Byzantine Fault-Tolerant Distributed Machine Learning Using Stochastic Gradient Descent (SGD) and Norm-Based Comparative Gradient Elimination (CGE)

Nirupam Gupta Shuo Liu Nitin H. Vaidya

Department of Computer Science
Georgetown University
Washington DC, USA

Abstract

This report considers the problem of Byzantine fault-tolerance in homogeneous multi-agent distributed learning. In this problem, each agent samples i.i.d. data points, and the goal for the agents is to compute a mathematical model that optimally fits, in expectation, the data points sampled by all the agents. We consider the case when a certain number of agents may be Byzantine faulty. Such faulty agents may not follow a prescribed learning algorithm. Faulty agents may share arbitrary incorrect information regarding their data points to prevent the non-faulty agents from learning a correct model.

We propose a fault-tolerance mechanism for the distributed stochastic gradient descent (D-SGD) method – a standard distributed supervised learning algorithm. Our fault-tolerance mechanism relies on a norm based gradient-filter, named *comparative gradient elimination* (CGE), that aims to mitigate the detrimental impact of malicious incorrect stochastic gradients shared by the faulty agents by limiting their Euclidean norms. We make the following contributions in this report.

- We show that the CGE gradient-filter guarantees fault-tolerance against a bounded number of Byzantine faulty agents if the stochastic gradients computed by the non-faulty agents satisfy the standard assumption of bounded variance.
- We demonstrate the applicability of the CGE gradient-filer for distributed supervised learning of artificial neural networks.
- We show that the fault-tolerance by the CGE gradient-filter is comparable to that by other state-of-the-art gradient-filters, namely the multi-KRUM, geometric median of means, and coordinate-wise trimmed mean.
- Lastly, we propose a *gradient averaging* scheme that aims to reduce the sensitivity of a supervised learning process to individual agents' data batch-sizes. We show that gradient averaging improves the fault-tolerance property of a gradient-filter, including, but not limited to, the CGE gradient-filter.

Contents

1 Introduction

The problem of distributed multi-agent learning or *federated* learning has gained significant attention in recent years [7, 31, 16, 35, 36]. Unlike the case of centralized learning, in distributed learning a central machine, i.e., a *server*, cannot directly access the data points. Instead, there are multiple machines, i.e., *agents*, in the system and each agent samples data points independently. In the fault-free setting, i.e., when all the agents are free from faults, the goal is to design a *distributed algorithm* that allows the agents to collectively compute (or *learn*) a mathematical model that optimally fits all the data points sampled by all the agents. We, however, consider a scenario wherein some of agents in the system are faulty.

Specifically, we consider a system with n agents where up to f (out of n) agents are Byzantine faulty. The server is assumed fault-free, and the identity of the Byzantine agents is a priori unknown. Byzantine faulty agents may share malicious incorrect information with the server [26]. Thus, in the presence of faulty agents, the 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 sampled only by the non-faulty agents. The mathematical formulation of *fault-tolerant distributed learning* is presented below.

1.1 Fault-tolerant distributed learning

Each non-faulty agent i samples data points independently from an unknown probability distribution $\mathcal{D}$ in the m-dimensional real vector space $\mathbb{R}^m$. The server fixes a mathematical model Π, for instance, a neural network [6], which is characterized by a *parameter vector* w belonging to the d-dimensional real vector space $\mathbb{R}^d$. Each data point $z \in \mathbb{R}^m$ has a loss value which is determined by a real-valued *loss function* $\ell : (w,\, z) \mapsto \mathbb{R}$. We define the non-faulty *expected loss function* to be

$$Q(w) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}} \ell(w,\, z). \tag{1}$$

The fault-tolerance objective for the non-faulty agents is to compute an optimal parameter vector w^* that minimizes $Q(w)$. We define a fault-tolerant distributed learning algorithm formally below.

Definition 1. *A distributed learning algorithm is said to be* fault-tolerant *if it enables the non-faulty agents to compute an optimum parameter vector w^* such that*

$$w^* \in \arg\min_{w \in \mathbb{R}^d} Q(w). \tag{2}$$

System architecture: We consider a server-based distributed system architecture where a *fault-free* server collaborates with all the agents to achieve the above fault-tolerant learning objective. For now, the system is assumed to be synchronous. In the fault-free setting, the *distributed stochastic gradient descent* (D-SGD) method described below is a commonly used algorithm for solving the distributed learning problem using the server-based architecture [6]. However, the D-SGD method is rendered ineffective in presence of faulty agents [5]. Our goal is to design a *fault-tolerant mechanism* that imparts resilience to the D-SGD method against a bounded number of Byzantine faulty agents.

1.2 Distributed stochastic gradient descent method

The D-SGD method is an iterative algorithm in which the server maintains an estimate of an optimal learning parameter, such as w^* defined in (2), and updates it iteratively using *stochastic gradients* computed by the individual agents. Specifically, for an iteration $t \in \{0, 1, \ldots\}$, let w^t denote the current estimate of an optimal learning parameter maintained at the server. The server broadcasts w^t to all the agents. Each agent i computes a stochastic gradient g_i^t, which is a noisy estimate of the true gradient $\nabla Q(w^t)$. As elaborated in Section 2, the agents can compute stochastic gradients independently by sampling a finite number of data points from the probability distribution $\mathcal{D}$. Upon receiving the stochastic gradients from the agents, the server updates the current estimate w^t to

$$w^{t+1} = w^t - \eta_t \sum_{i=1}^{n} g_i^t, \tag{3}$$

where η_t is a positive real value commonly referred as the *step-size* for iteration t.

The above D-SGD method, however, is rendered ineffective in presence of Byzantine faulty agents that may send malicious incorrect gradients to the server [5]. In recent years, several *gradient-filters* have been proposed to make the D-SGD method robust against the faulty agents [5, 11, 14, 34, 42, 45]. Section 1.4 below discusses the existing gradient-filters. In particular, the server uses a gradient-filter to pre-process the gradients received from all the agents, and then uses the pre-processed (or *filtered*) gradients to update the estimates.

We study the fault-tolerance properties of a gradient-filter, named *comparative gradient elimination* (CGE), for the above D-SGD method when solving the distributed multi-agent learning problem. In the CGE gradient-filter, to tolerate up to f Byzantine faulty agents out of n total agents, in each iteration the server eliminates f stochastic gradients received with the largest f Euclidean norms. That is, the server uses the aggregate of only $n - f$ stochastic gradients received with $n - f$ smallest norms for the iterative update step (3). The norm-based gradient elimination was originally proposed for conferring fault-tolerance to the distributed *gradient-descent* method when solving the multi-agent *distributed optimization* problem [18]. However, unlike the distributed gradient-descent method, in the D-SGD method the agents only send *stochastic gradients*, instead of the true gradients, of their individual loss functions to the server. A detailed description of the resulting D-SGD algorithm with the CGE gradient filter, and its formal fault-tolerance properties are presented in Section 2. The algorithm schematic is shown in Figure 1.

The computational complexity of the CGE gradient-filter for tolerating up to f faulty agents out of n total agents is $\mathcal{O}(n(\log n + d))$, which is significantly less compared to the complexity of state-of-the-art gradient-filters, namely the multi-KRUM gradient-filter [5], the geometric median of means gradient-filter [11], and the spectral gradient-filters [10, 14, 15, 34]. The computational complexity of both the mult-KRUM and the geometric median of means gradient-filters is $\mathcal{O}(n(n + d))$, i.e., quadratic in n. A spectral gradient-filter relies on singular value decomposition (SVD) of a $(d \times n)$-dimensional matrix obtained by column-wise stacking of the stochastic

gradients computed by the n agents, and therefore, has a computational complexity of $\mathcal{O}(nd \min\{n, d\})$.

Unlike the previously introduced applications of norm-based *gradient elimination*, for solving other unrelated problems, in the gradient-descent method [32, 38], the CGE gradient-filter employs an *adaptive* threshold. Specifically, the norm threshold for eliminating stochastic gradients in our case is not a constant but varies depending upon the norms of the non-faulty agents' stochastic gradients.

1.3 Summary of our contributions

In this report, we propose a gradient-filter, named comparative gradient elimination or CGE, for providing resilience to the D-SGD method against a bounded number of Byzantine faulty agents. We present both theoretical and experimental analyses of our algorithm, each described briefly below. Also, we study the effect of *gradient averaging* on fault-tolerance achieved by any of the competent gradient-filters.

1. **Theory:** We show rigorously, in Section 2.3, that our distributed learning algorithm can tolerate up to a bounded number of Byzantine faulty agents if -

   - the stochastic gradients computed by the non-faulty agents have bounded variance,
   - the gradient of the loss function $\ell(w, z)$ with respect to w is Lipschitz continuous, and
   - the expected loss function $Q(w)$, defined in (1), is strongly convex.

   We note that the above assumptions are fairly standard in pragmatic machine learning settings [6].

2. **Experiments:** As elaborated in Section 3, we demonstrate though experiments the applicability of the CGE gradient-filter for distributed training of artificial neural networks wherein the expected loss function $Q(w)$ need not be convex.

   Besides the CGE gradient-filter, we also simulate other state-of-the-art gradient-filters, namely multi-KRUM [5], geometric median of means [11], and coordinate-wise trimmed mean [45, 39]. For our experiments we consider two openly available benchmark data-sets; MNIST and CIFAR-10.

   To evaluate and compare the performance of different gradient-filters we conduct experiments under different system settings, described below, each of which distinctively affects the fault-tolerance property of a gradient-filter.

   - We consider different data *batch-sizes* used by the agents for computing their individual stochastic gradients.
   - We consider different types of faults for the faulty agents.
   - Lastly, we consider different fractions of faulty agents, i.e., f/n.

   From the above experiments we demonstrate that the fault-tolerance obtained by the CGE gradient-filter is comparable to the state-of-the-art gradient-filters.

3. **Gradient-averaging:** In Section 3.5, we present a technique of *gradient averaging* wherein, in each iteration, the server computes a weighted average of the stochastic gradients sent by the agents in all the previous and the current iterations. The server applies a gradient-filter to the averaged stochastic gradients, and uses the resulting filtered gradient for updating its current estimates. Though experiments we observe that

   - gradient averaging attenuates the sensitivity of gradient-filters to the variance of the stochastic gradients, and
   - *stabilizes* the distributed learning process in presence of faulty agents.

1.4 Related Work

In this subsection, we present comparisons between our contributions and that of the related prior works on fault-tolerance in distributed machine learning.

Subsequent to the initial work on fault-tolerance in distributed *optimization* by Su and Vaidya [40], the problem of fault-tolerance in distributed *learning* has gained significant attention in recent years [1, 3, 5, 9, 11, 13, 14, 37, 42, 43, 44]. The distributed learning problem can be modeled as a special case of the more general distributed optimization problem where the cost function (or the loss function) for each non-faulty agent is equal to the expected loss function defined in (1). However, unlike distributed optimization, in distributed learning the agents may only compute *approximate or noisy* gradients of their individual expected loss functions. Therefore, it is non-trivial to extend the applicability of the algorithms originally proposed for fault-tolerance in distributed optimization, such as the ones in [18, 21, 40, 41], to fault-tolerance in distributed machine learning.

It should be noted that exact fault-tolerance in distributed optimization is achievable in presence of up to f faulty agents if and only if the non-faulty agents' cost functions satisfy the $2f$*-redundancy* property [20, 21]. In the case of homogeneous distributed learning problem, described above in Section 1, as the non-faulty agents' expected loss functions are identical, the $2f$*-redundancy* property is naturally satisfied if $n > 2f$. Therefore, in principle, we can compute an optimal learning parameter defined by (2) despite the presence of some Byzantine faulty agents as long as the faulty agents are in the minority. However, note that in pragmatic distributed learning settings, the agents may only be able to compute and send to the server partial information about their individual expected loss functions. In most cases, the partial information is in the form of *stochastic gradients*. Therefore, even if exact fault-tolerance is feasible in theory, its achievability relies on additional assumptions besides $2f$*-redundancy* as is evident from prior work on this problem [3, 5, 14, 19, 43].

The CGE gradient-filter studied in this report was originally proposed and studied for fault-tolerance in the generic distributed *optimization* setting in our prior work [18, 20]. In the problem of distributed optimization, different agents have different cost functions. Alternately, the setting of distributed optimization considered in [18, 20] is equivalent to the setting of *heterogeneous* distributed learning wherein different agents could sample data points from different probability distributions, unlike

in the setting of homogeneous learning described above in Section 1. However, in our prior work we assume that the agents compute *true* gradients of their individual cost functions to the server. In the learning setting the agents can only compute *stochastic* gradients, i.e., noisy estimates of the true gradients. An extended applicability of the CGE gradient-filter to fault-tolerance in stochastic gradient-descent based distributed *linear regression* was presented in [19]. In the current report, however, we consider a more general learning setting than linear regression wherein the expected loss function $Q(w)$ is an arbitrary strongly convex function with Lipschitz continuous gradients. Moreover, through experiments in the current report we have also demonstrated the applicability of the CGE gradient-filter to fault-tolerance in distributed learning of artificial neural networks wherein the function $Q(w)$ may be non-convex.

In recent years, several other gradient-filters have been proposed and studied for fault-tolerance in the D-SGD method based distributed learning. For all these gradient-filters, including the CGE gradient-filter, in each iteration the server replaces the aggregate of the received stochastic gradients in (3) with a *robust estimate* of the aggregate of the non-faulty stochastic gradients. For instance, in the multi-KRUM gradient-filter the server uses the aggregate of only a few stochastic gradients (out of the n) received depending upon their proximity to each other [5]. In the geometric median of means gradient-filter [11], and the SVD based gradient-filter [14, 15, 34], the server implements standard agnostic mean estimation techniques from the robust statistics literature [10, 24]. The coordinate-wise trimmed mean gradient-filter [44, 45] is an extension of the truncated mean filter which was originally proposed for scalar gradients [40] to the case of higher-dimensional stochastic gradients.

Gradient filtering based on their Euclidean norms (or magnitude), similar to the CGE gradient-filter, has been studied recently for fault-tolerance in the distributed learning problem [17]. In [17], however, each agent samples apriori a finite number of data points from the probability distribution $\mathcal{D}$, and then sends to the server in each iteration the *true* gradient, instead of a *stochastic* gradient, of its individual expected loss function. Ghosh et al. [17] has shown that norm-based gradient filtering achieves order optimal *statistical error rate*, in presence of Byzantine faulty agents, if the probability distribution of data points $\mathcal{D}$ is Gaussian and the true gradients of the agents' individual expected loss functions have a *sub-exponential* probability distribution. In the current report, we consider a more pragmatic setting wherein the non-faulty agents may only send independently computed *stochastic* gradients of the expected loss function $Q(w)$ to the server, and we do not make any assumptions about the probability distribution $\mathcal{D}$ of data points.

Finally, we also note that norm-based gradient elimination similar to the CGE gradient-filter has been utilized in the past for solving unrelated problems, such as mitigating the impact of *gradient explosion* and controlling the privacy-accuracy trade-offs in *differential privacy* protocols, in the D-SGD method [32, 38]. We use norm-based gradient elimination for Byzantine fault-tolerance. Besides this difference in the objectives, unlike these past works, we use a *dynamic* (or adaptive) threshold for eliminating the gradients instead of a *static* threshold. This difference is critical for the fault-tolerance property of the CGE gradient-filter.

2 Algorithm and its Fault-Tolerance Property

In this section, we present the comparative gradient elimination (CGE) gradient-filter for tolerating faulty agents in the distributed stochastic gradient descent (D-SGD) method for solving the distributed learning problem. The description of the algorithm below is followed by its fault-tolerance guarantee in Section 2.3.

Similar to the traditional D-SGD method, the server maintains an estimate of an optimal learning parameter, such as w^* defined in (2), which is updated in each iteration of the algorithm. The initial estimate, named w^0, is chosen arbitrarily by the server from $\mathbb{R}^d$. In each iteration $t \in \{0, 1, \ldots\}$, the server computes estimate w^{t+1} using Steps S1 and S2 presented below. Please refer Figure 1 for an illustration of these steps.

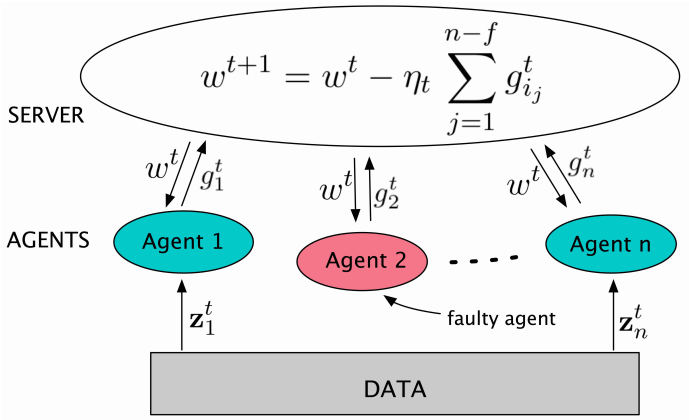

Figure 1: Schematic of our algorithm. Here, the fault agent 2 may concoct arbitrary data points, and may send an arbitrary vector for its stochastic gradient g^t_2.

