their instantaneous values in the D-SGD method. We empirically show in Section III-B that exponential averaging can notably improve the performance of a fault-tolerance mechanism such as the CGE gradient-filter.

A. Summary of our contributions

We summarize below our main contributions and results.

- **A new algorithm and its fault-tolerance property:** We show that our algorithm, D-SGD method with CGE gradient-filter, guarantees Byzantine fault-tolerance under standard assumptions [7]. Our result is summarized below, and presented formally in Section II-A. Notation Pr denotes probability.

  Theorem (Informal)**.** *Suppose that the variance σ^2 of the non-faulty stochastic gradients is bounded, and the loss function $Q(w)$ is λ-strongly convex with μ-Lipschitz continuous gradients. Let w^* denote a minimum of $Q(w)$ and w^t denote an estimate of w^* at the server in the t-th iteration of our algorithm. If f/n is less than $\lambda/(2\lambda+\mu)$ then $\exists \rho \in (0,\, 1)$ and $M \in \Theta(\sigma)$ such that, $\forall \epsilon > 0$,*

  $$\lim_{t \to \infty} Pr\left(\left\|w^t - w^*\right\|^2 \leq \epsilon\right) \geq 1 - \frac{1}{\epsilon}\,\left(\frac{\mathsf{M}^2}{1-\rho}\right).$$

  Our algorithm is also effective in the heterogeneous data setting, where different agents may have different data distribution, provided that the agents' respective loss functions have some minimal redundancy, see [18], [19], [23]. For simplicity, we only consider the homogeneous data setting in this paper.

- **Exponential averaging for improved fault-tolerance:** We empirically show (for the first time) the efficacy of *exponential averaging* of stochastic gradients for improving Byzantine fault-tolerance of a gradient-filter.

- **Empirical results:** We empirically show in Section III the efficacy of our algorithm for distributed learning on neural networks. We conduct experiments on a benchmark classification task, MNIST [6], under varied fault types and fraction of faulty agents. We show that the fault-tolerance of CGE gradient-filter is *comparable* to the existing gradient-filters, namely multi-KRUM, geometric median-of-means, and coordinate-wise trimmed mean.

B. Comparisons with Related Work

We present below key comparisons between CGE and other existing gradient-filters for fault-tolerance in D-SGD.

- **Computational simplicity:** The computational time complexity of CGE gradient-filter compares favourably to prominent gradient-filters, namely *multi-KRUM* [5], *geometric median-of-means* [11], and the *spectral gradient-filters* [10], [13], [14], [27]. In particular, the complexity of CGE gradient-filter is $\mathcal{O}(n(\log n + d))$, in comparison to $\mathcal{O}(n(n + d))$ of multi-KRUM and geometric median-of-means, and $\mathcal{O}(nd \min\{n,\, d\})$ of the spectral gradient-filters. We empirically show that despite its computational simplicity, the fault-tolerance of the CGE gradient-filter is comparable to these aforementioned gradient-filters.

- **Standard stochastic assumptions:** The coordinate-wise trimmed mean filter [35], the norm-based filter [16], and the signSGD filter [3] have similar time complexity as CGE. However, the known fault-tolerance guarantees of these gradient-filters rely on strong assumptions about the non-faulty stochastic gradients that are uncommon in many learning problems [7]. In particular, [16], [35] assume non-faulty stochastic gradients to be *sub-exponential* random variables, and [3] assumes non-faulty stochastic gradients to have a *unimodal symmetric* probability distributions. We, however, only assume the non-faulty stochastic gradients to have bounded variances – an assumption that is common in all the prior works, and is required for the convergence of the D-SGD method even in the fault-free case [6], [7].

Other related works [22], [34] consider the problem of Byzantine fault-tolerance in distributed learning using the *consensus optimization* methods [8], [24].

For obtaining the formal fault-tolerance property of our algorithm we assume the expected loss function $Q(w)$ to be strongly convex, unlike some of the aforementioned prior works. However, in learning problems when $Q(w)$ is convex, e.g., support vector machine and logistic regression, it is often regularized to a strongly convex function to mitigate overfitting [7]. In many learning problems even if $Q(w)$ is not *globally* strongly convex, it is so in a neighborhood of local minimizers, thus our result can show the convergence of the D-SGD method with CGE gradient-filter in such regions of the search space.

II. Proposed Algorithm

In this section, we present our CGE gradient-filter for tolerating Byzantine faulty agents in distributed learning using the D-SGD method. The description of the algorithm below is followed by its fault-tolerance guarantee in Section II-A.

Algorithm 1: D-SGD with CGE Gradient-filter

The server chooses the initial estimate w^0 arbitrarily from $\mathbb{R}^d$. Steps executed in each iteration t:

◇ **Step S1**: The server broadcasts the current estimate w^t to all the agents.

Each non-faulty agent i sends to the server a stochastic gradient of the global expected loss function $Q(w)$ at w^t, i.e., a noisy estimator of the gradient $\nabla Q(w^t)$. A faulty agent may send an incorrect arbitrary vector for its stochastic gradient.

Let g^t_i denote the gradient received by the server from agent i. If no gradient is from some agent i, i must be faulty (since the system is assumed synchronous), and the server eliminates i from the system.

◇ **Step S2 (CGE gradient-filter):** The server sorts the n received gradients as follows:

$$\left\|g^t_{i_1}\right\| \leq \dots \leq \left\|g^t_{i_{n-f}}\right\| \leq \left\|g^t_{i_{n-f+1}}\right\| \leq \dots \leq \left\|g^t_{i_n}\right\|. \tag{2}$$

where $g^t_{i_j}$, with j-th smallest norm, is from agent i_j. The server updates its current estimate using only $n-f$ stochastic gradients with smallest $n-f$ norms:

$$w^{t+1} = w^t - \eta_t \sum_{j=1}^{n-f} g^t_{i_j} \tag{3}$$

where η_t is the *step-size* of iteration t.

Similar to the traditional D-SGD, the server maintains an estimate of an optimal learning parameter which is updated in each iteration using Algorithm 1. For each iteration $t \in \{0, 1, \ldots\}$, let w^t denote the estimate of the server. In **Step S1**, the server obtains from the agents their locally computed *stochastic gradients* of the expected loss function $Q(w)$ at w^t. There are multiple methods for computing stochastic gradients [7, Section 5], one of which is described below. Note that a Byzantine faulty agent may send an arbitrary vector. In Step S2, to mitigate the detrimental impact of incorrect stochastic gradients, the algorithm uses a filter to "robustify" the gradient aggregation used for computing the updated estimate w^{t+1}. In particular, the server eliminates the stochastic gradients with the largest f Euclidean norms, and uses the aggregate of the remaining $n-f$ stochastic gradients with $n-f$ smallest Euclidean norms to compute w^{t+1}, as shown in Equation (3) below. We refer to the method used in **Step S2** for elimination the largest f gradients as *Comparative Gradient Elimination* (CGE) gradient-filter, since the norms of the gradients are compared together to eliminate (or filter out) the gradients with the largest f norms.

To compute a stochastic gradient in an iteration t, a non-faulty agent i samples k i.i.d. data points $z^t_{i_1}, \ldots, z^t_{i_k}$ from the distribution $\mathcal{D}$. Then, for each non-faulty agent i,

$$g^t_i = \frac{1}{k} \sum_{j=1}^{k} \nabla \ell(w^t,\, z^t_{i_j}). \tag{4}$$

k is referred as the *data batch-size* or simply batch-size.

Complexity: Note that computing the Euclidean norms of n d-dimensional vector takes $\mathcal{O}(nd)$ time, and sorting of n values takes $\mathcal{O}(n \log n)$ time. Thus, the time complexity of the CGE gradient-filter is $\mathcal{O}(n(d + \log n))$ in each iteration.

A. Fault-Tolerance Property

In this section, we present a formal convergence result for our algorithm under standard assumptions that hold true in most learning problems [7].