In Step S1, the server obtains from the agents their locally computed *stochastic gradients* of the average loss function $Q(w)$ at w^t. Now, there are various methods for computing stochastic gradients [6, Section 5], one of which is described below in Section 2.2. Note that a Byzantine faulty agent may send an arbitrary vector for its stochastic gradient. 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 (7) 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.

For two arbitrary real-valued vectors u, v of equal dimensions, let $\langle u, v \rangle$ denote their inner product, i.e.,

$$\langle u, v \rangle = u^T v \tag{4}$$

where $(\cdot)^T$ denotes the transpose. For an arbitrary real-valued vector v, its Euclidean norm, denoted by $\|v\|$, is defined to be

$$\|v\| = \sqrt{\langle v, v \rangle}. \tag{5}$$

2.1 Steps in each iteration

The steps executed by the server and the agents in the t-th iteration are described as follows.

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

Each non-faulty agent i will then send to the server a stochastic gradient of the expected loss function $Q(w)$ defined by (1) at w^t, i.e., a noisy estimator of the gradient $\nabla Q(w^t)$. However, a faulty agent may send an incorrect arbitrary vector for its stochastic gradient.

The gradient received by the server from agent i is denoted as g_i^t. If no gradient is received from some agent i, then agent i must be faulty (because the system is assumed to be synchronous) – in this case, the server assumes a default value of $\mathbf{0}$ vector for the missing gradient g_i^t.

S2: CGE gradient-filter: The server sorts the n received gradients as follows:

$$\left\|g_{i_1}^t\right\| \leq \ldots \leq \left\|g_{i_{n-f}}^t\right\| \leq \left\|g_{i_{n-f+1}}^t\right\| \leq \ldots \leq \left\|g_{i_n}^t\right\|. \tag{6}$$

Stochastic gradient with the j-th smallest norm, $g_{i_j}^t$, is received from agent i_j.

The server updates its current estimate using only $n - f$ stochastic gradients with smallest $n - f$ norms as shown below.

$$w^{t+1} = w^t - \eta_t \sum_{j=1}^{n-f} g_{i_j}^t \tag{7}$$

where η_t, the *step-size* for iteration t, is a positive real value.

The computation of the norm of the n gradients takes $\mathcal{O}(nd)$ time. Sorting of these norms takes additional $\mathcal{O}(n \log n)$ time. Hence, the per iteration computational complexity of the CGE gradient-filter is $\mathcal{O}(n(d + \log n))$.

We present below a standard method for computing stochastic gradients correctly.

2.2 Computing Stochastic Gradients

For computing a stochastic gradient of the expected loss function $Q(w)$, in each iteration t, an agent i chooses k data points $\{z_{i_1}^t, \ldots, z_{i_k}^t\}$. Each data point is sampled independently and identically from the probability distribution $\mathcal{D}$. As elaborated below in Section 2.3, the average of the gradients of the loss functions $\ell(w, z_{i_j}^t),\ j = 1, \ldots, k$, with respect to w is an unbiased noisy estimator of the true

gradient $\nabla Q(w)$, and can be used as a stochastic gradient of $Q(w)$. Thus, for each non-faulty agent i,

$$g_i^t = \frac{1}{k} \sum_{j=1}^{k} \nabla \ell(w^t,\, z_{i_j}^t). \tag{8}$$

The *variance* of g_i^t is inversely proportional to k, i.e., the estimation accuracy of the stochastic gradients is improved if agents sample more data points in each iteration. This obviously has a concomitant computation overhead of computing higher number of gradients. Alternately, the agents can compute stochastic gradients with improved estimation accuracy by using other sophisticated techniques, such as the *dynamic data sampling size method* and *gradient aggregation method*, presented in [6, section 5].

We now present the formal fault-tolerance guarantee of the above algorithm by assuming bounded variance of the stochastic gradients, and *strong convexity* of the expected loss function $Q(w)$.

2.3 Fault-Tolerance Property

In this subsection, we present the formal convergence rate of the above algorithm. For doing so, we assume that the stochastic gradients have bounded variance, the gradient $\nabla Q(w)$ is Lipschitz continuous and the expected loss function $Q(w)$ is strongly convex. The assumptions are stated formally below.

Note, from (7), that for each iteration t the updated estimate w^{t+1} is a function of the current estimate w^t, and the random variables $g_1^t, \ldots, g_n^t$. In case agent i is non-faulty, then recall from Section 2.2 that g_i^t, defined by (8), is a function of w^t and the collection of k data points

$$\mathbf{z}_i^t = \{z_{i_1}^t, \ldots, z_{i_k}^t\} \tag{9}$$

that are independent and identically distributed (i.i.d.) by $\mathcal{D}$. In case agent i is faulty, then g_i^t is an arbitrary d-dimensional random variable. For each $i = 1, \ldots, n$, we define a random variable

$$\zeta_i^t = \begin{cases} \mathbf{z}_i^t & , \quad i \text{ is non-faulty} \\ \\ g_i^t & , \quad i \text{ is faulty} \end{cases} \tag{10}$$

Let,

$$\zeta^t = \left\{\zeta_1^t, \ldots, \zeta_n^t\right\}. \tag{11}$$

For each agent i and each iteration t, let $\mathbb{E}_{\zeta_i^t}(\cdot)$ denote the expected value of a function of the random variable ζ_i^t given the current estimate w^t. Similarly, for each non-faulty agent i, let $\mathbb{E}_{\mathbf{z}_i^t}(\cdot)$ denote the expected value of a function of the random variable $\mathbf{z}_i^t$ given the estimate w^t.

Now, consider an arbitrary non-faulty agent i. From (10) above, note that

$$\mathbb{E}_{\zeta_i^t}\left(g_i^t\right) = \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right), \quad \forall t.$$

Upon substituting g_i^t from (8) on the right hand side above we obtain that

$$\mathbb{E}_{\zeta_i^t}\left(g_i^t\right) = \left(\frac{1}{k}\right) \mathbb{E}_{\mathbf{z}^t} \sum_{j=1}^{k} \left(\nabla \ell(w^t,\, z_{i_j}^t)\right) \tag{12}$$

where the gradient of loss function $\ell(\cdot,\, \cdot)$ is with respect to its first argument w. From (9), 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 (12) we obtain that

$$\mathbb{E}_{\zeta_i^t}\left(g_i^t\right) = \frac{1}{k} \sum_{j=1}^{k} \mathbb{E}_{z_{i_j}^t \sim \mathcal{D}} \left(\nabla \ell(w^t,\, z_{i_j}^t)\right), \quad \forall t. \tag{13}$$

Now, upon applying the gradient operation with respect to w on both sides of (1) we obtain that

$$\nabla Q(w) = \mathbb{E}_{z \sim \mathcal{D}} \left(\nabla \ell(w,\, z)\right), \quad \forall w \in \mathbb{R}^d. \tag{14}$$

Substituting from (14) in (13) we obtain that an arbitrary non-faulty agent i,

$$\mathbb{E}_{\zeta_i^t}\left(g_i^t\right) = \frac{1}{k} \sum_{j=1}^{k} \left(\nabla Q(w^t)\right) = \nabla Q(w^t). \tag{15}$$

Assumption 1 (Bounded variance)**.** *For each non-faulty agent i, assume that the variance of g_i^t is bounded. Specifically, there exists a finite real value σ such that for all non-faulty agent i,*

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

As stated below, we also assume that the gradient of the expected loss function $\nabla Q(w)$ is Lipschitz continuous, and that the function $Q(w)$ is strongly convex [6].

Assumption 2 (Lipschitz continuity of gradients)**.** *Assume that there exists a finite positive real value μ such that for all $w,\, w' \in \mathbb{R}^d$,*

$$\left\| \nabla Q(w) - \nabla Q(w') \right\| \leq \mu \left\| w - w' \right\|.$$

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

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

To be able to state the main convergence result of our algorithm in Theorem 1 we introduce some notation.

- We define a *fault-tolerance margin* α that determines the fraction of faulty agents f/n that can be tolerated by our algorithm for given parameters μ and λ. Specifically,

$$\alpha = \frac{\lambda}{2\lambda + \mu} - \frac{f}{n}. \tag{16}$$

- For each iteration t, let

$$\zeta^t = \{\zeta_i^t,\ i = 1, \ldots, n\}. \tag{17}$$

  Notation $\mathbb{E}_t(\cdot)$ denotes the expectation of a random variable that is a function of the collective random variables $\zeta^0, \ldots, \zeta^t$ given the initial estimate w^0. Specifically,

$$\mathbb{E}_t(\cdot) = \mathbb{E}_{\zeta^0, \ldots, \zeta^t}(\cdot), \quad \forall t \geq 0. \tag{18}$$

- We define a parameter

$$\overline{\eta} = \left(\frac{2(2\lambda + \mu)n}{n^2 + (n-f)^2\mu^2} \right) \alpha \tag{19}$$

  that determines the value for the step-size in (7).

The key result on the convergence of the proposed algorithm is stated below.

Theorem 1. *Consider the iterative algorithm presented above in Section 2. Suppose that the Assumptions 1-3 hold true, the fault-tolerance margin α is positive, and the step-size $\eta_t = \eta > 0$ for all t in (7). Let,*

$$\mathsf{M}^2 = \left(\frac{f^2 \left(1 + \sqrt{n-f-1}\right)^2}{n^2} + \eta^2 (n-f)^2 \right) \sigma^2. \tag{20}$$

If $\eta < \overline{\eta}$ then the following holds true.

1. *The value of*

$$\rho = 1 - \left(n^2 + (n-f)^2\mu^2\right)\, \eta\, (\overline{\eta} - \eta) \tag{21}$$

   is positive and less than 1.

2. *Recall the definition of w^* from (2). Given the initial estimate w^0, that may be chosen arbitrarily from $\mathbb{R}^d$, for all $t \geq 0$,*

$$\mathbb{E}_t \left\| w^{t+1} - w^* \right\|^2 \leq \rho^{t+1} \left\| w^0 - w^* \right\|^2 + \left(\frac{1 - \rho^{t+1}}{1 - \rho} \right) \mathsf{M}^2. \tag{22}$$

The proof of Theorem 1 is deferred to Appendix A.

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

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

then for small enough value of the step-size in (7) the proposed iterative algorithm converges *linearly* in expectation to the neighborhood of an optimal learning parameter defined by (2). Specifically, as $\lim_{t\to\infty} \rho^t = 0$,

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

We now present the fault-tolerance guarantee of our algorithm in probability. Specifically, we have the following corollary of Theorem 1. Notation $Pr(\cdot)$ denotes the probability of an event.

Corollary 1. *Consider the iterative algorithm presented above in Section 2. If the conditions stated in Theorem 1 hold true then for every positive real value ϵ,*

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

The proof for Corollary 1, presented in Appendix C, relies on Markov's inequality. In the subsequent section, we present the formal proof for Theorem 1.

According to Corollary 1, the D-SGD method with the CGE gradient-filer guarantees approximate fault-tolerance. The sequence of estimates converge within ϵ distance to the optimal point with a probability of at least

$$1 - \frac{1}{\epsilon} \left(\frac{\mathsf{M}^2}{1-\rho} \right).$$

The factor M^2, defined in (20), is directly proportional to the variance σ^2 of the stochastic gradients computed by the non-faulty agents. Thus, for a given ϵ, smaller variance of stochastic gradients implies higher probability of convergence within ϵ distance from the solution w^*, and therefore, improved fault-tolerance. Note that the stochastic gradients are unbiased estimators of the true gradient of the global expected loss function, as shown in Section 2.2. Smaller is the variance σ^2, the more accurate are the stochastic gradients (see Assumption 1). Therefore, it is quite intuitive that fault-tolerance of the proposed algorithm should improve when the accuracy of the stochastic gradients computed by the non-faulty agents improves.

In the subsequent section, we demonstrate though experiments the applicability of the CGE gradient-filter for artificial neural networks, in which case the expected loss function $Q(w)$ need not be a convex function. We compare the fault-tolerance achieved by the CGE gradient-filter with that of the other existing gradient-filters, namely multi-KRUM [5], geometric median of means [11], and coordinate-wise trimmed mean [45, 39].

3 Experiments

In this section, we present our empirical studies on fault-tolerance in distributed learning of artificial neural networks using the D-SGD method and different gradient-filters, including our proposed CGE gradient-filter. To evaluate the fault-tolerance of different gradient-filters we conduct experiments with different fractions of faulty agents f/n, different fault types, and different data batch-sizes k used for computing stochastic gradients by non-faulty agents (see Section 2.2).

3.1 Implementation and setup

We simulate the distributed server-based system architecture by spawning multiple threads, one for a server and others for the agents, where inter-thread communication is done through message passing interface (MPI). Our simulator is built in Python using PyTorch [33] and MPI4py [12], and is deployed on a Google Cloud Platform cluster made available by the Massive Data Institute at Georgetown University. The cluster we use has 64 vCPU cores and a memory of 100 GB.

In our experiments, we consider distributed learning of artificial neural networks for supervised classification of two benchmark datasets; the MNIST dataset [27] and the CIFAR-10 dataset [25]. For all our experiments we consider the system to be synchronous and $n = 10$ agents. The server thread initiates the D-SGD method with the initial estimate w^0, a d-dimensional vector where the value of d is equal to the total number of parameters of the artificial neural network being learned, using the initialization technique of Kaiming uniform[22]. For each iteration, in step **S2**, the step-size $\eta_t = 0.1$ for all the gradient-filters described below.

In the subsequent sections, Section 3.2 and Section 3.3, we describe the other gradient-filters and the different types of faults, respectively.

3.2 Other gradient-filters

Besides the CGE gradient-filter, we also evaluate fault-tolerance by four other state-of-the-art gradient-filters: geometric median, geometric median of means [11], coordinate-wise trimmed mean [45], and multi-KRUM [5]. Similar to the CGE gradient-filter, these other gradient-filters are implemented by the server in the second step, i.e., step **S2**, of each iteration of the iterative algorithm described in Section 2.1.

- **Geometric median** (GeoMed): For a set of vectors $\{y_1, \ldots, y_n\}$ in $\mathbb{R}^d$, their geometric median denoted by $\mathsf{med}\{y_1, \ldots, y_n\}$ is defined to be

$$\mathsf{med}\{y_1, \ldots, y_n\} = \arg\min_{y \in \mathbb{R}^d} \sum_{j=1}^{n} \|y_j - y\| \,. \tag{23}$$

  The geometric median is less sensitive to outlying vectors than the arithmetic average [28, 30].

In GeoMed gradient-filter, in step **S2** of each iteration t the server computes the geometric median of the received stochastic gradients $\mathsf{med}\{g_1^t, \ldots, g_n^t\}$, and updates the current estimate w^t to

$$w^{t+1} = w^t - \eta_t \cdot \mathsf{med}\{g_1^t, \ldots, g_n^t\}$$

where η_t denotes the step-size for iteration t.

- **Geometric median of means** (MoM): For the MoM gradient-filter the server divides apriori the agents into l groups indexed from 1 to l. For simplicity, we assume that the total number of agents n is a multiple of l. Thus, each group has $b = n/l$ agents. Let agents $\{1, \ldots, b\}$ belong to group 1, agents $\{b+1, \ldots, 2b\}$ belong to group 2, and so on.

  In step **S2** of each iteration t, for each group $j \in \{1, \ldots, l\}$ the server computes the average $\widehat{g_j^t}$ of the stochastic gradients received from the agents in group j. Specifically, for each group $j \in \{1, \ldots, l\}$,

$$\widehat{g_j^t} = \frac{1}{b} \sum_{j=1}^{b} g_j^t.$$

  Then, the server computes the geometric median of all the groups' averaged stochastic gradients $\mathsf{med}\left\{\widehat{g_1^t}, \ldots, \widehat{g_n^t}\right\}$ as defined in (23). Finally, the server updates the current estimate w^t to

$$w^{t+1} = w^t - \eta_t \cdot \mathsf{med}\left\{\widehat{g_1^t}, \ldots, \widehat{g_n^t}\right\}.$$

- **Coordinate-wise trimmed mean** (CWTM): For a set of real scalar values $\{a_1, ..., a_n\}$, with $a_1 \leq \ldots \leq a_n$, the f-trimmed mean denoted by $\mathsf{tm}_f\{a_1, ..., a_n\}$ is defined to be

$$\mathsf{tm}_f\{a_1, \ldots, a_n\} = \frac{1}{n-2f} \sum_{j=f+1}^{n-f} a_j. \tag{24}$$

  The CWTM gradient-filter is an extension of the scalar trimmed mean filter which was proposed by Su and Vaidya [40]. In CWTM gradient-filter in each iteration the server updates its current estimate using a vector whose elements are f-trimmed means of the corresponding elements of the stochastic gradients received. Specifically, let the l-th element of a vector $y \in \mathbb{R}^d$ be denoted by $y[l]$. In step **S2** of each iteration t, the server computes a vector $\widehat{g^t}$ such that for each $l = 1, ..., d$,

$$\widehat{g^t}[l] = \mathsf{tm}_f\left\{g_1^t[l], \ldots, g_n^t[l]\right\}.$$

  Finally, the server updates the current estimate w^t to

$$w^{t+1} = w^t - \eta_t \cdot \widehat{g^t}.$$

- **Multi-KRUM**: In the multi-KRUM gradient-filter, in each iteration t the server computes the *KRUM score* for each stochastic gradient received from the agents [5, Section 2]. The server chooses m stochastic gradients with m smallest KRUM scores and computes their average g_{km}^t. The parameter m is chosen a priori such that $1 \leq m \leq n-f$. The server updates the current estimate w^t to

$$w^{t+1} = w^t - \eta_t \cdot g_{km}^t.$$

The value of m chosen for our experiments is specified later.

3.3 Fault types

Recall that Byzantine faulty agents can exhibit arbitrary faults. We conduct experiments for 5 different types of faults, described below. In the first three fault types, the faulty agents are assumed Byzantine and omniscient. In the last two fault types, the faulty agents are assumed inadvertent.

- **Gradient-reverse faults**: In this particular fault type, each faulty agent reverses its correct stochastic gradient and scales the reversed gradients by a factor of 100. Specifically, for each iteration t and each agent j, let s_j^t denote agent j's correct stochastic gradient for iteration t. If agent j is faulty then it sends to the server an incorrect gradient

$$g_j^t = -100 \cdot s_j^t.$$

- **Coordinate-wise faults**: In this particular fault type, each fault agent sends a vector whose elements are equal to the f-th smallest respective elements of the stochastic gradients computed by the non-faulty agents. Specifically, if $\mathcal{H}$ denotes the set of all non-faulty (or honest) agents then for each faulty j and $l \in \{1, \ldots, d\}$,

$$g_j^t[l] = \text{the } f\text{-th smallest amongst } \left\{g_i^t[l],\ i \in \mathcal{H}\right\}, \quad \forall t.$$

Recall that for a vector $y \in \mathbb{R}^d$, $y[l]$ denotes its l-th element.

- **Norm-confusing faults**: In this particular fault type, each faulty agent reverses its correct stochastic gradient, and re-scales it so that its norm is equal to the $(f+1)$-th largest norm amongst the stochastic gradients computed by all the non-faulty agents. Specifically, for each iteration t, let s_i^t denote a correct stochastic gradient for each agent i. Let, ν_t denote the non-faulty agent whose stochastic gradient's norm is $(f+1)$-th largest amongst the norms of all the non-faulty agents' stochastic gradients. Then each faulty agent j sends to the server an incorrect stochastic gradient

$$g_j^t = -\left(\frac{\left\|g_{\nu_t}^t\right\|}{\left\|s_j^t\right\|}\right) \cdot s_j^t, \quad \forall t.$$

We also conduct experiments for a few non-Byzantine inadvertent fault types described below. These faults are known to occur in practice due to system hardware failures [42, 23].

- **Label-flipping faults**: In this particular fault type, a faulty agent sends incorrect stochastic gradients due to the output classification labels of their sampled data points being erroneous. To simulate this fault in the case of MNIST or CIFAR-10 data-sets, where there are 10 differnet labels in each of them, for each data point sampled by a faulty agent, we change the original label of the data point y to $\widetilde{y} = 9 - y$.

- **Random faults**: As the name suggests, in this particular fault type, each faulty agent sends a randomly chosen vector from $\mathbb{R}^d$ for its stochastic gradient to the server in each iteration. In our experiments, we consider a scenario where the faulty agents send i.i.d. Gaussian random vectors of mean 0 and an isotropic covariance matrix with standard deviation of 200.

3.4 Results and analysis

As mentioned above, we consider supervised learning of artificial neural networks for classifying images in two different data-sets; MNIST and CIFAR-10. First, we will present our results for the MNIST dataset below.

3.4.1 MNIST Dataset

MNIST is an image-classification dataset of handwritten digits comprising $60,000$ training samples and $10,000$ testing samples [27]. We consider a convolutional neural network called `LeNet` for the classification task that comprises of $431,080$ learning parameters. Thus, the value of dimension $d = 431,080$.

The outcomes for our different experiments are presented below.

Different fault types: To compare the performance of different gradient-filters under the different types of faults described above in Section 3.3,

- we fix the data batch-size used for computing stochastic gradients by all the non-faulty agents to $k = 16$ (see Section 2.2), and

- we consider just 1 faulty agent in the system, i.e., $f = 1$, out of $n = 10$ total agents. Without loss of generality we let agent 1 to be the faulty agent. Note that the identity of the faulty agent is hidden from other non-faulty agents and the server.

The plots in Figure 2 show the learning curve for the different Byzantine faults, while the plots in Figure 3 show the learning curve for the different inadvertent faults.

As is shown in the figures, since the fraction of faulty agents is small here, the D-SGD method converges for all five gradient-filters. We also observe that, unlike the other gradient-filters, the fault-tolerance by CGE gradient-filter is consistently good for all the different types of faults. However, note that in the case of *coordinate-wise faults* shown in Figure 2(b), the learning curve for all the gradient-filters is quite erratic. There are large 'spikes' in the loss plots, and corresponding 'dives' in the precision plots.

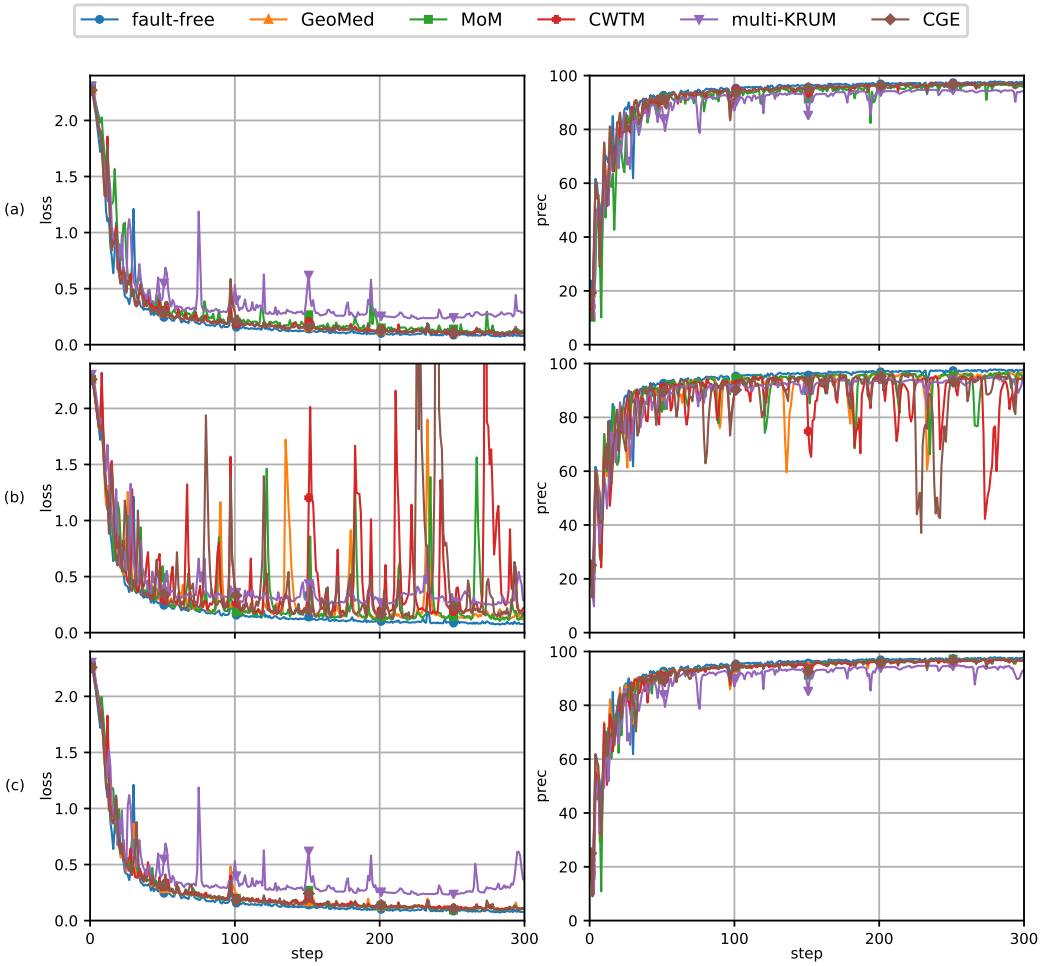

Figure 2: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the different gradient-filters as mentioned in presence of $f = 1$ faulty agent exhibiting the different Byzantine faults. Each row corresponds to a different fault type: (a) *gradient reverse*, (b) *coordinate-wise*, and (c) *norm-confusing*. The first column plots the training loss, whilst the second column plots the testing precision, versus the number of iterations or steps.

Different data batch-sizes: To evaluate the influence of individual agents' data batch-sizes, i.e., the value of k in Section 2.2, on fault-tolerance by a gradient-filter, we simulate the multi-KRUM and CGE gradient-filters for two different data batch-sizes: $k = 16$ and $k = 64$. The learning curves for the two filters for two different Byzantine fault types, namely the *gradient reverse faults* and the *norm-confusing faults*, are shown in Figures 4, 5, 6 and 7. In all these figures we plot the loss and precision upon the completion of the 100-th step or iteration.

Upon comparing the plots in the aforementioned figures we observe that the fault-tolerance of both multi-KRUM and CGE gradient-filter improves significantly with increase in the data batch-size. This is indeed coherent with the theoretical analysis of CGE gradient-filter presented in Section 2.3. Intuitively, a higher batch-size

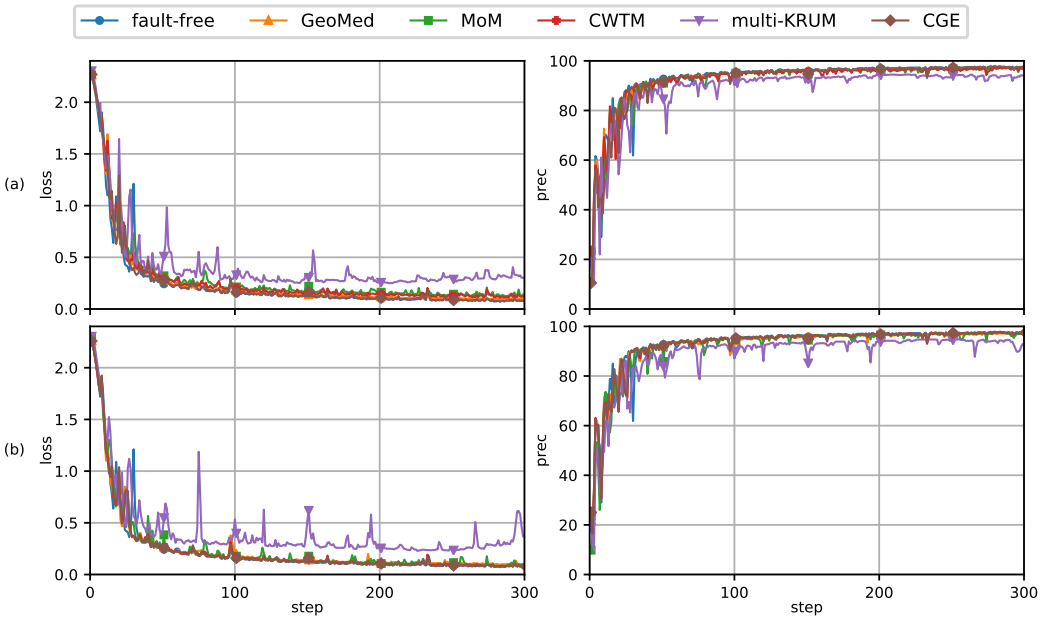

Figure 3: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the different gradient-filters as mentioned in presence of $f = 1$ faulty agent exhibiting the different inadvertent faults. Each row corresponds to a different type of fault: (a) *label-flipping* and (b) *random*. The two columns plot the training loss and the testing precision, respectively.

results in stochastic gradients with improved accuracy and reduced noise or variance. Thus, it is only reasonable for gradient-filters to achieve better fault-tolerance when non-faulty agents compute stochastic gradients by sampling a larger number of data points in each iteration. However, this improvement is accompanied by an increased cost for computation of stochastic gradients in each iteration.

Different fractions of faulty agents: We evaluate the fault-tolerance of mult-KRUM and CGE gradient-filters in the presence of varied fraction of faulty agents, with different faulty behaviors. The plots are shown in Figures 8 to 11. As expected, the learning process for both the gradient-filters deteriorates with increase in the fraction of faults.

From the plots in the aforementioned figures, we observe that whilst the increase in f/n has adversarial effects on the fault-tolerance, the increase in the batch-size improves the fault-tolerance for any given fraction of faults. At large, the observed effects on the fault-tolerance achieved by the CGE gradient-filter to change in the fraction of faults f/n is coherent with the theoretical analysis presented in Section 2.3.

Next, we present experimental results for the case of CIFAR-10 dataset.

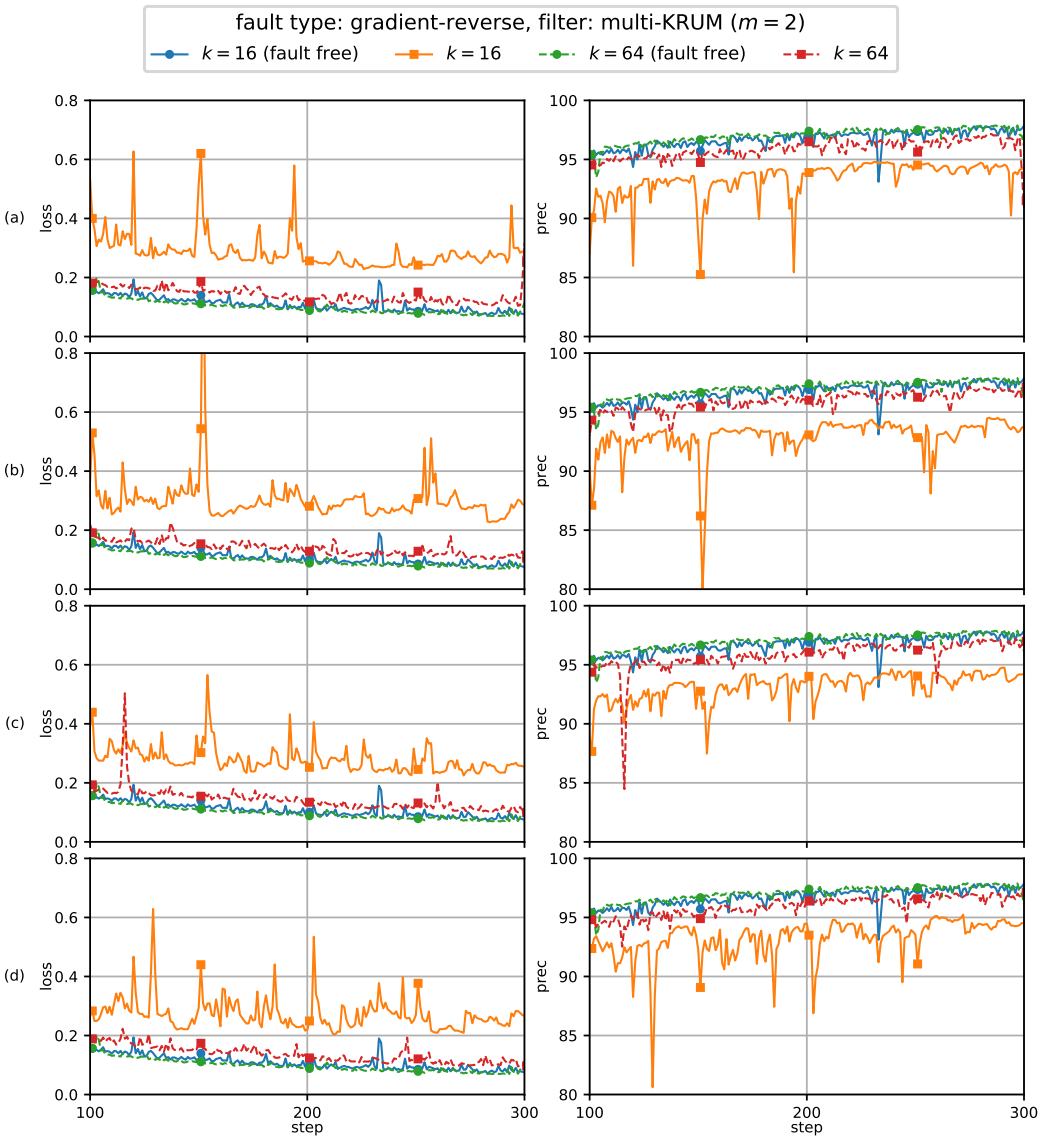

Figure 4: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the multi-KRUM gradient-filter ($m = 2$) with different data batch-sizes. The faulty agents exhibit the *gradient reverse* faults. Different rows present the cases with different number of faults f; (a) $f = 1$, (b) $f = 2$, (c) $f = 3$, and (d) $f = 4$. The two columns plot the training loss and testing precision, respectively.