For each non-faulty agent i, let

$$\mathbf{z}^t_i = \{z^t_{i_1}, \ldots, z^t_{i_k}\} \tag{5}$$

denote the collection of k i.i.d. data points sampled by the agent i in iteration t. Now, for each agent i and iteration t we define a random variable

$$\zeta^t_i = \begin{cases} \mathbf{z}^t_i & , \quad \text{agent } i \text{ is non-faulty} \\ g^t_i & , \quad \text{agent } i \text{ is faulty} \end{cases} \tag{6}$$

Recall that g^t_i may be an arbitrary d-dimensional random variable for each Byzantine faulty agent i. For each iteration t, let

$$\zeta^t = \{\zeta^t_i,\ i = 1, \ldots, n\}, \tag{7}$$

and let $\mathbb{E}_t$ denote the expectation of a function of the collective random variables $\zeta^0, \ldots, \zeta^t$, given the initial estimate w^0. Specifically,

$$\mathbb{E}_t(\cdot) = \mathbb{E}_{\zeta^0, \ldots, \zeta^t}(\cdot), \quad \forall t \geq 0. \tag{8}$$

We make the following assumptions that are satisfies in many machine learning problems [7], [35].

Assumption 1 (Bounded variance)**.** *Assume that there exists a finite real value σ such that for all non-faulty agent i,*

$$\mathbb{E}_{\zeta_i^t} \left\| g_i^t - \mathbb{E}_{\zeta_i^t}\left(g_i^t\right) \right\|^2 \leq \sigma^2, \quad \forall t.$$

Assumption 2 (Lipschitz smoothness)**.** *Assume that there exists a finite positive real value μ such that*

$$\left\| \nabla Q(w) - \nabla Q(w') \right\| \leq \mu \left\| w - w' \right\|, \quad \forall w,\, w' \in \mathbb{R}^d.$$

Assumption 3 (Strong convexity)**.** *Assume that there exists a finite positive real value λ such that for all $w,\, w' \in \mathbb{R}^d$,*

$$\left\langle w - w',\, \nabla Q(w) - \nabla Q(w') \right\rangle \geq \lambda \left\| w - w' \right\|^2.$$

We define a *fault-tolerance margin*

$$\alpha = \frac{\lambda}{2\lambda + \mu} - \frac{f}{n} \tag{9}$$

that determines the maximum fraction of faulty agents f/n tolerable by our algorithm. Lastly, we define an upper bound for the step-size η_t in (3),

$$\overline{\eta} = \left(\frac{2(2\lambda + \mu)n}{n^2 + (n-f)^2\mu^2} \right) \alpha. \tag{10}$$

Theorem 1 below presents a key fault-tolerance property of our algorithm, i.e., Algorithm 1. Recall that w^* denotes a minimum of the global expected loss function $Q(w)$, i.e.,

$$w^* \in \arg\min_{w \in \mathbb{R}^d} Q(w).$$

Theorem 1. *Consider Algorithm 1. Suppose that the Assumptions 1, 2, and 3 hold true. If the fault-tolerance margin α is positive, and in the update law* (3) *the step-size $\eta_t = \eta \in (0,\ \overline{\eta})$ for all t then*

$$\rho = 1 - \left(n^2 + (n-f)^2\mu^2\right) \eta \left(\overline{\eta} - \eta\right) \in (0,\ 1), \tag{11}$$

and, for all $t \geq 1$,

$$\mathbb{E}_{t-1} \left\| w^t - w^* \right\|^2 \leq \rho^t \left\| w^0 - w^* \right\|^2 + \left(\frac{1 - \rho^t}{1 - \rho} \right) \mathsf{M}^2 \tag{12}$$

where

$$\mathsf{M}^2 = \left(\frac{f^2 \left(1 + \sqrt{n - f - 1}\right)^2}{n^2} + \eta^2 (n-f)^2 \right) \sigma^2. \tag{13}$$

According to Theorem 1, if $\alpha > 0$, i.e.,

$$\frac{f}{n} < \frac{\lambda}{2\lambda + \mu}, \tag{14}$$

then for small enough step-size, in (3), our algorithm converges *linearly* to a neighborhood of a minimum of the global expected loss function (1). As $\rho < 1$, (12) implies that

$$\lim_{t \to \infty} \mathbb{E}_{t-1} \left\| w^t - w^* \right\|^2 \leq \frac{\mathsf{M}^2}{1 - \rho}.$$

Upon using the Markov's inequality, we obtain the probabilistic guarantee on training accuracy stated in Section I-A.

III. Experiments

In this section, we present our key empirical results on fault-tolerance in distributed learning on neural networks using the D-SGD method with different gradient-filters. The fault-tolerance of different gradient-filters is evaluated through multiple experiments with varied fractions of faulty agents f/n, different types of faults and the data batch-size k.

We use multiple threads to simulate the distributed server-based system (ref. Fig. 1), one for the server and others for agents. The inter-thread communication is handled through message passing interface. The simulator is built in Python using PyTorch [26] and MPI4py [12], deployed on a Google Cloud Platform cluster with 64 vCPUs and 100 GB memory.

We experiment on the dataset MNIST [6], an image-classification dataset of handwritten digits comprising $60,000$ training and $10,000$ testing samples. We use a state-of-the-art neural network LeNet with of $431,080$ learning parameters. Thus, the value of dimension $d = 431,080$.

We simulate a distributed system with $n = 40$ agents, among which f faulty agents are chosen randomly. The server initiates the D-SGD method by choosing the initial estimate w^0, a d-dimensional vector, by uniform distributions near 0. The step-size is $\eta_t = 0.01$ in every iteration t. To tolerate faulty agents, the server uses a gradient-filter as shown in Step S2 of Algorithm 1. We compare the fault-tolerance of our CGE gradient-filter with the following prominent gradient-filters.

- Geometric median (GeoMed);
- Geometric median-of-means (MoM) [11], with group size $b = 2$;
- Coordinate-wise trimmed mean (CWTM) [35];
- Multi-KRUM [5], with $m = 5$.

Types of Faults: We simulate faulty agents that can exhibit two different types of faults listed below. The second one simulates inadvertently faulty agents that exhibit faulty behaviors due to hardware failures [20].

- **Gradient-reverse fault**: A faulty agent sends to the server a vector directed opposite to its correct stochastic gradient with the same norm. Specifically, if s_i^t denotes a correct stochastic gradient of faulty agent i in iteration t then agent i sends to the server a vector $g_i^t = -s_i^t$.
- **Label-flipping fault**: A faulty agent sends incorrect stochastic gradients due to erroneous output labels of its data points. Specifically, in our experiments with 10 different labels in MNIST, the original label of a data point y sampled by a faulty agent is changed to $\widetilde{y} = 9 - y$.

A. Results and analysis

We now present our experimental results with the above setup under various settings specified below. We fix the number of faulty agents $f = 8$ out of $n = 40$ total agents, and the data batch-size $k = 64$. We do multiple experimental runs of the D-SGD method with different gradient-filters under different types of faults. The identity of the faulty agents is fixed across all the experiments. We also fix the

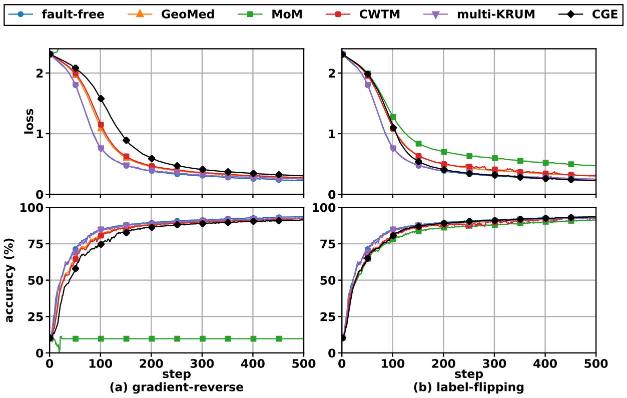


Fig. 2: *Distributed learning of* LeNet *for MNIST using the D-SGD method with different gradient-filters (represented by different colors) to tolerate $f = 8$ faulty agents in the system. The training losses and the testing accuracies evaluated by the server after* 0 *to* 500 *iterations (or steps) of the different learning algorithms are plotted in the first and the second rows, respectively. Different columns contain the results for different types of faults simulated by the faulty agents. Here, the data batch-size $k = 64$.*

TABLE I: Running time for distributed learning of LeNet for MNIST using D-SGD method with batch-size $k = 64$ and the different gradient-filters in presence of $f = 8$ faulty agents exhibiting *label-flipping* type of fault, corresponding to the training process shown in Fig. 2(b). Total running time refers to the time for 500 iterations.