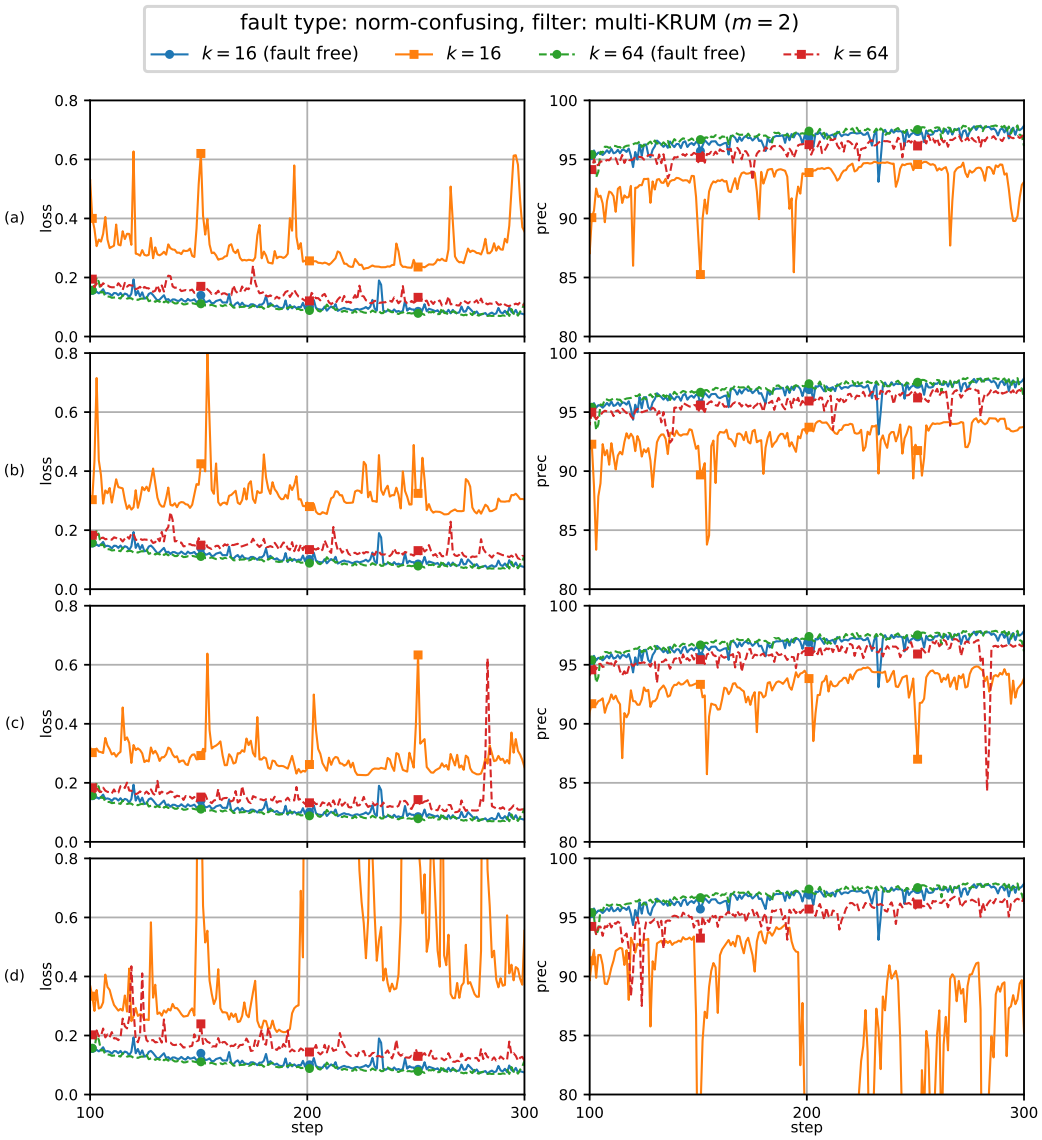

Figure 5: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the multi-KRUM gradient-filter ($m = 2$) with different data batch-sizes. The faulty agents exhibit the *non-confusing* faults. Different rows present the cases with different number of faults f; (a) $f = 1$, (b) $f = 2$, (c) $f = 3$, and (d) $f = 4$. The two columns plot the training loss and testing precision, respectively.

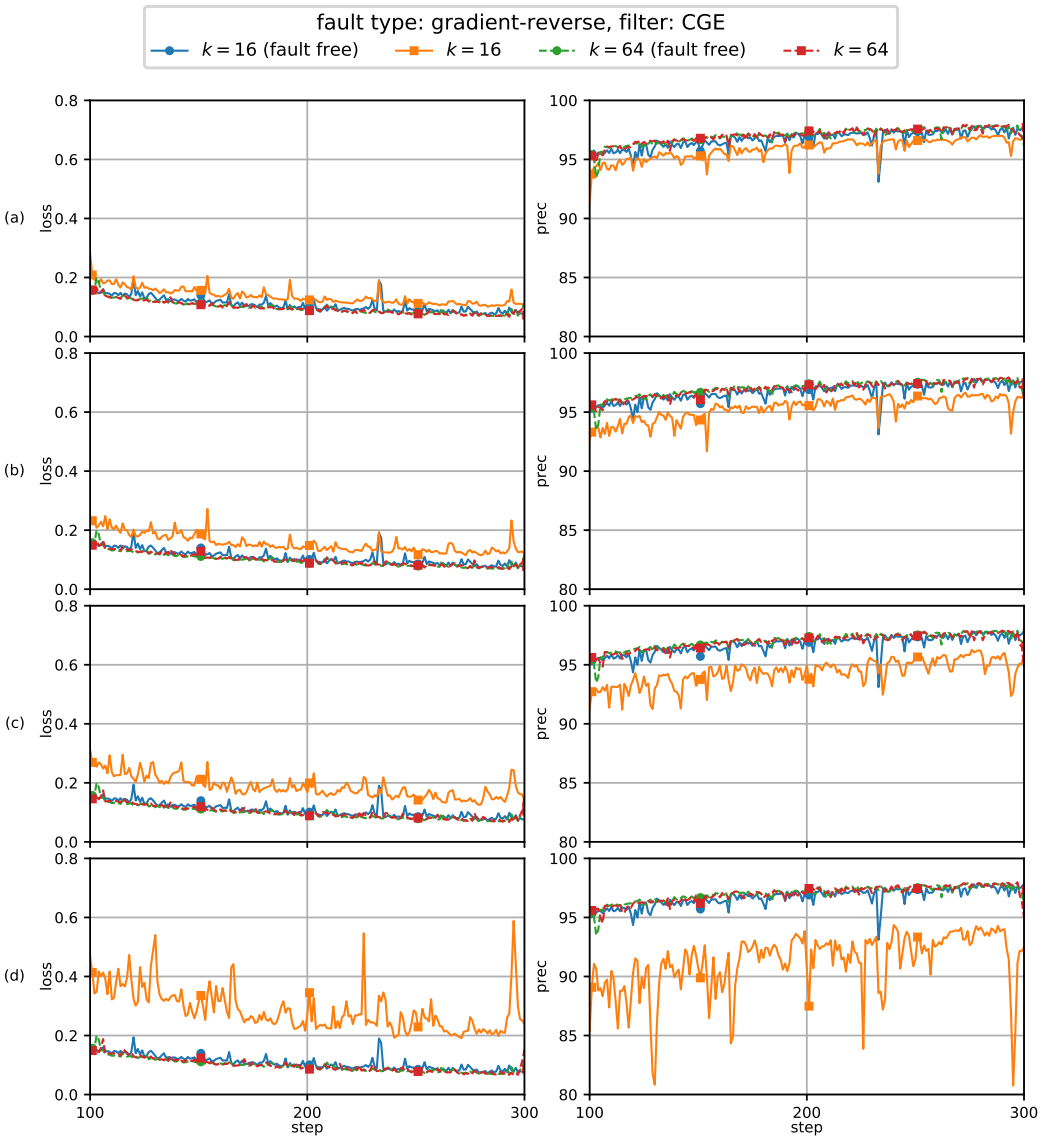

Figure 6: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the CGE gradient-filter with different data batch-sizes. The faulty agents exhibit the *gradient reverse* faults. Different rows present the cases with different number of faults f; (a) $f = 1$, (b) $f = 2$, (c) $f = 3$, and (d) $f = 4$. The two columns plot the training loss and testing precision, respectively.

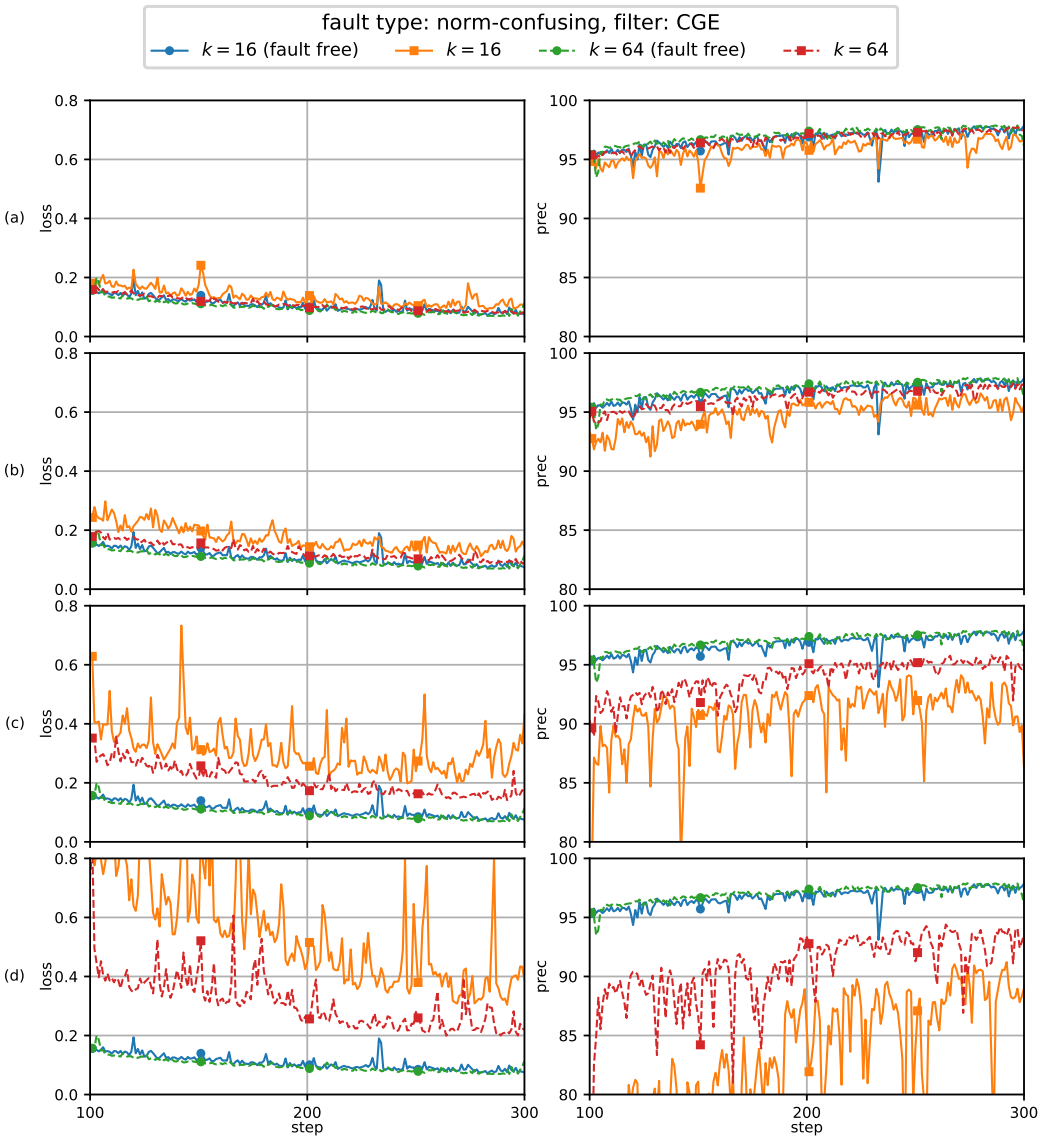

Figure 7: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the CGE gradient-filter with different data batch-sizes. The faulty agents exhibit the *norm-confusing* faults. Different rows present the cases with different number of faults f; (a) $f = 1$, (b) $f = 2$, (c) $f = 3$, and (d) $f = 4$. The two columns plot the training loss and testing precision, respectively.

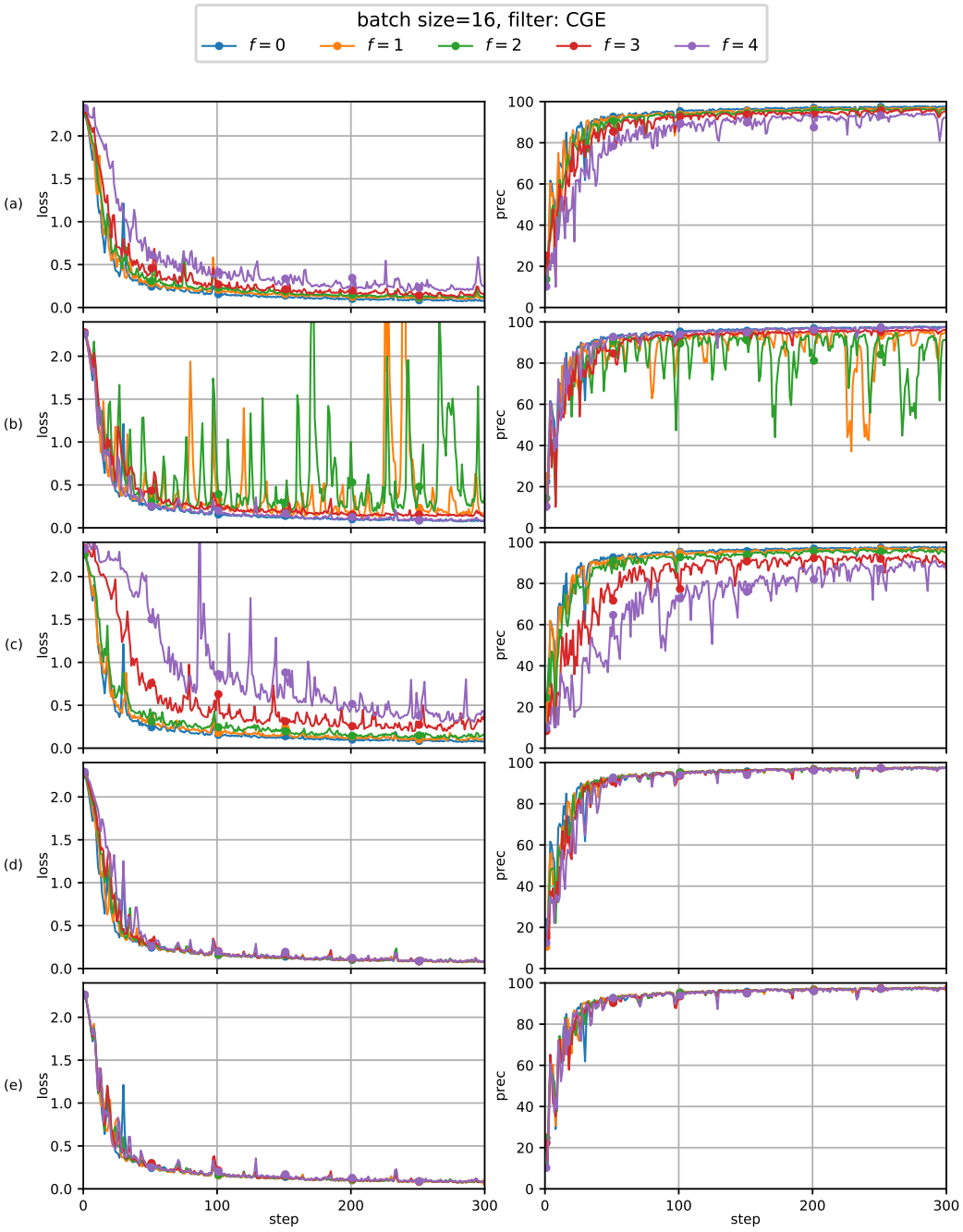

Figure 8: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the CGE gradient-filter under different number of faulty agents f out of $n = 10$. In these experiments the data batch-size $k = 16$. Different rows present different fault types: (a) *gadient reverse*, (b) *coordinate-wise*, (c) *norm-confusing*, (d) *label-flipping*, and (e) *random*. The two columns plot the training loss and the testing precision, respectively.

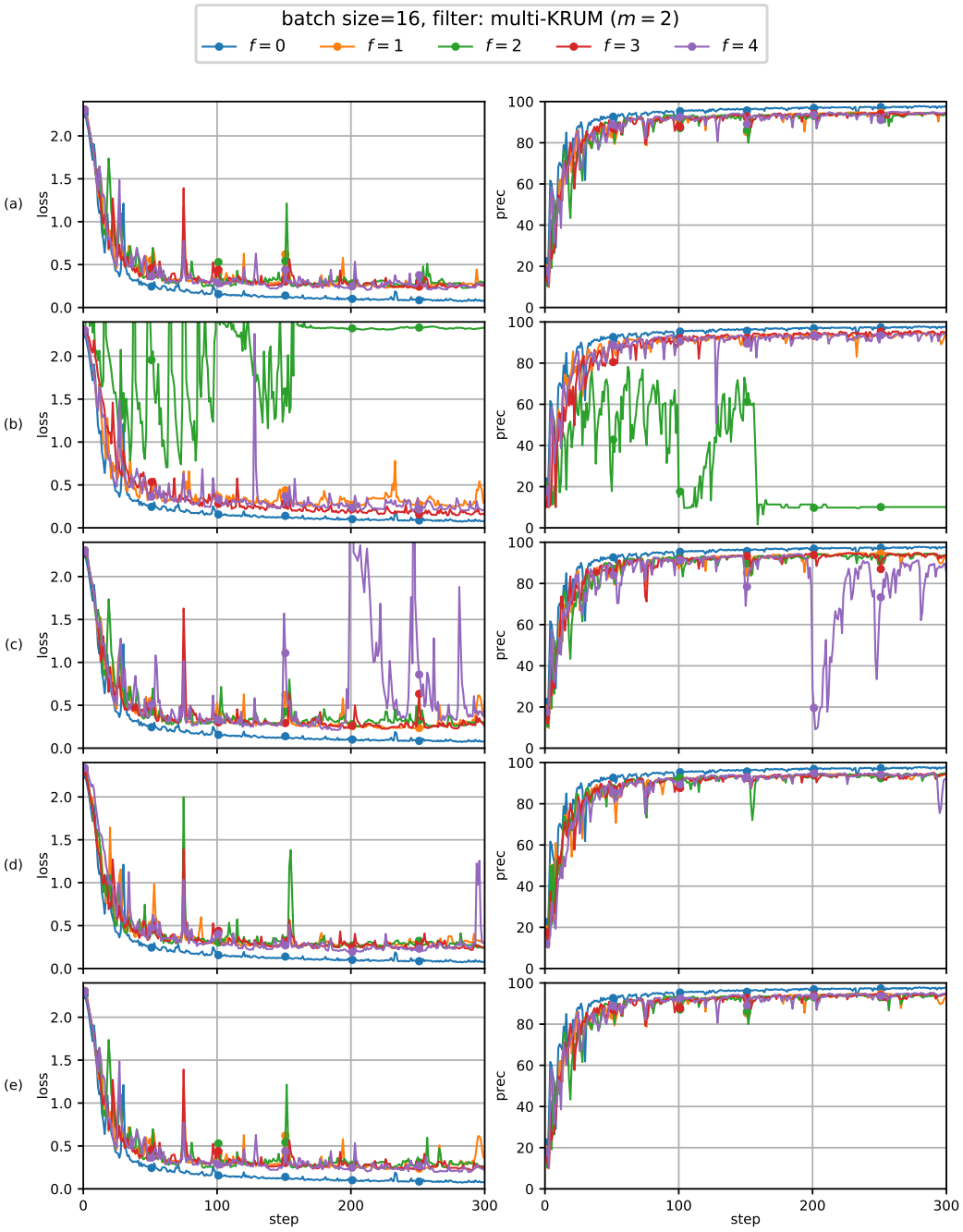

Figure 9: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the multi-KRUM ($m = 2$) gradient-filter under different number of faulty agents f out of $n = 10$. In these experiments the data batch-size $k = 16$. Different rows present different fault types: (a) *gadient reverse*, (b) *coordinate-wise*, (c) *norm-confusing*, (d) *label-flipping*, and (e) *random*. The two columns plot the training loss and the testing precision, respectively.

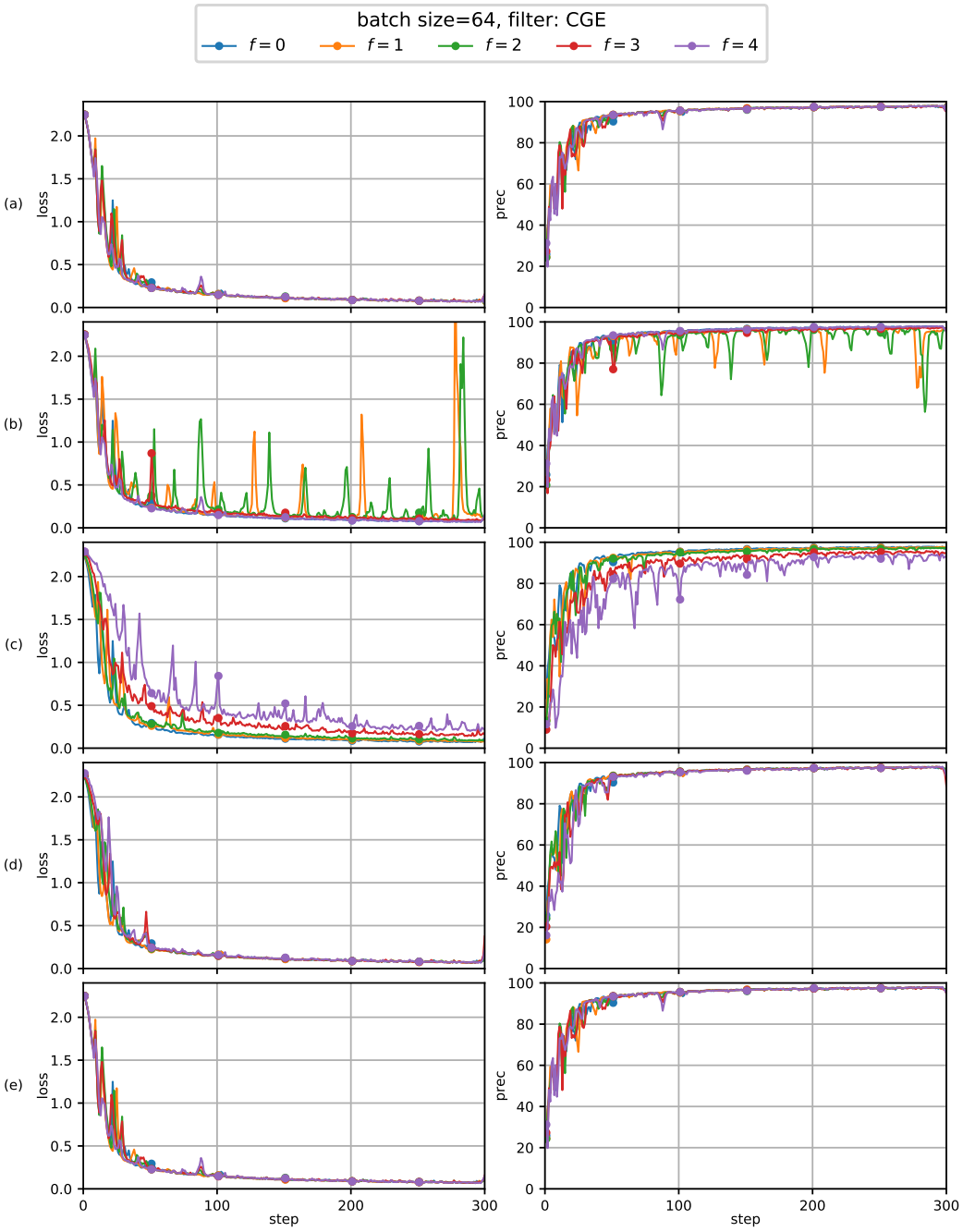

Figure 10: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the CGE gradient-filter under different number of faulty agents f out of $n = 10$. In these experiments the data batch-size $k = 64$. Different rows present different fault types: (a) *gadient reverse*, (b) *coordinate-wise*, (c) *norm-confusing*, (d) *label-flipping*, and (e) *random*. The two columns plot the training loss and the testing precision, respectively.

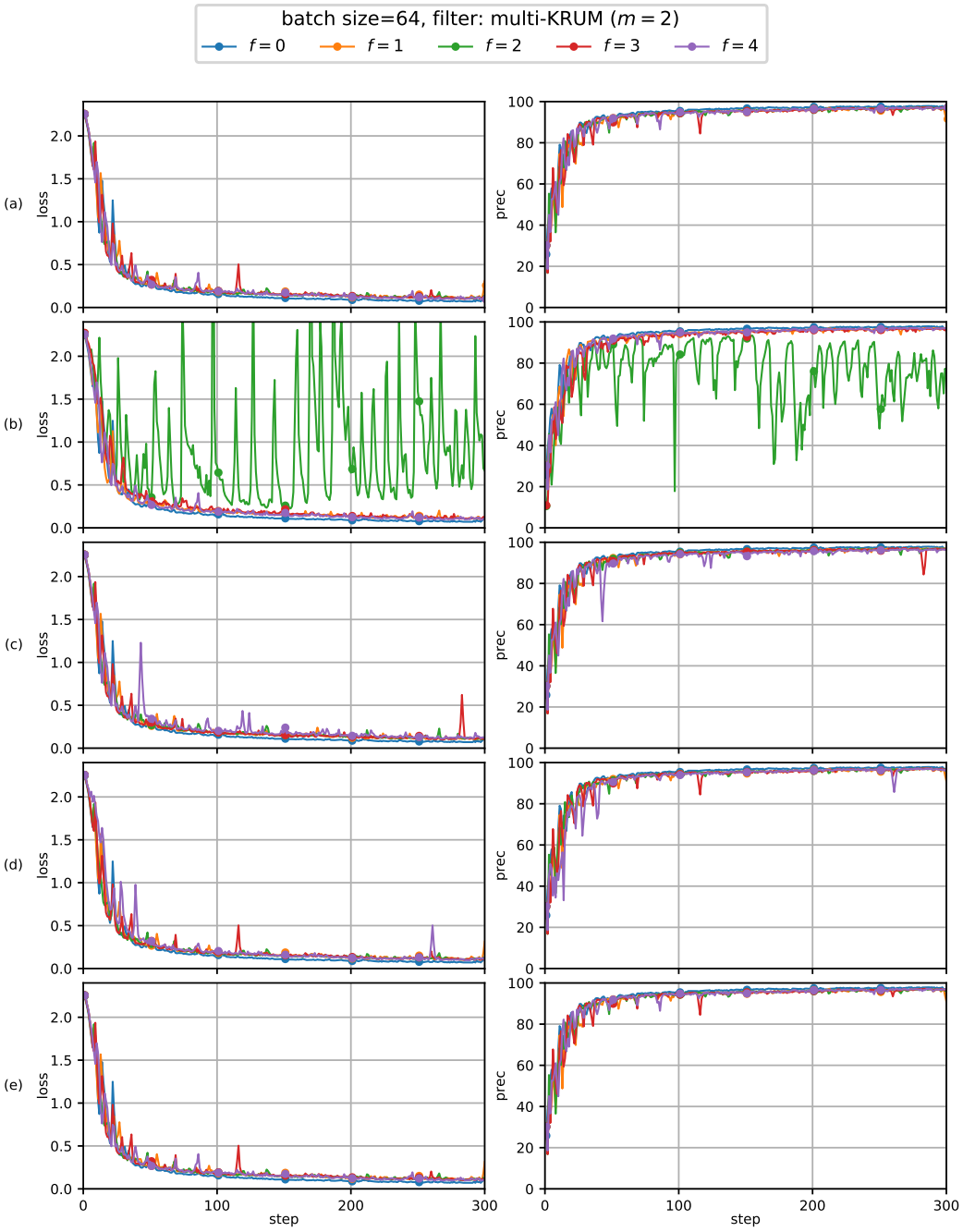

Figure 11: Distributed learning of neural networks for MNIST dataset using the D-SGD method and the multi-KRUM ($m = 2$) gradient-filter under different number of faulty agents f out of $n = 10$. In these experiments the data batch-size $k = 64$. Different rows present different fault types: (a) *gadient reverse*, (b) *coordinate-wise*, (c) *norm-confusing*, (d) *label-flipping*, and (e) *random*. The two columns plot the training loss and the testing precision, respectively.

3.4.2 CIFAR-10 Dataset

Similar to MNIST dataset, CIFAR-10 is also an image-classification dataset comprising $60,000$ small color images in 10 mutually exclusive categories, *e.g.* automobile, bird, dog, etc., with equal number of images in each class [25]. The CIFAR dataset is further divided into $50,000$ training samples and $10,000$ testing samples. Classification in CIFAR-10 is expected to be more difficult than MNIST, since the images in MNIST are monochromatic and has less variety. On the other hand, the images in CIFAR-10 are quite diverse. For instance, an image of a sedan and a SUV both belong to the class of automobiles, and different breeds of dogs can look very differently from each other. Similar to MNIST, we again consider the convolutional neural network LeNet with $657,080$ learning parameters. Thus, the value of dimension $d = 657,080$ in this particular case. The experimental results are shown in Figure 12 with specific details provided in the captions.

Next, we present a *gradient averaging* scheme that aims to reduce the sensitivity of fault-tolerance achieved by a gradient-filter to individual agents' data batch-size k.

3.5 Gradient averaging

We observe from the experiments above, specifically the plots in Figures 4, 5, 6 and 7, that fault-tolerance by using gradient-filters is quite sensitive to the individual agents' data batch-sizes. For instance, consider the case of distributed learning with CGE gradient-filter in presence of faulty agents exhibiting *coordinate-wise* faults shown in Figure 8(b). Although the different between the training losses in presence and in absence of faults is small for more iterations (or steps), the training loss in presence of faults is littered with spikes even in the later stages of the learning process. However, we observe that the magnitude of these spikes attenuates significantly when we increase the data batch-size k from 16 to 64, as shown in Figure 10(b). The reason why this happens is the fact that larger is the batch-size for computing the stochastic gradients smaller is the variance σ^2, and therefore, as per our theoretical analysis presented in Section 2.3, better is the fault-tolerance by the CGE gradient-filter. Similar theoretical observations have also been made for the other gradient-filters [5, 11, 45]. In general, it is safe to say that fault-tolerance of a gradient-filter improves if the variance σ^2 of the stochastic gradients computed by the non-faulty agents reduces.

Motivated from the above observations, we propose a technique of *gradient averaging* below that allows the non-faulty agents to compute stochastic gradients with reduced variance σ^2 without increasing their data batch-size k. Now, it should be noted that for reducing the variance of the stochastic gradients we could also use other existing variance reduction techniques from the stochastic optimization literature [6]. For now, we consider the gradient averaging presented below.