Filter	**Time per iteration** (s)	**Total running time** (s)
GeoMed	2.473	1236.5
MoM	1.150	575.1
CWTM	0.903	451.8
Multi-KRUM	2.225	1112.4
CGE	**0.573**	**286.3**

random seeds used by agents for sampling data in training phase, so that across different experiments the same agent samples the same mini-batch of data for the same iteration. From our experiments, we observe that other than median-of-means, the three gradient-filters, including CGE, have comparable fault-tolerance against the two types of faults exhibited by the faulty agents. Median-of-means cannot tolerate *gradient-reverse* faults, and exhibits a performance gap between other filters when facing *label-flipping* faults. The plots for the losses and accuracies versus the number of iterations (or steps) are shown in Fig. 2 for the different experiments. Table I shows the running time for each filter under *label-flipping* faults. CGE has significantly smaller running time while preserving comparable performance.

To evaluate the influence of individual agents' data batch-size on the fault-tolerance by CGE gradient-filter, we conduct experiments with four different batch-sizes: $k = 32$, 64, 128, and 256. For these experiments all faulty agents exhibit *label-flipping* faults. The average training losses and testing accuracies between the 475-th and 500-th iterations (or steps) for different batch-sizes are shown in Fig. 3 where different colors represent different numbers f of faulty agents. A larger batch-size results in stochastic gradients with smaller variances (see (4) in Section II), and thus, as expected from our theoretical results in Section II-A, the fault-tolerance of CGE gradient-filter improves with increased batch-size. However, an increase in batch-size also increases the costs for computing the stochastic gradients in each iteration. We later present a *exponential averaging* technique that reduces the variances of the stochastic gradients, and improves the fault-tolerance of the CGE gradient-filter, without increasing batch-sizes or the costs for computing stochastic gradients.

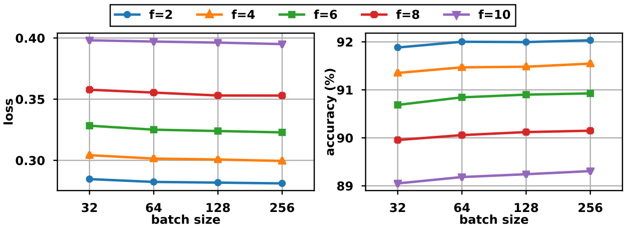


Fig. 3: *Average training losses and testing accuracies evaluated between* 475 *to* 500 *steps of the D-SGD method with CGE gradient-filter for distributed learning of* LeNet *in the presence of different number of faulty agents f (represented by different colors) each exhibiting the* gradient-reverse *Byzantine fault, for different data batch-sizes.*

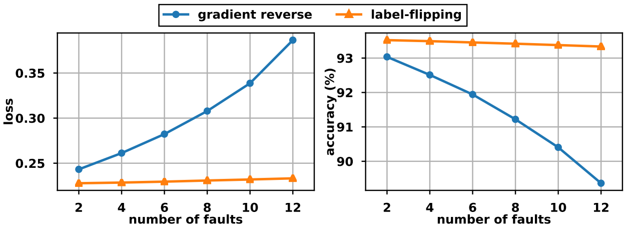
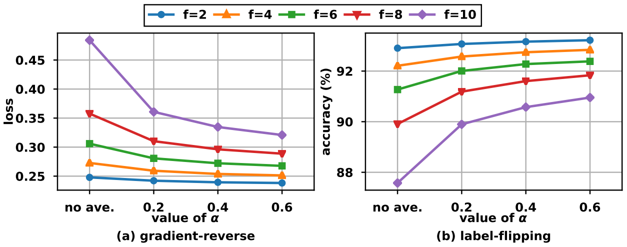


Fig. 4: *Average training losses and testing accuracies evaluated between* 475 *to* 500 *steps of the D-SGD method with CGE gradient-filter for distributed learning of* LeNet *in the presence of varied number of faulty agents f. Different colors represent different types of faults. The data batch-size $k = 64$.*

Lastly, to evaluate the effect of the fraction of faulty agents f/n on the fault-tolerance of CGE gradient-filter we conduct experiments for different values of f. In these experiments we set batch-size $k = 64$. The trend of average losses and accuracies observed between the 475-th to 500-th iterations (or steps) is shown in Fig. 4. As expected from our theoretical results in Section II-A, the fault-tolerance of the CGE gradient-filter deteriorates with increase in the fraction of faulty agents.

B. Exponential averaging of stochastic gradients

We observe from our experiments above, specifically plots in Fig. 3, that fault-tolerance of CGE gradient-filter improves with increase in batch-size k. The reason why this happens is the fact that larger batch-size results in stochastic gradients with smaller variances σ^2, which, owing to our results in Section II-A, results in improved fault-tolerance. However, increase in batch-size also increases the cost of computing stochastic gradients in each iteration. Motivated from this observation, we propose a more economical technique, *exponential averaging*, allowing non-faulty agents to compute stochastic gradients with reduced variances without increasing batch-size and their cost for computing stochastic gradients. Alternately, we may also use other existing variance reduction techniques from the stochastic optimization literature [7].



(a) gradient-reverse

(b) label-flipping

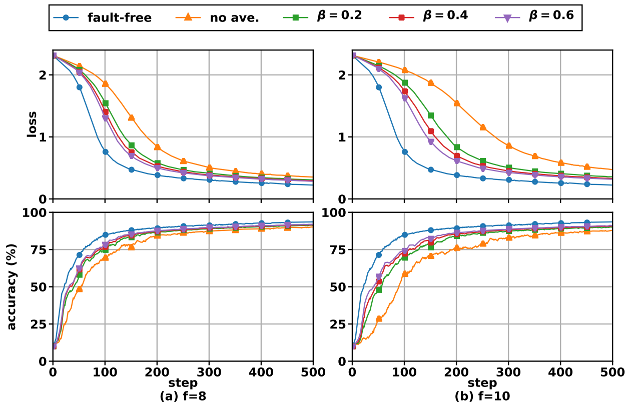
Fig. 5: *Average training losses and testing accuracies evaluated between* 475 *to* 500 *steps of the D-SGD method with CGE gradient-filter for distributed learning of* LeNet *in the presence of different number of faulty agents f (represented by different colors) when applying* exponential averaging *of stochastic gradients for different values of β.* no ave. *indicates the case when no averaging is not used. Here, the data batch-size $k = 64$, and the fault type is* norm-confusing.

For each iteration t and agent i, the server maintains an **exponentially weighted average** h_i^t of the stochastic gradients received from agent i so far. Specifically, for $\beta \in [0, 1)$,

$$h_i^t = \beta\, h_i^{t-1} + (1-\beta)\, g_i^t. \tag{15}$$

where g_i^t denotes the stochastic gradient received by the server from agent i in iteration t. Given a set of n vectors $y_1, \ldots, y_n$, let $\mathsf{CGE}_f\{y_1, \ldots, y_n\}$ denote the output of our CGE gradient-filter defined in step S2 of Algorithm 1. For each iteration t, in step S2 of Algorithm 1 the server updates the current estimate w^t to $w^{t+1} = w^t + \eta_t \cdot \mathsf{CGE}_f\{h_1^t, \ldots, h_n^t\}$. It should be noted that the above averaging scheme does not increase the per iteration computation cost for an individual agent, unlike the case when we increase the data batch-size.