For each iteration t and agent i, let h_i^t denote the weighted average of the stochastic gradients received till the t-th iteration by the server from agent i. Specifically, let $\alpha \in [0,\, 1)$ and let g_i^t denote the stochastic gradient received from agent i in iteration t. Then, for each agent i and iteration t,

$$h_i^t = \alpha\, h_i^{t-1} + (1 - \alpha)\, g_i^t. \tag{25}$$

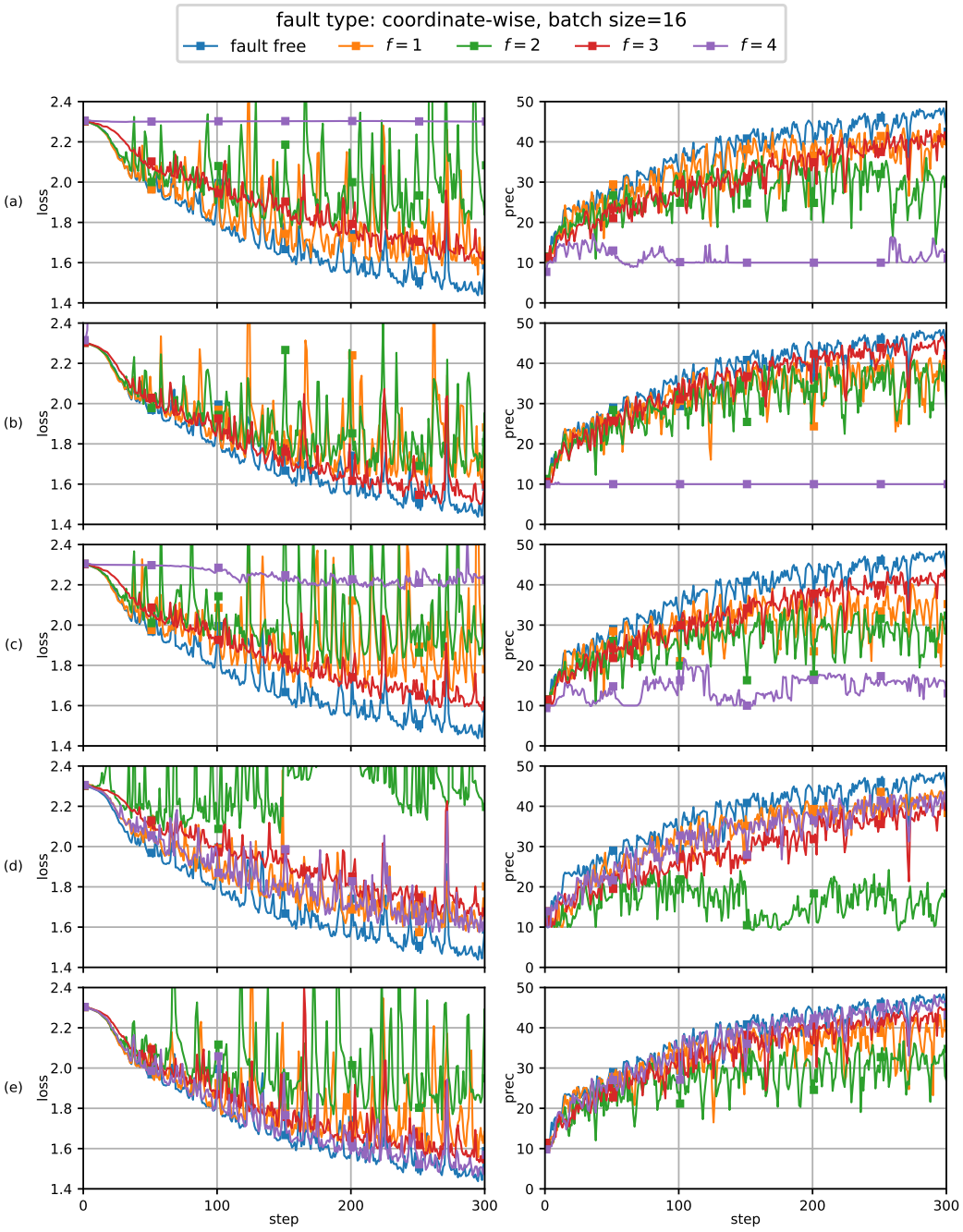

Figure 12: Distributed learning of neural networks for CIFAR-10 dataset using the D-SGD method and the different gradient-filters in presence of $f = 1$ faulty agent, out of $n = 10$ total agents, exhibiting the *coordinate-wise* faults. In these experiments the data batch-size $k = 16$. Different rows correspond to different gradient-filters: (a) geometric median, (b) median of means, (c) coordinate-wise trimmed mean, (d) multi-KRUM with $m = 2$, and (e) CGE.

Given a set of n vectors $y_1, \ldots, y_n$, let $\mathsf{Filter}\{y_1, \ldots, y_n\}$ denote the output of a gradient-filter, such as CGE or multi-KRUM. Then, in step **S2** of each iteration t of the D-SGD algorithm presented in Section 2.1, the server updates the current estimate w^t to

$$w^t = w^{t-1} + \eta_t \cdot \mathsf{Filter}\left\{h_1^t, \ldots, h_n^t\right\}.$$

Recall that η_t is the step-size for iteration 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 for reducing the variance of stochastic gradients.

To evaluate the impact of gradient averaging on fault-tolerance by gradient-filter, we repeat the above experiments on the distributed learning of LeNet neural networks for MNIST and CIFAR-10 datasets. The outcomes of some of our experiments are shown below in Figures 13, 14, 15 and 16. We observe that in general the stability of the learning process improves with gradient averaging, and especially for the case when the faulty agents exhibit the *coordinate-wise* faults, as shown in Figures 13(b), 14(b), 15(b) and 16(b). Moreover, we also observe that the fault-tolerance achieved by a gradient-filter improves with increase in the value of α.

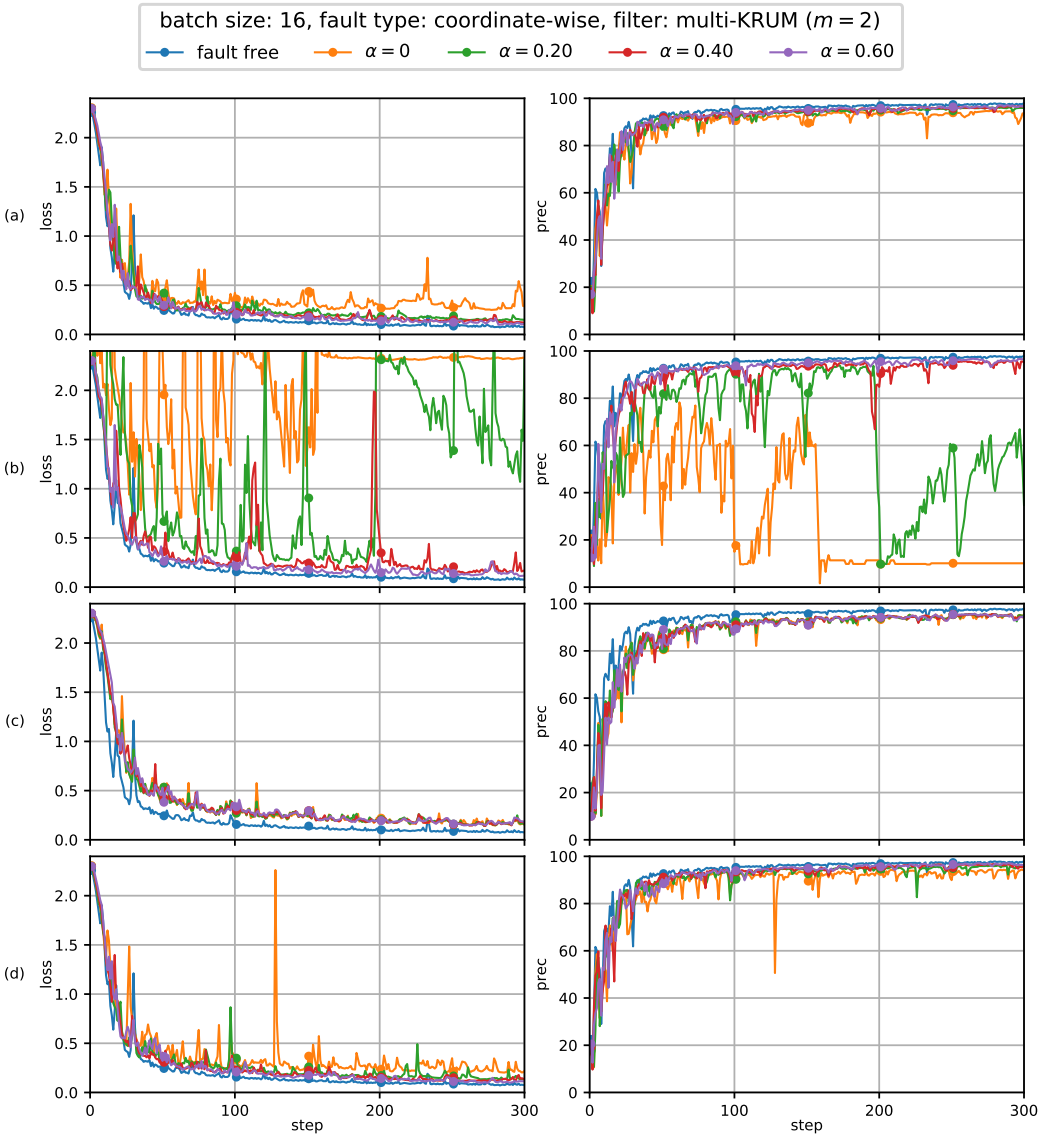

Figure 13: Distributed learning of neural networks for MNIST dataset using the D-SGD method with gradient averaging and the multi-KRUM ($m = 2$) gradient-filter in presence of Byzantine faulty agents exhibiting the *coordinate-wise* faults. The different values of weight α are mentioned at the top. For all these experiments we used data batch-size $k = 16$. Different rows correspond to different number of faults: (a) $f = 1$, (b) $f = 2$, (c) $f = 3$, and (d) $f = 4$.

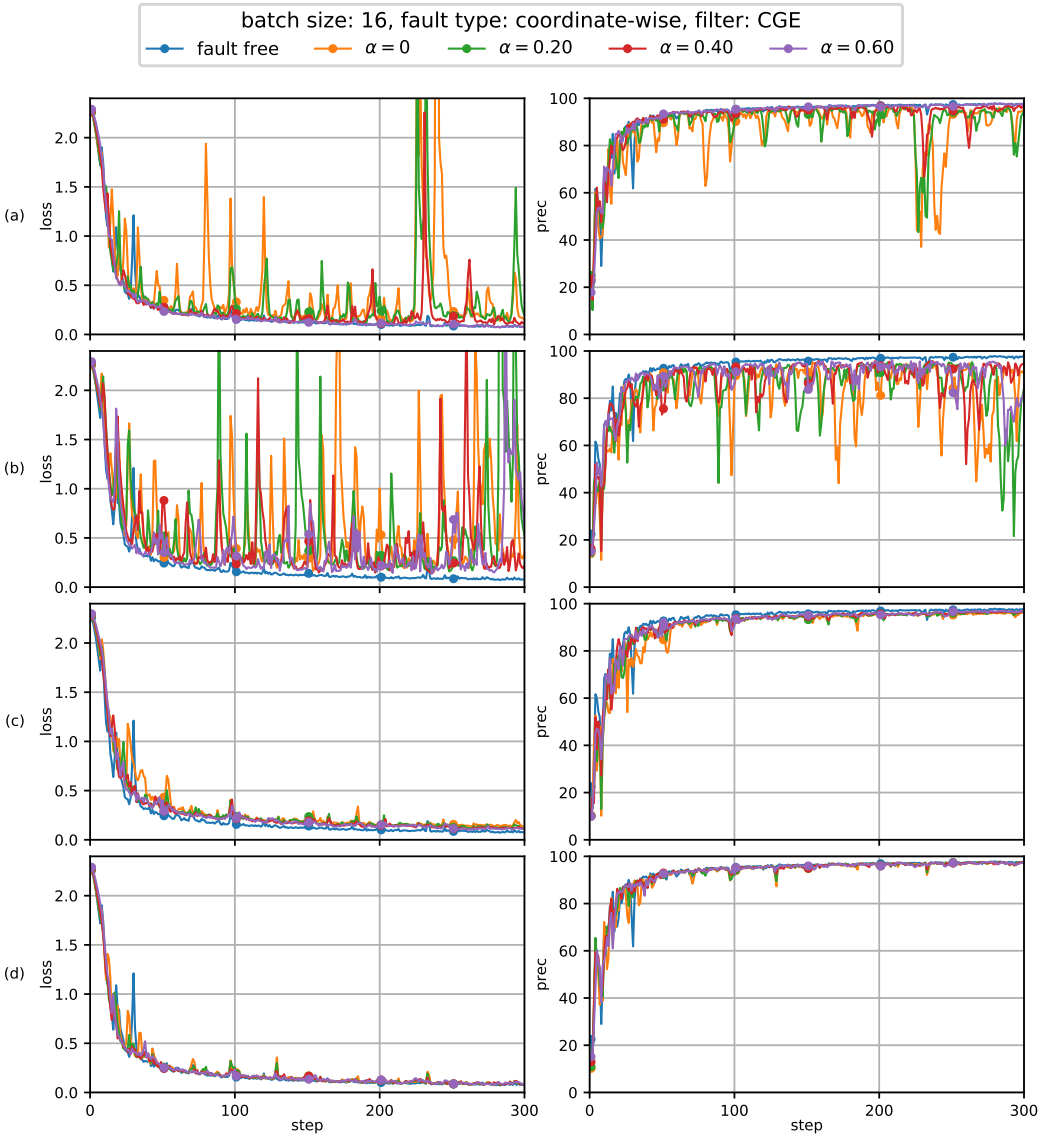

Figure 14: Distributed learning of neural networks for MNIST dataset using the D-SGD method with gradient averaging and the CGE gradient-filter in presence of Byzantine faulty agents exhibiting the *coordinate-wise* faults. The different values of weight α are mentioned at the top. For all these experiments we used data batch-size $k = 16$. Different rows correspond to different number of faults: (a) $f = 1$, (b) $f = 2$, (c) $f = 3$, and (d) $f = 4$.

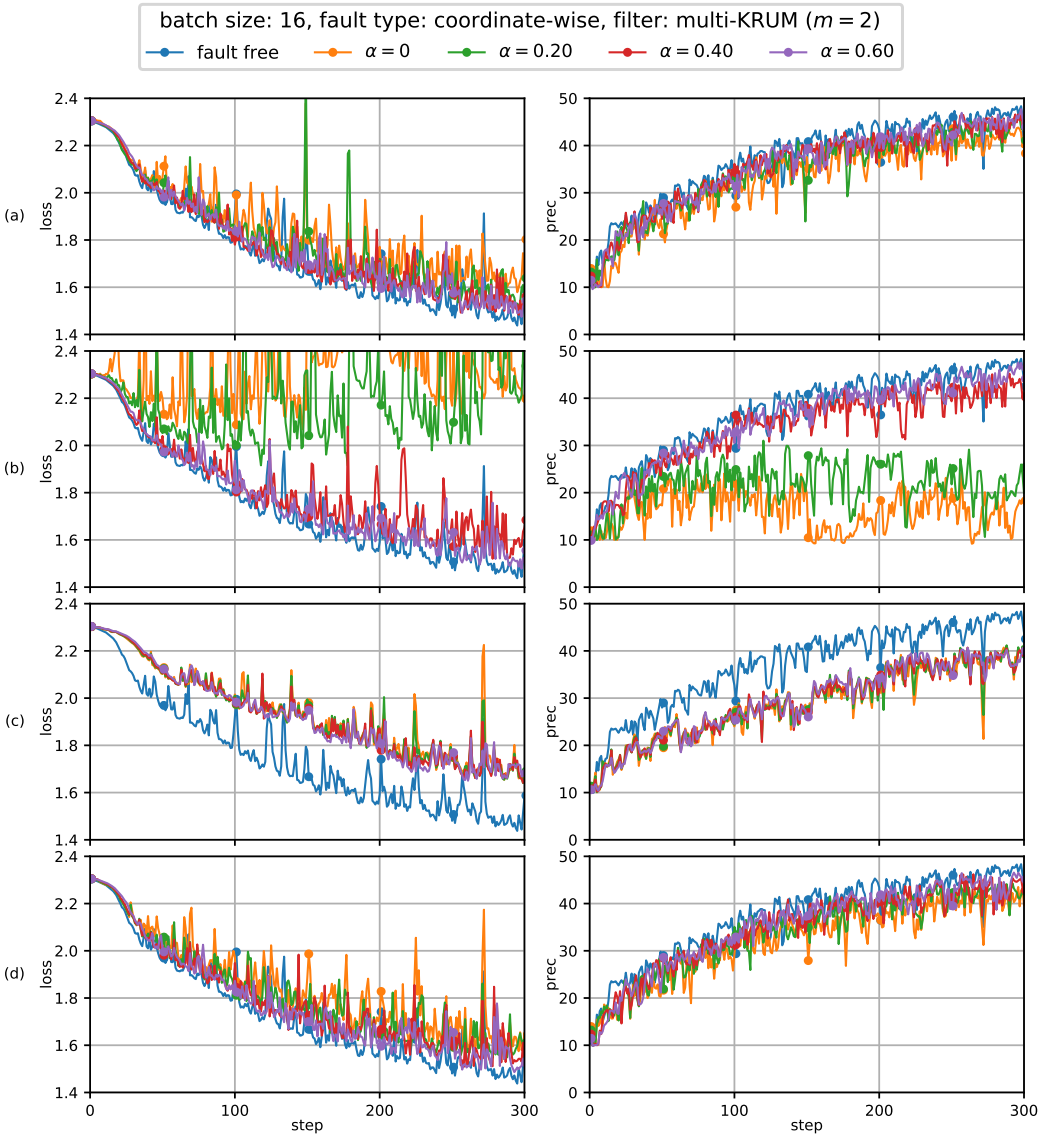

Figure 15: Distributed learning of neural networks for CIFAR-10 dataset using the D-SGD method with gradient averaging and the multi-KRUM ($m = 2$) gradient-filter in presence of Byzantine faulty agents exhibiting the *coordinate-wise* faults. The different values of weight α are mentioned at the top. For all these experiments we used data batch-size $k = 16$. Different rows correspond to different number of faults: (a) $f = 1$, (b) $f = 2$, (c) $f = 3$, and (d) $f = 4$.

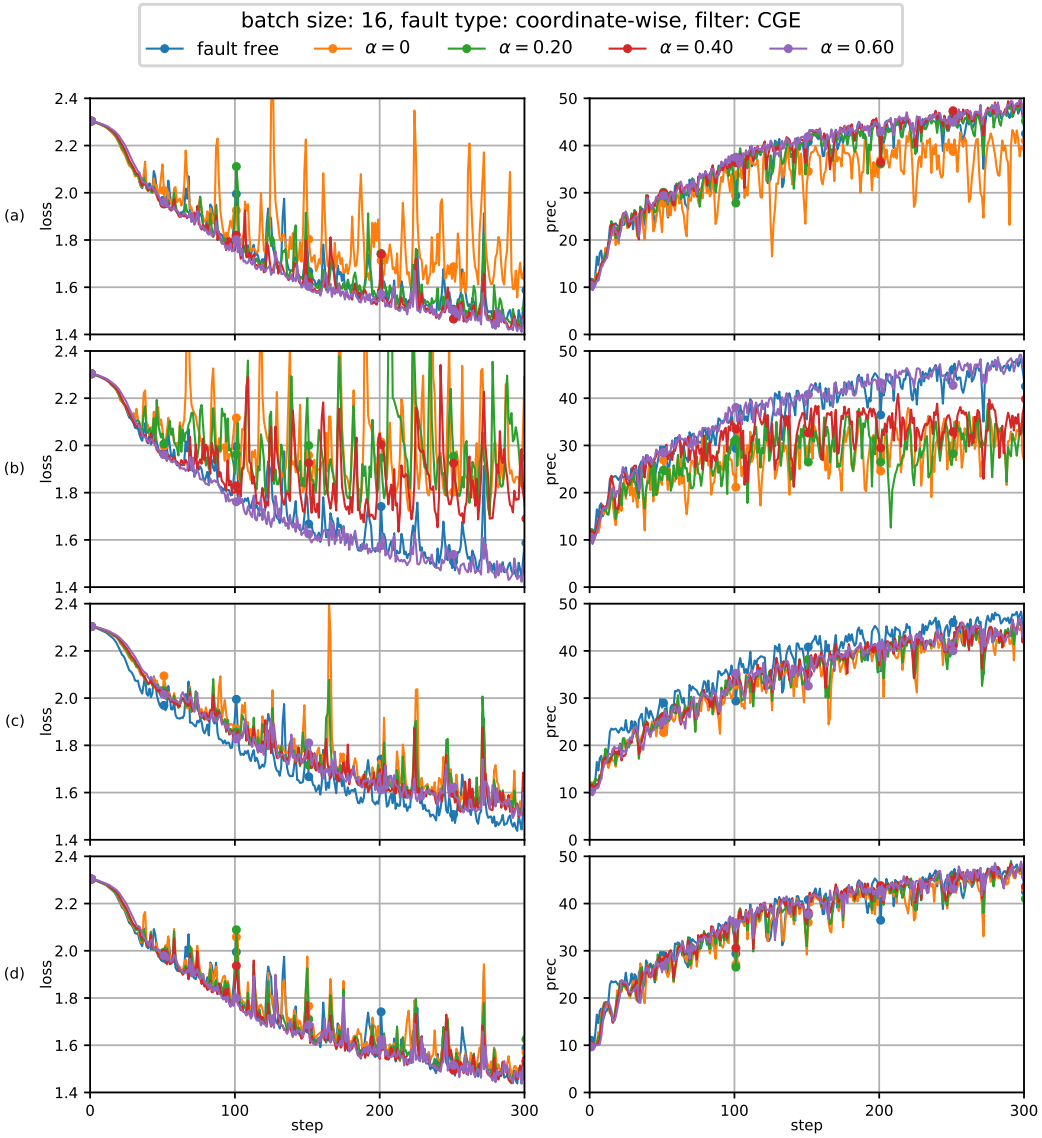

Figure 16: Distributed learning of neural networks for CIFAR-10 dataset using the D-SGD method with gradient averaging and the CGE gradient-filter in presence of Byzantine faulty agents exhibiting the *coordinate-wise* faults. The different values of weight α are mentioned at the top. For all these experiments we used data batch-size $k = 16$. Different rows correspond to different number of faults: (a) $f = 1$, (b) $f = 2$, (c) $f = 3$, and (d) $f = 4$.

4 Summary

In this report, we have considered the problem of Byzantine fault-tolerance in distributed learning using the distributed stochastic gradient descent (D-SGD) method, and a norm based gradient-filter named comparative gradient elimination (CGE). We have shown that the CGE gradient-filter guarantees fault-tolerance against a bounded number of Byzantine faulty agents, if the stochastic gradients computed by the non-faulty agents have bounded variance and the expected loss function is strongly convex with Lipschitz continuous gradients.

Through experimentation we have demonstrated the applicability of our CGE gradient-filter to fault-tolerance in distributed supervised learning of artificial neural networks for image classification. In our experiments we considered two benchmark image classification tasks: MNIST [27] and CIFAR-10 [25]. We have shown that the fault-tolerance achieved by the CGE gradient-filter is comparable to that of other state-of-the-art gradient-filters, namely the multi-KRUM [5], geometric median of means [11], and coordinate-wise trimmed mean [45, 39].

We also have proposed a *gradient averaging* scheme that aims to reduce the sensitivity of a supervised learning process to individual agents' batch-sizes. We have observed that gradient averaging largely improves the fault-tolerance achieved by a gradient-filter, be it the CGE gradient-filter or the multi-KRUM gradient-filter.

Acknowledgements

Research reported in this paper was sponsored in part by the Army Research Laboratory under Cooperative Agreement W911NF- 17-2-0196, and by the National Science Foundation award 1842198. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, National Science Foundation or the U.S. Government.

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 Byzantine fault tolerant. *arXiv preprint arXiv:1810.05291*, 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, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[7] 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.

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

[9] Xinyang Cao and Lifeng Lai. Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers. *IEEE Transactions on Signal Processing*, 67(22):5850–5864, 2019.

[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] Georgios Damaskinos, Rachid Guerraoui, Rhicheek Patra, Mahsa Taziki, et al. Asynchronous Byzantine machine learning (the case of sgd). In *International Conference on Machine Learning*, pages 1153–1162, 2018.

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

[15] 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.

[16] 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.

[17] 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.

[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. Byzantine fault-tolerant parallelized stochastic gradient descent for linear regression. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 415–420. IEEE, 2019.

[20] 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.

[21] Nirupam Gupta and Nitin H Vaidya. Resilience in collaborative optimization: Redundant and independent cost functions. *arXiv preprint arXiv:2003.09675*, 2020.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

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

[24] Peter J Huber. *Robust statistics*. Springer, 2011.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] P. Milasevic and G. R. Ducharme. Uniqueness of the spatial median. *The Annals of Statistics*, 15(3):1332–1333, 1987.

[29] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[30] Jyrki Möttönen, Klaus Nordhausen, Hannu Oja, et al. Asymptotic theory of the spatial median. In *Nonparametrics and Robustness in Modern Statistical Inference and Time Series Analysis: A Festschrift in honor of Professor Jana Jurečková*, pages 182–193. Institute of Mathematical Statistics, 2010.

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

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

[33] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

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

[35] Michael Rabbat and Robert Nowak. Distributed optimization in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 20–27, 2004.

[36] Robin L Raffard, Claire J Tomlin, and Stephen P Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 3, pages 2453–2459. IEEE, 2004.

[37] 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.

[38] 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.

[39] Lili Su and Shahin Shahrampour. Finite-time guarantees for Byzantine-resilient distributed state estimation with noisy measurements. *arXiv preprint arXiv:1810.10086*, 2018.

[40] 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.

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

[42] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized Byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*, 2018.

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

[44] Zhixiong Yang and Waheed U. Bajwa. Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning, 2017.

[45] 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.

A Proof for Theorem 1

In this appendix, we present the rigorous proof for the main result of this paper, Theorem 1, restated below.

Theorem (Review)**.** *Suppose that the Assumptions 1-3 hold true, the fault-tolerance margin α is positive, and the step-size $\eta_t = \eta > 0$ for all t in (7). Recall M^2 from (20). If $\eta < \overline{\eta}$ then the following holds true.*

1. *The value of*

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

is positive and less than 1.

2. *Recall the definition of w^* from (2). Given the initial estimate w^0, that may be chosen arbitrarily from $\mathbb{R}^d$, for all $t \geq 0$,*

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

Throughout the rest of the appendix, the assumptions and the conditions stated in the theorem hold true. First, in Section A.1, we will prove Part 1 of the theorem. Subsequently, in Section A.2, we will prove Part 2 of the theorem.

A.1 Proof of Part 1 of Theorem 1

In this section we will prove that

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

has value in the interval (0, 1).

Recall, from (21), that

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

where $0 < \eta < \overline{\eta}$. As $\eta\,(\overline{\eta} - \eta) > 0$, it is obvious that $\rho < 1$. We show as follows that $\rho > 0$.

Note that for every $\eta \in (0, \overline{\eta})$ there exists a real value δ in $(0, \overline{\eta})$ such that

$$\eta = \overline{\eta} - \delta.$$

Substituting η from above in (21) we obtain that

$$\begin{aligned}
\rho &= 1 - \left(n^2 + (n-f)^2\mu^2\right)\, (\overline{\eta} - \delta)\delta \\
&= \left(n^2 + (n-f)^2\mu^2\right) \delta^2 - \overline{\eta}\left(n^2 + (n-f)^2\mu^2\right) \delta + 1. \\
&= \left(n^2 + (n-f)^2\mu^2\right) \left(\delta - \frac{\overline{\eta}}{2}\right)^2 + 1 - \frac{\overline{\eta}^2\left(n^2 + (n-f)^2\mu^2\right)}{4}. \qquad (26)
\end{aligned}$$

As $\left(\delta - \frac{\overline{\eta}}{2}\right)^2 \geq 0$, with equality when $\delta = \overline{\eta}/2$, (26) implies that

$$\rho \geq 1 - \frac{\overline{\eta}^2 \left(n^2 + (n-f)^2\mu^2\right)}{4}. \tag{27}$$

Substituting $\overline{\eta}$ from (19) in (28) we obtain that

$$\rho \geq 1 - \frac{(2\lambda+\mu)^2 n^2 \alpha^2}{(n^2 + (n-f)^2\mu^2)}. \tag{28}$$

Recall, from (16), that

$$\alpha = \frac{\lambda n - f(2\lambda+\mu)}{n(2\lambda+\mu)}. \tag{29}$$

Substituting from above in (28) we obtain that

$$\rho \geq 1 - \frac{(\lambda n - f(2\lambda+\mu))^2}{(n^2 + (n-f)^2\mu^2)} = 1 - \frac{((n-f)\lambda - f(\lambda+\mu))^2}{(n^2 + (n-f)^2\mu^2)}. \tag{30}$$

Now, as $\alpha > 0$, from (29) we obtain that

$$(n-f)\lambda - f(\lambda+\mu) > 0. \tag{31}$$

Also, note that

$$(n-f)\lambda - f(\lambda+\mu) \leq (n-f)\lambda. \tag{32}$$

From (31) and (32) we obtain that

$$((n-f)\lambda - f(\lambda+\mu))^2 \leq (n-f)^2\lambda^2. \tag{33}$$

Substituting from (33) in (30) we obtain that

$$\rho \geq 1 - \frac{(n-f)^2\lambda^2}{(n^2 + (n-f)^2\mu^2)}. \tag{34}$$

We will now show below that if Assumptions 2 and 3 hold true then $\lambda \leq \mu$.

Consider a point w^* defined in (2), and an arbitrary finite $w \in \mathbb{R}^d$. Note that $\nabla Q(w^*) = 0$. Thus, Assumption 2 implies that

$$\|\nabla Q(w)\| \leq \mu \|w - w^*\|. \tag{35}$$

Now, Assumption 3 implies that

$$\langle w - w^*, \nabla Q(w)\rangle \geq \lambda \|w - w^*\|^2. \tag{36}$$

From Cauchy-Schwartz inequality we note that $\langle w - w^*, \nabla Q(w)\rangle \leq \|w - w^*\| \|\nabla Q(w)\|$. Using this in (36) implies that

$$\|Q(w)\| \geq \lambda \|w - w^*\|. \tag{37}$$

From (35) and (37) we obtain that $\lambda \leq \mu$. Upon using this inequality in (34) we obtain that

$$\rho \geq 1 - \frac{(n-f)^2\mu^2}{(n^2 + (n-f)^2\mu^2)}. \tag{38}$$

As $n > 0$, $(n-f)^2\mu^2 < n^2 + (n-f)^2\mu^2$. Thus, (38) implies that $\rho > 0$.

Next, we present the proof for (22).

A.2 Proof of Part 2 of Theorem 1

In this section, we will prove Part 2 of Theorem 1, 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. Throughout the proof, let notation $|\cdot|$ denote the set cardinality. Recall the following notation introduced in Section 2.3.

- For each agent i and iteration t, g_i^t denotes the stochastic gradient sent by an agent i in iteration t to the server. If agent i is non-faulty then g_i^t is computed as per the method described in Section 2.2. Otherwise, if agent i is faulty then g_i^t may be an arbitrary vector from $\mathbb{R}^d$.
- In each iteration t, each non-faulty agent samples k data points $\{z_{i_1}^t, \ldots, z_{i_k}^t\}$ that are independent and identically distributed (i.i.d.) according to distribution $\mathcal{D}$. Recall from (9) that
$$\mathbf{z}_i^t = \{z_{i_1}^t, \ldots, z_{i_k}^t\}.$$
- Recall from (10) that, for each agent i,
$$\zeta_i^t = \begin{cases} \mathbf{z}_i^t & , \quad i \text{ is non-faulty} \\ g_i^t & , \quad i \text{ is faulty} \end{cases}$$
- Recall from (11) that
$$\zeta^t = \left\{ \zeta_1^t, \ldots, \zeta_n^t \right\}.$$
- Recall from Section 2.3 that for each agent i and iteration t, $\mathbb{E}_{\zeta_i^t}(\cdot)$ denotes the expected value of a function of the random variable ζ_i^t given the current estimate w^t. Let
$$\mathbb{E}_{\zeta^t}(\cdot) = \mathbb{E}_{\zeta_1^t, \ldots, \zeta_n^t}(\cdot). \tag{39}$$
- Finally, recall (18) that $\mathbb{E}_t(\cdot)$ denotes the joint expectation of a random variable that is a function of the random variables $\zeta^0, \ldots, \zeta^t$ given the initial estimate w^0. Specifically,
$$\mathbb{E}_t(\cdot) = \mathbb{E}_{\zeta^0, \ldots, \zeta^t}(\cdot).$$

Note that for each non-faulty agent i, and a deterministic real-valued function Ψ,

$$\mathbb{E}_{\zeta^t} \Psi\left(g_i^t\right) = \mathbb{E}_{\zeta_1^t, \ldots, \zeta_n^t} \Psi\left(g_i^t\right). \tag{40}$$

Note that

$$\zeta_i^t = \mathbf{z}_i^t \tag{41}$$

for each non-faulty agent i. Now, given the current estimate w^t, recall from Section 2.2 that 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, (41) implies that for each non-faulty agent i,

$$\mathbb{E}_{\zeta_1^t,\ldots,\zeta_n^t}\Psi\left(g_i^t\right) = \mathbb{E}_{\mathbf{z}_i^t}\Psi\left(g_i^t\right). \tag{42}$$

Upon substituting from (42) in (40) we obtain that for each non-faulty agent i,

$$\mathbb{E}_{\zeta^t}\Psi\left(g_i^t\right) = \mathbb{E}_{\mathbf{z}_i^t}\Psi\left(g_i^t\right). \tag{43}$$

The above observation, i.e., (43) is used in the later stages of the proof, and also in proving the following two relevant implications, Lemmas 1 and 2, of Assumption 1.

Recall, from Assumption 1, that σ^2 is the upper bound on the variance of stochastic gradients computed by the non-faulty agents.

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

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

Note that, as there are at most f Byzantine faulty agents, there are at least $n-f$ non-faulty agents in the system. We define a set $\mathcal{H}$ with $|\mathcal{H}| = n-f$ that comprises of only non-faulty agents that may be chosen arbitrarily. Note that if $i \in \mathcal{H}$ then i is a non-faulty agent. Let $\mathcal{B} = \{1,\ldots,n\} \setminus \mathcal{H}$ denote the remaining f agents, some of which may be non-faulty. However, note that the set of faulty agents is a subset of $\mathcal{B}$.

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

$$\mathbb{E}_{\zeta^t}\left\|g_{\nu_t}^t\right\| \leq \sigma\left(1+\sqrt{n-f-1}\right) + \left\|\nabla Q(w^t)\right\|. \tag{44}$$

The proofs of Lemma 1 and Lemma 2 are presented in Appendix B. Note that the proof of Lemma 2 relies on an existing result for an upper bound on the highest-order statistic's expected value [2, 4].