To evaluate the impact of the above exponential averaging on the fault-tolerance of CGE gradient-filter, we introduce a new type of fault designed deliberately against our CGE filter:



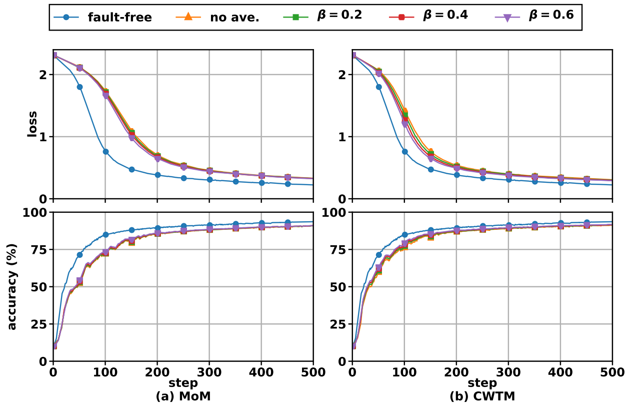
(a) f=8

(b) f=10

Fig. 6: *Distributed learning of* LeNet *for MNIST using the D-SGD method with batch-size $k = 64$ and CGE gradient-filter in the presence of $f = 8$ and* 10 *faulty agents exhibiting* norm-confusing *type of faults.* Exponential averaging *of stochastic gradients applied with different values of β (represented using different colors).*

TABLE II: Running time for distributed learning of LeNet for MNIST using D-SGD method with batch-size $k = 64$ and the different gradient-filters in presence of $f = 8$ faulty agents exhibiting *norm-confusing* type of fault with or without applying *exponential averaging* for different values of β, corresponding to the training process shown in Fig. 6(a). Total running time refers to the time for 500 iterations.

Filter	**Time per iteration** (s)	**Total running time** (s)
CGE (No ave.)	0.613	306.9
CGE ($\beta = 0.2$)	0.733	366.7
CGE ($\beta = 0.4$)	0.731	365.5
CGE ($\beta = 0.6$)	0.733	366.7



(a) MoM

(b) CWTM

Fig. 7: *Distributed learning of* LeNet *for MNIST dataset using the D-SGD method with batch-size $k = 64$ and different gradient-filters in the presence of $f = 10$ faulty agents exhibiting* norm-confusing *type of faults.* Exponential averaging *of stochastic gradients applied with different values of β (represented using different colors). Filters used here are* (a) *median-of-means* (MoM)*, and* (b) *coordinate-wise trimmed mean* (CWTM)*, the two that has relatively poor performance without exponential averaging.*

Norm-confusing fault: A faulty agent sends to the server a vector directed opposite to its correct stochastic gradient. However, different form *gradient-reverse* fault, the norm of the vector is scaled to the $(f+1)$-th largest norm amongst the stochastic gradients of all the $n - f$ non-faulty agents.

Experiments are conducted on the distributed learning of LeNet for MNIST with $k = 64$ and different values of f. The outcomes are shown in Fig. 5. Smaller training loss can be achieved with exponential averaging and larger value of β in the same number of steps. Fig. 6 further shows that even if there are large number f of faulty agents and plain CGE converges slowly, exponential averaging can still significantly improve the performance. As shown in Table II, exponential averaging does not have a significant impact on running time.

Experiments also show that exponential averaging improves the fault-tolerance of other gradient-filters. For example, Fig. 7 shows the outcome of the distributed learning task with $k = 64$ and $f = 10$, using median-of-means and coordinate-wise trimmed mean filters. With exponential averaging and larger value of β, faster convergence can be achieved for both filters.

IV. Summary

In this paper, we have proposed a gradient-filter named comparative gradient elimination (CGE) to confer Byzantine fault-tolerate to distributed learning using the distributed stochastic gradient descent method. We have shown that our algorithm tolerates a bounded fraction of Byzantine faulty agents, under some standard assumptions. We have demonstrated through experiments the applicability of our algorithm to distributed learning of neural networks. We have empirically shown that the fault-tolerance of CGE gradient-filter is comparable to state-of-the-art gradient-filters, namely multi-KRUM, geometric median-of-means, and coordinate-wise trimmed mean gradient-filters. Finally, we have also illustrated the effectiveness of exponential averaging for improving upon the fault-tolerance of any given gradient-filter in the D-SGD method.

References

[1] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 4618–4628, 2018.

[2] Barry C Arnold, Richard A Groeneveld, et al. Bounds on expectations of linear systematic statistics based on dependent samples. *The Annals of Statistics*, 7(1):220–223, 1979.

[3] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. In *International Conference on Learning Representations*, 2018.

[4] Dimitris Bertsimas, Karthik Natarajan, and Chung-Piaw Teo. Tight bounds on expected order statistics. *Probability in the Engineering and Informational Sciences*, 20(4):667, 2006.

[5] 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.

[6] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

[7] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[8] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[10] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60, 2017.

[11] 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):44, 2017.

[12] Lisandro D Dalcin, Rodrigo R Paz, Pablo A Kler, and Alejandro Cosimo. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011.

[13] Deepesh Data and Suhas Diggavi. Byzantine-resilient sgd in high dimensions on heterogeneous data. *arXiv preprint arXiv:2005.07866*, 2020.

[14] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. *arXiv preprint arXiv:1803.02815*, 2018.

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

[16] Avishek Ghosh, Raj Kumar Maity, Swanand Kadhe, Arya Mazumdar, and Kannan Ramchandran. Communication-efficient and byzantine-robust distributed learning. *arXiv preprint arXiv:1911.09721*, 2019.

[17] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530, 2018.

[18] Nirupam Gupta and Nitin H Vaidya. Byzantine fault tolerant distributed linear regression. *arXiv preprint arXiv:1903.08752*, 2019.

[19] Nirupam Gupta and Nitin H Vaidya. Fault-tolerance in distributed optimization: The case of redundancy. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 365–374, 2020.

[20] Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via resampling, 2020.

[21] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[22] 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.

[23] Shuo Liu, Nirupam Gupta, and Nitin H Vaidya. Approximate byzantine fault-tolerance in distributed optimization. *arXiv preprint arXiv:2101.09337*, 2021.

[24] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[25] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR, abs/1211.5063*, 2, 2012.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[27] Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

[28] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015.

[29] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[30] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pages 3517–3529, 2017.

[31] Lili Su and Nitin H Vaidya. Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms. In *Proceedings of the 2016 ACM symposium on principles of distributed computing*, pages 425–434. ACM, 2016.

[32] Shreyas Sundaram and Bahman Gharesifard. Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, 2018.

[33] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[34] Zhixiong Yang and Waheed U Bajwa. Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning. *IEEE Transactions on Signal and Information Processing over Networks*, 5(4):611–627, 2019.

[35] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5636–5645, 2018.

V. Appendix: Proof of Theorem 1

Before we present our proof of the theorem we state below some basic results that are used later in the proof.

A. Useful Observations and Lemmas

Recall from (7) in Section II-A that for each non-faulty agent i, and a deterministic real-valued function Ψ,

$$\mathbb{E}_{\zeta^t}\, \Psi\left(g_i^t\right) = \mathbb{E}_{\zeta_1^t,\ldots,\zeta_n^t}\, \Psi\left(g_i^t\right). \tag{16}$$

Also, from (6), recall that for each non-faulty agent i,

$$\zeta_i^t = \mathbf{z}_i^t. \tag{17}$$

For a given the current estimate w^t, the stochastic gradient g_i^t is a function of data points $\mathbf{z}_i^t$ sampled by the agent i. As the non-faulty agents choose their data points independently and identically from distribution $\mathcal{D}$ in each iteration, (17) implies that for each non-faulty agent i,

$$\mathbb{E}_{\zeta_1^t, \ldots, \zeta_n^t} \Psi\left(g_i^t\right) = \mathbb{E}_{\mathbf{z}_i^t} \Psi\left(g_i^t\right). \tag{18}$$

Upon substituting from (18) in (16) we obtain that for each non-faulty agent i,

$$\mathbb{E}_{\zeta^t} \Psi\left(g_i^t\right) = \mathbb{E}_{\mathbf{z}_i^t} \Psi\left(g_i^t\right). \tag{19}$$

For an arbitrary non-faulty agent i and t, $\mathbb{E}_{\zeta_i^t}\left(g_i^t\right) = \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)$. Upon substituting g_i^t from (4) we obtain that

$$\mathbb{E}_{\zeta_i^t}\left(g_i^t\right) = \left(\frac{1}{k}\right) \mathbb{E}_{\mathbf{z}^t} \sum_{j=1}^{k} \left(\nabla \ell(w^t, \, z_{i_j}^t)\right) \tag{20}$$

where the gradient of loss function $\ell(\cdot, \, \cdot)$ is with respect to its first argument w. From (5), recall that $\mathbf{z}^t$ constitutes k data points that are i.i.d. as per the probability distribution $\mathcal{D}$. Upon using this fact in (20) we obtain that

$$\mathbb{E}_{\zeta_i^t}\left(g_i^t\right) = \frac{1}{k} \sum_{j=1}^{k} \mathbb{E}_{z_{i_j}^t \sim \mathcal{D}} \left(\nabla \ell(w^t, \, z_{i_j}^t)\right), \quad \forall t. \tag{21}$$

Note that

$$\nabla Q(w) = \mathbb{E}_{z \sim \mathcal{D}} \left(\nabla \ell(w, \, z)\right), \quad \forall w \in \mathbb{R}^d. \tag{22}$$

Substituting from (22) in (21) we obtain that an arbitrary non-faulty agent i,

$$\mathbb{E}_{\zeta_i^t}\left(g_i^t\right) = \frac{1}{k} \sum_{j=1}^{k} \left(\nabla Q(w^t)\right) = \nabla Q(w^t). \tag{23}$$

Recall, from Assumption 1, that the variance of each non-faulty agent's stochastic gradient is bounded by σ^2.

Lemma 1. *For an arbitrary iteration t, if Assumption 1 holds true then for each non-faulty agent i,*

$$\mathbb{E}_{\zeta^t} \left\|g_i^t\right\|^2 \leq \sigma^2 + \left\|\nabla Q(w^t)\right\|^2.$$

Proof. Let i be an arbitrary non-faulty agent. Using the definition of Euclidean norm, note that for each iteration t,

$$\left\|g_i^t - \mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2 = \left\|g_i^t\right\|^2 - 2\left\langle g_i^t, \, \mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\rangle + \left\|\mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2. \tag{24}$$

As the expected value of a constant is the constant itself, upon taking expectations on both sides in (24) we obtain that

$$\mathbb{E}_{\zeta^t} \left\|g_i^t - \mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2 = \mathbb{E}_{\zeta^t} \left\|g_i^t\right\|^2 - \left\|\mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2. \tag{25}$$

Note, from (19), that $\mathbb{E}_{\zeta^t}\left(g_i^t\right) = \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)$, and $\mathbb{E}_{\zeta^t} \left\|g_i^t - \mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2 = \mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t - \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|^2$. Substituting these in (25) we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t - \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|^2 = \mathbb{E}_{\zeta^t} \left\|g_i^t\right\|^2 - \left\|\mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|^2. \tag{26}$$

Recall from (23) that $\mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right) = \nabla Q(w^t)$. Substituting this above we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t - \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|^2 = \mathbb{E}_{\zeta^t} \left\|g_i^t\right\|^2 - \left\|\nabla Q(w^t)\right\|^2. \tag{27}$$

As Assumption 1 holds true, $\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t - \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|^2 \leq \sigma^2$. Substituting this in (27) proves the lemma. □

Let $|\cdot|$ denote the set cardinality. Recall that there are at least $n - f$ non-faulty agents. We define a set $\mathcal{H}$ constituting of $n - f$ non-faulty agents, i.e., if $i \in \mathcal{H}$ then agent i is non-faulty and $|\mathcal{H}| = n - f$. Let $\mathcal{B} = \{1, \ldots, n\} \setminus \mathcal{H}$ denote the remaining f agents, some of which may be non-faulty.

Lemma 2. *For each iteration t, let ν_t denote the non-faulty agent in $\mathcal{H}$ with stochastic gradient $g_{\nu_t}^t$ of largest Euclidean norm, that is,*

$$\left\|g_{\nu_t}^t\right\| \geq \left\|g_i^t\right\|, \quad \forall i \in \mathcal{H}.$$

For an arbitrary iteration t, if Assumption 1 holds true then

$$\mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\| \leq \sigma\left(1 + \sqrt{n - f - 1}\right) + \left\|\nabla Q(w^t)\right\|. \tag{28}$$

Proof. We begin our proof by reviewing a generic result on the upper bounds on the expectation of highest order statistic [2], [4]. For a positive finite integer p, let $R_1, \ldots, R_p$ be p independent real-valued random variables. Consider a random variable $R_\nu = \max\{R_1, \ldots, R_p\}$. Let $\mathbb{E}(\cdot)$ denote the mean value of a random variable. If the mean and the variance of the random variables $R_1, \ldots, R_p$ are identically equal to $\mathbb{E}(R)$ and $\mathsf{Var}(R)$, respectively, then (see [2])

$$\mathbb{E}(R_\nu) \leq \mathbb{E}(R) + \sqrt{\mathsf{Var}(R)\,(p-1)}. \tag{29}$$

Consider an arbitrary iteration t. Recall that $\mathcal{H}$ comprises of only non-faulty agents, specifically $n - f$ non-faulty agents. Thus, from (19) we obtain that, for all $i \in \mathcal{H}$,

$$\mathbb{E}_{\zeta^t} \left\|g_i^t\right\| = \mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\|, \text{and} \tag{30}$$

$$\mathbb{E}_{\zeta^t} \left(\left\|g_i^t\right\| - \mathbb{E}_{\zeta^t} \left\|g_i^t\right\|\right)^2 = \mathbb{E}_{\mathbf{z}_i^t} \left(\left\|g_i^t\right\| - \mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\|\right)^2. \tag{31}$$

Now, recall from the definition of ν_t and $g_{\nu_t}^t$ that $\left\|g_{\nu_t}^t\right\|$ is a real-valued random variable such that

$$\left\|g_{\nu_t}^t\right\| = \max\left\{\left\|g_i^t\right\|, \ i \in \mathcal{H}\right\}.$$

Therefore, substituting from (30) and (31) in (29), we obtain that for an arbitrary agent $i \in \mathcal{H}$,

$$\mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\| \leq \mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\| + \sqrt{\mathbb{E}_{\mathbf{z}_i^t} \left(\left\|g_i^t\right\| - \mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\|\right)^2 (n - f - 1)}. \tag{32}$$

Owing to Jensen's inequality [9], for all $i \in \mathcal{H}$,

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\| = \mathbb{E}_{\mathbf{z}_i^t} \sqrt{\left\|g_i^t\right\|^2} \leq \sqrt{\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\|^2}. \tag{33}$$

As Assumption 1 holds true, from Lemma 1 we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\|^2 \leq \sigma^2 + \left\|\nabla Q(w^t)\right\|^2, \quad \forall i \in \mathcal{H}. \tag{34}$$

Substituting from (34) in (33) we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\| \leq \sqrt{\sigma^2 + \left\|\nabla Q(w^t)\right\|^2}\,, \quad \forall i \in \mathcal{H}. \tag{35}$$

From triangle inequality, $\sqrt{\sigma^2 + \left\|\nabla Q(w^t)\right\|^2} \leq \sigma + \left\|\nabla Q(w^t)\right\|$. Substituting this in (35) proves that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\| \leq \sigma + \left\|\nabla Q(w^t)\right\| \;, \quad \forall i \in \mathcal{H}. \tag{36}$$

Now, note that for all i,

$$\mathbb{E}_{\mathbf{z}_i^t} \left(\left\|g_i^t\right\| - \mathbb{E}_{\mathbf{z}_i^t}\left\|g_i^t\right\|\right)^2 = \mathbb{E}_{\mathbf{z}_i^t}\left\|g_i^t\right\|^2 - \left(\mathbb{E}_{\mathbf{z}_i^t}\left\|g_i^t\right\|\right)^2. \tag{37}$$

As the Euclidean norm $\|\cdot\|$ is a convex function [9], Jensen's inequality implies that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\| \geq \left\|\mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|, \quad \forall i \in \mathcal{H}. \tag{38}$$

Recall from (23) that $\mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right) = \nabla Q(w^t)$ for all $i \in \mathcal{H}$. Upon substituting this in (38) we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t\right\| \geq \left\|\nabla Q(w^t)\right\|, \quad \forall i \in \mathcal{H}. \tag{39}$$

Substituting from (39) in (37) we obtain that for all $i \in \mathcal{H}$,

$$\mathbb{E}_{\mathbf{z}_i^t} \left(\left\|g_i^t\right\| - \mathbb{E}_{\mathbf{z}_i^t}\left\|g_i^t\right\|\right)^2 \leq \mathbb{E}_{\mathbf{z}_i^t}\left\|g_i^t\right\|^2 - \left\|\nabla Q(w^t)\right\|^2. \tag{40}$$

Substituting from (34) in (40) above we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left(\left\|g_i^t\right\| - \mathbb{E}_{\mathbf{z}_i^t}\left\|g_i^t\right\|\right)^2 \leq \sigma^2 \;, \quad \forall i \in \mathcal{H}. \tag{41}$$

Substituting from (36) and (41) in (32) proves the lemma. □

B. *Proof of* (11) *in Theorem 1*

In this subsection, we prove (11), i.e.,

$$\rho = 1 - \left(n^2 + (n-f)^2\mu^2\right)\,\eta\left(\overline{\eta} - \eta\right) \in (0,\, 1)$$

where $0 < \eta < \overline{\eta}$. As $\eta\left(\overline{\eta} - \eta\right) > 0$, obviously, $\rho < 1$. As $\eta \in (0,\, \overline{\eta})$, upon substituting $\eta = \overline{\eta} - \delta$, where $\delta \in (0, \overline{\eta})$, in (11) we obtain that

$$\begin{aligned}\rho &= 1 - \left(n^2 + (n-f)^2\mu^2\right)\,(\overline{\eta} - \delta)\delta \qquad (42)\\ &= 1 + \left(n^2 + (n-f)^2\mu^2\right)\delta^2 - \overline{\eta}\left(n^2 + (n-f)^2\mu^2\right)\delta = 1\\ &+ \left(n^2 + (n-f)^2\mu^2\right)\left(\delta - \frac{\overline{\eta}}{2}\right)^2 - \frac{\overline{\eta}^2\left(n^2 + (n-f)^2\mu^2\right)}{4}.\end{aligned}$$

As $\left(\delta - \frac{\overline{\eta}}{2}\right)^2 \geq 0$, (42) implies that

$$\rho \geq 1 - \frac{\overline{\eta}^2\left(n^2 + (n-f)^2\mu^2\right)}{4}. \tag{43}$$

Substituting $\overline{\eta}$ from (10) in (44) we obtain that

$$\rho \geq 1 - \frac{(2\lambda + \mu)^2 n^2 \alpha^2}{(n^2 + (n-f)^2\mu^2)}. \tag{44}$$

Recall, from (9), that

$$\alpha = \frac{\lambda n - f(2\lambda + \mu)}{n(2\lambda + \mu)}. \tag{45}$$

Substituting from above in (44) we obtain that

$$\rho \geq 1 - \frac{(\lambda n - f(2\lambda + \mu))^2}{(n^2 + (n-f)^2\mu^2)} = 1 - \frac{((n-f)\lambda - f(\lambda + \mu))^2}{(n^2 + (n-f)^2\mu^2)}. \tag{46}$$

As $\alpha > 0$, $(n-f)\lambda - f(\lambda + \mu) > 0$. Thus,

$$((n-f)\lambda - f(\lambda + \mu))^2 \leq (n-f)^2\lambda^2. \tag{47}$$

Substituting from (47) in (46) we obtain that

$$\rho \geq 1 - \frac{(n-f)^2\lambda^2}{(n^2 + (n-f)^2\mu^2)}. \tag{48}$$

Now, consider a minimum point w^* of the expected loss function $Q(w)$, and an arbitrary finite $w \in \mathbb{R}^d$. Note that $\nabla Q(w^*) = 0$. Thus, Assumption 2 implies that

$$\left\|\nabla Q(w)\right\| \leq \mu \left\|w - w^*\right\|. \tag{49}$$

Now, under Assumption 3,

$$\left\langle w - w^*,\, \nabla Q(w)\right\rangle \geq \lambda \left\|w - w^*\right\|^2. \tag{50}$$

Due to Cauchy-Schwartz inequality, $\left\langle w - w^*,\, \nabla Q(w)\right\rangle \leq \left\|w - w^*\right\| \left\|\nabla Q(w)\right\|$. Thus, (50) implies that

$$\left\|Q(w)\right\| \geq \lambda \left\|w - w^*\right\|. \tag{51}$$

From (49) and (51) we obtain that $\lambda \leq \mu$. Upon using this inequality in (48) we obtain that

$$\rho \geq 1 - \frac{(n-f)^2\mu^2}{(n^2 + (n-f)^2\mu^2)}. \tag{52}$$

As $(n-f)^2\mu^2 < n^2 + (n-f)^2\mu^2$, (52) implies that $\rho > 0$.

C. *Proof of* (12) *in Theorem 1*

In this section, we prove (12), i.e, for all $t \geq 0$,

$$\mathbb{E}_t \left\|w^{t+1} - w^*\right\|^2 \leq \rho^{t+1}\left\|w^0 - w^*\right\|^2 + \left(\frac{1 - \rho^{t+1}}{1 - \rho}\right)\mathsf{M}^2.$$

Consider an arbitrary iteration t. Recall from (2) in Algorithm 1 that the stochastic gradient with the j-th smallest norm, $g_{i_j}^t$, is sent by agent i_j where $j \in \{1, \ldots, n\}$. Let

$$\mathfrak{g}^t = \sum_{j \in \{i_1, \ldots, i_{n-f}\}} g_j^t \tag{53}$$

denote the aggregate of the $n-f$ stochastic gradients received by the server with the $n-f$ smallest Euclidean norms. Upon substituting $\eta_t = \eta$, and $\mathfrak{g}^t$ from (53), in (3) we obtain that

$$w^{t+1} = w^t - \eta\,\mathfrak{g}^t, \quad \forall t. \tag{54}$$

Thus, from the definition of Euclidean norm,

$$\left\|w^{t+1} - w^*\right\|^2 = \left\|w^t - w^*\right\|^2 - 2\eta\left\langle w^t - w^*,\, \mathfrak{g}^t\right\rangle + \eta^2\left\|\mathfrak{g}^t\right\|^2. \tag{55}$$

Now, owing to the triangle inequality, we obtain that

$$\left\|\mathfrak{g}^t\right\| \leq \sum_{j \in \{i_1, \ldots, i_{n-f}\}} \left\|g_j^t\right\|. \tag{56}$$

Recall that $\mathcal{H}$ denotes a set comprising $n-f$ non-faulty agents, i.e., $|\mathcal{H}| = n - f$, and $\{i_1, \ldots, i_{n-f}\}$ represents the agents that sent stochastic gradients with smallest $n-f$ norm. Thus,

$$\sum_{j \in \{i_1, \ldots, i_{n-f}\}} \left\|g_j^t\right\| \leq \sum_{j \in \mathcal{H}} \left\|g_j^t\right\|. \tag{57}$$

Substituting from (57) in (56) we obtain that

$$\left\|\mathfrak{g}^t\right\| \leq \sum_{j \in \mathcal{H}} \left\|g_j^t\right\|. \tag{58}$$

Thus, $\left\|\mathfrak{g}^t\right\|^2 \leq |\mathcal{H}| \sum_{j \in \mathcal{H}} \left\|g_j^t\right\|^2 = (n-f) \sum_{j \in \mathcal{H}} \left\|g_j^t\right\|^2$. Substituting this in (55) implies that

$$\begin{aligned}\left\|w^{t+1} - w^*\right\|^2 \leq \left\|w^t - w^*\right\|^2 - 2\eta \left\langle w^t - w^*, \, \mathfrak{g}^t \right\rangle \\ + \eta^2 (n-f) \sum_{j \in \mathcal{H}} \left\|g_j^t\right\|^2.\end{aligned} \tag{59}$$

Let $\mathcal{H}^t = \{i_1, \ldots, i_{n-f}\} \cap \mathcal{H}$ and $\mathcal{B}^t = \{i_1, \ldots, i_{n-f}\} \setminus \mathcal{H}^t$. Note that

$$\left|\mathcal{H}^t\right| \geq |\mathcal{H}| - f = n - 2f, \quad \text{and} \quad \left|\mathcal{B}^t\right| \leq f. \tag{60}$$

Therefore, recalling from (53),

$$\mathfrak{g}^t = \sum_{i \in \mathcal{H}^t} g_i^t + \sum_{j \in \mathcal{B}^t} g_j^t. \tag{61}$$

Thus,

$$\left\langle w^t - w^*, \, \mathfrak{g}^t \right\rangle = \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*, \, g_i^t \right\rangle + \sum_{j \in \mathcal{B}^t} \left\langle w^t - w^*, \, g_j^t \right\rangle. \tag{62}$$

Owing to Cauchy-Schwartz inequality, $\forall j$,

$$\left\langle w^t - w^*, \, g_j^t \right\rangle \geq - \left\|w^t - w^*\right\| \, \left\|g_j^t\right\|. \tag{63}$$

As in Lemma 2, let ν_t denote the non-faulty agent in set $\mathcal{H}$ having stochastic gradient with the largest norm in iteration t. Thus, $\left\|g_{i_{n-f}}^t\right\| \leq \left\|g_{\nu_t}^t\right\|$, and

$$\left\|g_j^t\right\| \leq \left\|g_{\nu_t}^t\right\|, \quad \forall j \in \mathcal{B}^t. \tag{64}$$

Substituting from (64) in (63) we obtain that

$$\left\langle w^t - w^*, \, g_j^t \right\rangle \geq - \left\|w^t - w^*\right\| \, \left\|g_{\nu_t}^t\right\|. \tag{65}$$

Upon substituting from (65) in (62) we obtain that

$$\left\langle w^t - w^*, \, \mathfrak{g}^t \right\rangle \geq \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*, \, g_i^t \right\rangle - \sum_{j \in \mathcal{B}^t} \left\|w^t - w^*\right\| \, \left\|g_{\nu_t}^t\right\|.$$

As $|\mathcal{B}^t| \leq f$ (see (60)), from above we obtain that

$$\left\langle w^t - w^*, \, \mathfrak{g}^t \right\rangle \geq \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*, \, g_i^t \right\rangle - f \left\|w^t - w^*\right\| \, \left\|g_{\nu_t}^t\right\|. \tag{66}$$

We define, for all t,

$$\phi_t = \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*, \, g_i^t \right\rangle - f \left\|w^t - w^*\right\| \, \left\|g_{\nu_t}^t\right\|. \tag{67}$$

Upon substituting from (67) in (66) we get $\left\langle w^t - w^*, \, \mathfrak{g}^t \right\rangle \geq \phi_t$. Substituting this in (59) we obtain that

$$\begin{aligned}\left\|w^{t+1} - w^*\right\|^2 \leq \left\|w^t - w^*\right\|^2 - 2\eta \, \phi_t \\ + \eta^2 (n-f) \sum_{j \in \mathcal{H}} \left\|g_j^t\right\|^2.\end{aligned} \tag{68}$$

Recall the definition of random variable ζ^t from (7). Taking the expectation $\mathbb{E}_{\zeta^t}$ on both sides in (68), and using the fact that $\mathbb{E}_{\zeta^t} \left\|w^t - w^*\right\|^2 = \left\|w^t - w^*\right\|^2$, implies that

$$\begin{aligned}\mathbb{E}_{\zeta^t} \left\|w^{t+1} - w^*\right\|^2 \leq \left\|w^t - w^*\right\|^2 - 2\eta \, \mathbb{E}_{\zeta^t}\left(\phi_t\right) \\ + \eta^2 (n-f) \sum_{j \in \mathcal{H}} \mathbb{E}_{\zeta^t} \left\|g_j^t\right\|^2.\end{aligned} \tag{69}$$

Taking expectation $\mathbb{E}_{\zeta^t}$ on both sides in (67) implies that

$$\begin{aligned}\mathbb{E}_{\zeta^t}\left(\phi_t\right) = \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*, \, \mathbb{E}_{\zeta^t}\left(g_i^t\right) \right\rangle \\ - f \left\|w^t - w^*\right\| \, \mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\|.\end{aligned} \tag{70}$$

As $\mathbb{E}_{\zeta^t}\left(g_i^t\right) = \nabla Q(w^t)$ for all $i \in \mathcal{H}$, (70) implies that

$$\begin{aligned}\mathbb{E}_{\zeta^t}\left(\phi_t\right) = \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*, \, \nabla Q(w^t) \right\rangle \\ - f \left\|w^t - w^*\right\| \, \mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\|.\end{aligned} \tag{71}$$

As w^* is a minimum of $Q(w)$, $\nabla Q(w^*) = 0$. Thus, Assumption 3, i.e., strong convexity of function $Q(w)$, implies that

$$\left\langle w^t - w^*, \, \nabla Q(w^t) \right\rangle \geq \lambda \left\|w^t - w^*\right\|^2. \tag{72}$$

Substituting from (72) in (71) we obtain that

$$\mathbb{E}_{\zeta^t}\left(\phi_t\right) \geq \left|\mathcal{H}^t\right| \, \lambda \left\|w^t - w^*\right\|^2 - f \left\|w^t - w^*\right\| \, \mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\|. \tag{73}$$

From Lemma 2, $\mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\| \leq \sigma\left(1 + \sqrt{n-f-1}\right) + \left\|\nabla Q(w^t)\right\|$. Substituting this in (73), and using the fact that that $|\mathcal{H}^t| \geq n - 2f$ (see (60)), we obtain that

$$\begin{aligned}\mathbb{E}_{\zeta^t}\left(\phi_t\right) \geq (n\lambda - f(2\lambda + \mu)) \left\|w^t - w^*\right\|^2 \\ - f\sigma\left(1 + \sqrt{n-f-1}\right) \left\|w^t - w^*\right\|.\end{aligned} \tag{74}$$

Now, owing to Lemma 1, $\mathbb{E}_{\zeta^t} \left\|g_j^t\right\|^2 \leq \sigma^2 + \left\|\nabla Q(w^t)\right\|^2$ for all $j \in \mathcal{H}$. Recall that $|\mathcal{H}| = n - f$. Thus,

$$\begin{aligned}\sum_{j \in \mathcal{H}} \mathbb{E}_{\zeta^t} \left\|g_j^t\right\|^2 &\leq |\mathcal{H}| \left(\sigma^2 + \left\|\nabla Q(w^t)\right\|^2\right) \\ &= (n-f) \left(\sigma^2 + \left\|\nabla Q(w^t)\right\|^2\right).\end{aligned} \tag{75}$$

As $\nabla Q(w^*) = 0$, Assumption 2 (i.e., Lipschitzness) implies that $\left\|\nabla Q(w^t)\right\| \leq \mu \left\|w^t - w^*\right\|$. Thus, (75) implies that

$$\sum_{j \in \mathcal{H}} \mathbb{E}_{\zeta^t} \left\|g_j^t\right\|^2 \leq (n-f) \left(\sigma^2 + \mu^2 \left\|w^t - w^*\right\|^2\right). \tag{76}$$

Finally, substituting from (74) and (76) in (69) implies that

$$\begin{aligned}&\mathbb{E}_{\zeta^t}\left\|w^{t+1}-w^*\right\|^2 \quad (77)\\ &\leq \left(1-2\eta\left(n\lambda-f(2\lambda+\mu)\right)+\eta^2(n-f)^2\mu^2\right)\left\|w^t-w^*\right\|^2\\ &+2\eta f\sigma\left(1+\sqrt{n-f-1}\right)\left\|w^t-w^*\right\|+\eta^2(n-f)^2\sigma^2.\end{aligned}$$

For real values a and b, $2ab \leq a^2+b^2$. Thus,

$$\begin{aligned}&2\eta f\sigma\left(1+\sqrt{n-f-1}\right)\left\|w^t-w^*\right\| \leq \eta^2 n^2\left\|w^t-w^*\right\|^2\\ &+\left(\frac{f}{n}\right)^2\sigma^2\left(1+\sqrt{n-f-1}\right)^2. \quad (78)\end{aligned}$$

Substituting from (78) in (77) we obtain that

$$\begin{aligned}&\mathbb{E}_{\zeta^t}\left\|w^{t+1}-w^*\right\|^2 \leq \left\{1-2\eta\left(n\lambda-f(2\lambda+\mu)\right)\right\}\left\|w^t-w^*\right\|^2\\ &+\eta^2\left(n^2+(n-f)^2\mu^2\right)\left\|w^t-w^*\right\|^2\\ &+\left(\frac{f^2\left(1+\sqrt{n-f-1}\right)^2}{n^2}+\eta^2(n-f)^2\right)\sigma^2.\end{aligned}$$

Substituting M^2 from (13) above we obtain that

$$\begin{aligned}&\mathbb{E}_{\zeta^t}\left\|w^{t+1}-w^*\right\|^2 \leq \left(1-2\eta\left(n\lambda-f(2\lambda+\mu)\right)\right)\left\|w^t-w^*\right\|^2\\ &+\eta^2\left(n^2+(n-f)^2\mu^2\right)\left\|w^t-w^*\right\|^2+\mathsf{M}^2. \quad (79)\end{aligned}$$

Substituting α from (9), we obtain that

$$n\lambda-f(2\lambda+\mu)=(2\lambda+\mu)n\,\alpha. \quad (80)$$

Therefore,

$$\begin{aligned}&2\eta\left(n\lambda-f(2\lambda+\mu)\right)-\eta^2\left(n^2+(n-f)^2\mu^2\right)\\ &=2\eta(2\lambda+\mu)n\alpha-\eta^2\left(n^2+(n-f)^2\mu^2\right)\\ &=\left(n^2+(n-f)^2\mu^2\right)\eta\left(\left(\frac{2(2\lambda+\mu)n}{n^2+(n-f)^2\mu^2}\right)\alpha-\eta\right).\end{aligned}$$

Substituting $\overline{\eta}$ from (10) above we obtain that

$$\begin{aligned}&2\eta\left(n\lambda-f(2\lambda+\mu)\right)-\eta^2\left(n^2+(n-f)^2\mu^2\right)\\ &=\left(n^2+(n-f)^2\mu^2\right)\eta\left(\overline{\eta}-\eta\right). \quad (81)\end{aligned}$$

Substituting ρ from (11) in (81) we obtain that

$$2\eta\left(n\lambda-f(2\lambda+\mu)\right)-\eta^2\left(n^2+(n-f)^2\mu^2\right)=1-\rho. \quad (82)$$

Substituting from (82) in (79) we obtain that

$$\mathbb{E}_{\zeta^t}\left\|w^{t+1}-w^*\right\|^2 \leq \rho\left\|w^t-w^*\right\|^2+\mathsf{M}^2. \quad (83)$$

Recall from (8) that $\mathbb{E}_0=\mathbb{E}_{\zeta^0}$. Thus, the above proves the theorem for $t=0$, i.e.,

$$\mathbb{E}_0\left\|w^1-w^*\right\|^2 \leq \rho\left\|w^0-w^*\right\|^2+\mathsf{M}^2. \quad (84)$$

Next, we consider the case when $t>0$ in (83).

From Section II-A, recall that the w^t is a function of random variable $\zeta^{t-1}=\{\zeta_1^{t-1},\ldots,\zeta_n^{t-1}\}$ given w^{t-1}. By retracing back to $t=0$ we obtain that w^t is a function of random variables $\zeta^0,\ldots,\zeta^{t-1}$, given the initial estimate w^0. As w^{t+1} is a function of w^t and ζ^t, $\left\|w^{t+1}-w^*\right\|^2$ is a function of random variables $\zeta^0,\ldots,\zeta^{t-1}$, given the initial estimate w^0. Let, for all $t>0$,

$$\mathbb{E}_{\zeta^t|\zeta^0,\ldots,\zeta^{t-1}}\left\|w^{t+1}-w^*\right\|^2$$

denote the conditional expectation of $\left\|w^{t+1}-w^*\right\|^2$ given the random variables $\zeta^0,\ldots,\zeta^{t-1}$ and w^0. Thus, for $t>0$,

$$\mathbb{E}_{\zeta^t}\left\|w^{t+1}-w^*\right\|^2=\mathbb{E}_{\zeta^t|\zeta^0,\ldots,\zeta^{t-1}}\left\|w^{t+1}-w^*\right\|^2. \quad (85)$$

Substituting from (85) in (83) we obtain that, given w^0,

$$\mathbb{E}_{\zeta^t|\zeta^0,\ldots,\zeta^{t-1}}\left\|w^{t+1}-w^*\right\|^2 \leq \rho\left\|w^t-w^*\right\|^2+\mathsf{M}^2. \quad (86)$$

Now, note that due to Baye's rule, for all $t>0$,

$$\begin{aligned}&\mathbb{E}_{\zeta^0,\ldots,\zeta^t}\left\|w^{t+1}-w^*\right\|^2\\ &=\mathbb{E}_{\zeta^0,\ldots,\zeta^{t-1}}\left(\mathbb{E}_{\zeta^t|\zeta^0,\ldots,\zeta^{t-1}}\left\|w^{t+1}-w^*\right\|^2\right).\end{aligned}$$

Substituting from (86) above implies that, given w^0,

$$\begin{aligned}&\mathbb{E}_{\zeta^0,\ldots,\zeta^t}\left\|w^{t+1}-w^*\right\|^2 \leq \mathbb{E}_{\zeta^0,\ldots,\zeta^{t-1}}\left(\rho\left\|w^t-w^*\right\|^2+\mathsf{M}^2\right)\\ &=\rho\,\mathbb{E}_{\zeta^0,\ldots,\zeta^{t-1}}\left\|w^t-w^*\right\|^2+\mathsf{M}^2, \quad \forall t>0.\end{aligned}$$

Recall from (8) that notation $\mathbb{E}_t$ represents the joint expectation $\mathbb{E}_{\zeta^0,\ldots,\zeta^t}$ given w^0 for all t. Upon substituting this notation above we obtain that, for all $t>0$,

$$\mathbb{E}_t\left\|w^{t+1}-w^*\right\|^2 \leq \rho\,\mathbb{E}_{t-1}\left\|w^t-w^*\right\|^2+\mathsf{M}^2. \quad (87)$$

Finally, we use induction to show (12), i.e., $\forall t \geq 0$,

$$\mathbb{E}_t\left\|w^{t+1}-w^*\right\|^2 \leq \rho^{t+1}\left\|w^0-w^*\right\|^2+\left(\frac{1-\rho^{t+1}}{1-\rho}\right)\mathsf{M}^2.$$

Recall that (12) trivially holds true for $t=0$ due to (84). Assume that (12) holds true for $t=\tau-1$ where $\tau \geq 2$, i.e.,

$$\mathbb{E}_{\tau-1}\left\|w^\tau-w^*\right\|^2 \leq \rho^\tau\left\|w^0-w^*\right\|^2+\left(\frac{1-\rho^\tau}{1-\rho}\right)\mathsf{M}^2. \quad (88)$$

From (87) we obtain that

$$\mathbb{E}_\tau\left\|w^{\tau+1}-w^*\right\|^2 \leq \rho\,\mathbb{E}_{\tau-1}\left\|w^\tau-w^*\right\|^2+\mathsf{M}^2. \quad (89)$$

Substituting from (88) in (89) above we obtain that

$$\mathbb{E}_\tau\left\|w^{\tau+1}-w^*\right\|^2 \leq \rho^{\tau+1}\left\|w^0-w^*\right\|^2+\left(\frac{1-\rho^{\tau+1}}{1-\rho}\right)\mathsf{M}^2.$$

The above proves (12) for $t=\tau$. Hence, due to reasoning by induction, (12) holds true for all $t \geq 0$.