Now, recall from (6) in the algorithm that for $j \in \{1,\ldots,n\}$ the stochastic gradient with the j-th smallest norm, $g_{i_j}^t$, is sent by agent i_j. Let

$$\mathfrak{g}^t = \sum_{j\in\{i_1,\ldots,i_{n-f}\}} g_j^t \tag{45}$$

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 (45), in (7) we obtain that

$$w^{t+1} = w^t - \eta\,\mathfrak{g}^t, \quad \forall t. \tag{46}$$

Thus, from the definition of Euclidean norm (5),

$$\left\|w^{t+1}-w^*\right\|^2=\left\|w^t-w^*\right\|^2-2\eta\left\langle w^t-w^*,\,\mathfrak{g}^t\right\rangle+\eta^2\left\|\mathfrak{g}^t\right\|^2. \tag{47}$$

Now, owing to the triangle inequality, we obtain that

$$\left\|\mathfrak{g}^t\right\|\leq\sum_{j\in\{i_1,\ldots,i_{n-f}\}}\left\|g_j^t\right\|. \tag{48}$$

Note that $|\mathcal{H}|=n-f$ and $\{i_1,\ldots,i_{n-f}\}$ represents the set of agents that have stochastic gradients with smallest $n-f$ norm. It follows that

$$\sum_{j\in\{i_1,\ldots,i_{n-f}\}}\left\|g_j^t\right\|\leq\sum_{j\in\mathcal{H}}\left\|g_j^t\right\|. \tag{49}$$

Substituting from (49) in (48) we obtain that

$$\left\|\mathfrak{g}^t\right\|\leq\sum_{j\in\mathcal{H}}\left\|g_j^t\right\|. \tag{50}$$

Note that, as $(\cdot)^2$ is a convex function, for p real values $a_1,\ldots,a_p$,

$$\left(\sum_{j=1}^{p}a_j\right)^2\leq p\sum_{j=1}^{p}a_j^2.$$

Using the above fact in (50) we obtain that

$$\left\|\mathfrak{g}^t\right\|^2\leq|\mathcal{H}|\sum_{j\in\mathcal{H}}\left\|g_j^t\right\|^2=(n-f)\sum_{j\in\mathcal{H}}\left\|g_j^t\right\|^2.$$

Substituting from above in (47) implies that

$$\left\|w^{t+1}-w^*\right\|^2\leq\left\|w^t-w^*\right\|^2-2\eta\left\langle w^t-w^*,\,\mathfrak{g}^t\right\rangle+\eta^2(n-f)\sum_{j\in\mathcal{H}}\left\|g_j^t\right\|^2. \tag{51}$$

We obtain below a lower bound on the inner product $\left\langle w^t-w^*,\,\mathfrak{g}^t\right\rangle$ in terms of the stochastic gradients sent by the non-faulty agents in the set $\mathcal{H}$.

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

$$\left|\mathcal{H}^t\right|\geq|\mathcal{H}|-f=n-2f,\quad\text{and}\quad\left|\mathcal{B}^t\right|\leq f. \tag{52}$$

Therefore, recalling from (45),

$$\mathfrak{g}^t=\sum_{i\in\mathcal{H}^t}g_i^t+\sum_{j\in\mathcal{B}^t}g_j^t. \tag{53}$$

Thus,

$$\left\langle w^t-w^*,\,\mathfrak{g}^t\right\rangle=\sum_{i\in\mathcal{H}^t}\left\langle w^t-w^*,\,g_i^t\right\rangle+\sum_{j\in\mathcal{B}^t}\left\langle w^t-w^*,\,g_j^t\right\rangle. \tag{54}$$

Recall that the set $\mathcal{B}_j^t$ may contain some faulty agents. Therefore, for $j \in \mathcal{B}_j^t$ the vector g_j^t may not be a correct stochastic gradient. Thus, we now obtain an upper bound on the inner product norm $\left\langle w^t - w^*,\, g_j^t \right\rangle$ for each $j \in \mathcal{B}^t$ in terms of the norm of the largest stochastic gradient in set $\mathcal{H}$, denoted by ν_t in Lemma 2 presented above.

Owing to Cauchy-Schwartz inequality, for all j we note that

$$\left\langle w^t - w^*,\, g_j^t \right\rangle \geq - \left\| w^t - w^* \right\| \, \left\| g_j^t \right\|. \tag{55}$$

Recall, from Lemma 2, that for iteration t agent $\nu_t \in \mathcal{H}$ sends stochastic gradient with the largest norm amongst the agents in $\mathcal{H}$. Thus, $\left\| g_{i_{n-f}}^t \right\| \leq \left\| g_{\nu_t}^t \right\|$, and

$$\left\| g_j^t \right\| \leq \left\| g_{\nu_t}^t \right\|, \quad \forall j \in \mathcal{B}^t. \tag{56}$$

Substituting from (56) in (55) we obtain that

$$\left\langle w^t - w^*,\, g_j^t \right\rangle \geq - \left\| w^t - w^* \right\| \, \left\| g_{\nu_t}^t \right\|. \tag{57}$$

Upon substituting from (57) in (54) 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 $\left| \mathcal{B}^t \right| \leq f$ (see (52)), upon replacing the summation by factor f in the second term on the right hand side above we obtain that

$$\left\langle w^t - w^*,\, \mathfrak{g}^t \right\rangle \geq \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*,\, g_i^t \right\rangle - f \left\| w^t - w^* \right\| \, \left\| g_{\nu_t}^t \right\|. \tag{58}$$

We define,

$$\phi_t = \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*,\, g_i^t \right\rangle - f \left\| w^t - w^* \right\| \, \left\| g_{\nu_t}^t \right\|. \tag{59}$$

Note that, upon substituting from (59) in (58),

$$\left\langle w^t - w^*,\, \mathfrak{g}^t \right\rangle \geq \phi_t.$$

Substituting from above in (51) we obtain that

$$\left\| w^{t+1} - w^* \right\|^2 \leq \left\| w^t - w^* \right\|^2 - 2\eta \, \phi_t + \eta^2 \, (n - f) \sum_{j \in \mathcal{H}} \left\| g_j^t \right\|^2. \tag{60}$$

Recall, from Section 2.3, that for each iteration t, w^{t+1} is a function of random variables $\zeta_1^t, \ldots, \zeta_n^t$ given w^t. Also, recall from definition (39) above that $\mathbb{E}_{\zeta^t}$ the expectation of a function of random variables $\zeta_1^t, \ldots, \zeta_n^t$ given w^t. Thus, upon taking the expectation $\mathbb{E}_{\zeta^t}$ on both sides in (60) we obtain that

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

As $\mathbb{E}_{\zeta^t} \left\|w^t - w^*\right\|^2 = \left\|w^t - w^*\right\|^2$, from above we obtain that

$$\mathbb{E}_{\zeta^t} \left\|w^{t+1} - w^*\right\|^2 \leq \left\|w^t - w^*\right\|^2 - 2\eta\, \mathbb{E}_{\zeta^t}\left(\phi_t\right) + \eta^2 (n-f) \sum_{j \in \mathcal{H}} \mathbb{E}_{\zeta^t} \left\|g_j^t\right\|^2. \tag{61}$$

We now obtain below a lower and an upper bounds, respectively, for $\mathbb{E}_{\zeta^t}\left(\phi_t\right)$ and $\sum_{j \in \mathcal{H}} \mathbb{E}_{\zeta^t} \left\|g_j^t\right\|^2$, in terms of $\left\|w^t - w^*\right\|$.

Note, from (59), that

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

Therefore,

$$\mathbb{E}_{\zeta^t}\left(\phi_t\right) = \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*,\, \mathbb{E}_{\zeta^t}\left(g_i^t\right) \right\rangle - f \left\|w^t - w^*\right\| \mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\|. \tag{62}$$

Recall from (43) that, for all $i \in \mathcal{H}$,

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

where $\mathbf{z}_i^t$ represent the data points sampled by the non-faulty agent i in iteration t. As shown by (15) in Section 2.3, $\mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right) = \nabla Q(w^t)$ for all $i \in \mathcal{H}$. Using this above we obtain that

$$\mathbb{E}_{\zeta^t}\left(g_i^t\right) = \nabla Q(w^t), \qquad \forall i \in \mathcal{H}.$$

Substituting the above in (62) implies that

$$\mathbb{E}_{\zeta^t}\left(\phi_t\right) = \sum_{i \in \mathcal{H}^t} \left\langle w^t - w^*,\, \nabla Q(w^t) \right\rangle - f \left\|w^t - w^*\right\| \mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\|. \tag{63}$$

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

$$\left\langle w^t - w^*,\, \nabla Q(w^t) \right\rangle \geq \lambda \left\|w^t - w^*\right\|^2. \tag{64}$$

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

$$\mathbb{E}_{\zeta^t}\left(\phi_t\right) \geq \left|\mathcal{H}^t\right| \lambda \left\|w^t - w^*\right\|^2 - f \left\|w^t - w^*\right\| \mathbb{E}_{\zeta^t} \left\|g_{\nu_t}^t\right\|. \tag{65}$$

From Lemma 2 we know that

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

Substituting from above in (65) we obtain that

$$\mathbb{E}_{\zeta^t}\left(\phi_t\right) \geq \left|\mathcal{H}^t\right| \lambda \left\|w^t - w^*\right\|^2 - f \left\|w^t - w^*\right\| \left(\sigma\left(1 + \sqrt{n-f-1}\right) + \left\|\nabla Q(w^t)\right\|\right).$$

Recall from (52) that $\left|\mathcal{H}^t\right| \geq n - 2f$. Using this above we obtain that

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

Upon rearranging the right hand side above we obtain that

$$\mathbb{E}_{\zeta^t}(\phi_t) \geq (n\lambda - f(2\lambda+\mu))\left\|w^t - w^*\right\|^2 - f\sigma\left(1+\sqrt{n-f-1}\right)\left\|w^t - w^*\right\|. \quad (66)$$

Now, owing to Lemma 1,

$$\mathbb{E}_{\zeta^t}\left\|g_j^t\right\|^2 \leq \sigma^2 + \left\|\nabla Q(w^t)\right\|^2, \quad \forall j \in \mathcal{H}.$$

Recall that $|\mathcal{H}| = n - f$. Thus,

$$\sum_{j\in\mathcal{H}} \mathbb{E}_{\zeta^t}\left\|g_j^t\right\|^2 \leq |\mathcal{H}|\left(\sigma^2 + \left\|\nabla Q(w^t)\right\|^2\right) = (n-f)\left(\sigma^2 + \left\|\nabla Q(w^t)\right\|^2\right). \quad (67)$$

As $\nabla Q(w^*) = 0$, Assumption 2 (i.e., Lipschitz continuity of $\nabla Q(w)$) implies that $\left\|\nabla Q(w^t)\right\| \leq \mu\left\|w^t - w^*\right\|$. Using this in (67) implies that,

$$\sum_{j\in\mathcal{H}} \mathbb{E}_{\zeta^t}\left\|g_j^t\right\|^2 \leq (n-f)\left(\sigma^2 + \mu^2\left\|w^t - w^*\right\|^2\right). \quad (68)$$

Finally, substituting from (66) and (68) in (61) we obtain that

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

Upon re-arranging the right hand side above we obtain that

$$\begin{aligned}\mathbb{E}_{\zeta^t}\left\|w^{t+1} - w^*\right\|^2 \leq \left(1 - 2\eta\left(n\lambda - f(2\lambda+\mu)\right) + \eta^2(n-f)^2\mu^2\right)\left\|w^t - w^*\right\|^2\\ + 2\eta f\sigma\left(1+\sqrt{n-f-1}\right)\left\|w^t - w^*\right\| + \eta^2(n-f)^2\sigma^2.\end{aligned} \quad (69)$$

Note that for two arbitrary real values a and b, $2ab \leq a^2 + b^2$. Therefore,

$$\begin{aligned}2\eta f\sigma\left(1+\sqrt{n-f-1}\right)\left\|w^t - w^*\right\| \leq \eta^2 n^2\left\|w^t - w^*\right\|^2\\ + \left(\frac{f}{n}\right)^2\sigma^2\left(1+\sqrt{n-f-1}\right)^2.\end{aligned} \quad (70)$$

Substituting from (70) in (69) we obtain that

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

Substituting M^2 from (20) above we obtain that

$$\begin{aligned}\mathbb{E}_{\zeta^t}\left\|w^{t+1} - w^*\right\|^2 \leq \left\{1 - 2\eta\left(n\lambda - f(2\lambda+\mu)\right) + \eta^2\left(n^2 + (n-f)^2\mu^2\right)\right\}\left\|w^t - w^*\right\|^2\\ + \mathsf{M}^2.\end{aligned} \quad (71)$$

Substituting α from (16), we obtain that

$$n\lambda - f(2\lambda+\mu) = (2\lambda+\mu)n\,\alpha. \tag{72}$$

Therefore,

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

Substituting $\overline{\eta}$ from (19) above we obtain that

$$2\eta\left(n\lambda - f(2\lambda+\mu)\right) - \eta^2\left(n^2+(n-f)^2\mu^2\right) = \left(n^2+(n-f)^2\mu^2\right)\eta\left(\overline{\eta}-\eta\right). \tag{73}$$

Substituting ρ from (21) in (73) we obtain that

$$2\eta\left(n\lambda - f(2\lambda+\mu)\right) - \eta^2\left(n^2+(n-f)^2\mu^2\right) = 1-\rho. \tag{74}$$

Substituting from (74) in (71) we obtain that

$$\mathbb{E}_{\zeta^t}\left\|w^{t+1}-w^*\right\|^2 \le \rho\left\|w^t-w^*\right\|^2 + \mathsf{M}^2. \tag{75}$$

Recall from (18) that $\mathbb{E}_0 = \mathbb{E}_{\zeta^0}$. Thus, the above proves the theorem for $t=0$, i.e.,

$$\mathbb{E}_0\left\|w^1-w^*\right\|^2 \le \rho\left\|w^0-w^*\right\|^2 + \mathsf{M}^2. \tag{76}$$

We now assume below that t in (75) is positive.

From Section 2.3, 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, from the above argument we obtain that $\left\|w^{t+1}-w^*\right\|^2$ is a function of random variables $\zeta^0,\ldots,\zeta^{t-1}$, given the initial estimate w^0. Let, for all $t>0$,

$$\mathbb{E}_{\zeta^t|\zeta^0,\ldots,\zeta^{t-1}}\left\|w^{t+1}-w^*\right\|^2 \tag{77}$$

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,

$$\mathbb{E}_{\zeta^t}\left\|w^{t+1}-w^*\right\|^2 = \mathbb{E}_{\zeta^t|\zeta^0,\ldots,\zeta^{t-1}}\left\|w^{t+1}-w^*\right\|^2, \quad \forall t>0. \tag{78}$$

Substituting from (78) in (75) we obtain that, given w^0,

$$\mathbb{E}_{\zeta^t|\zeta^0,\ldots,\zeta^{t-1}}\left\|w^{t+1}-w^*\right\|^2 \le \rho\left\|w^t-w^*\right\|^2 + \mathsf{M}^2, \quad \forall t>0. \tag{79}$$

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

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

Substituting from (79) 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 (18) 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

$$\mathbb{E}_t\left\|w^{t+1}-w^*\right\|^2\leq\rho\,\mathbb{E}_{t-1}\left\|w^t-w^*\right\|^2+\mathsf{M}^2,\quad\forall t>0. \tag{80}$$

Finally, we use (80) above and reasoning by induction to prove the convergence result (22), i.e,

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

Recall that (22) is trivially true for $t=0$ due to (76) above. Now, suppose that (22) holds true for $t=\tau-1$ where τ is an arbitrary integer of value greater than or equal to 2. Specifically,

$$\mathbb{E}_{\tau-1}\left\|w^\tau-w^*\right\|^2\leq\rho^\tau\left\|w^0-w^*\right\|^2+\left(\frac{1-\rho^\tau}{1-\rho}\right)\mathsf{M}^2. \tag{81}$$

We show below that (22) holds true for τ.

From (80) we obtain that

$$\mathbb{E}_\tau\left\|w^{\tau+1}-w^*\right\|^2\leq\rho\,\mathbb{E}_{\tau-1}\left\|w^\tau-w^*\right\|^2+\mathsf{M}^2. \tag{82}$$

Substituting from (81) above we obtain that

$$\begin{aligned}\mathbb{E}_\tau\left\|w^{\tau+1}-w^*\right\|^2&\leq\rho\left(\rho^\tau\left\|w^0-w^*\right\|^2+\left(\frac{1-\rho^\tau}{1-\rho}\right)\mathsf{M}^2\right)+\mathsf{M}^2\\ &=\rho^{\tau+1}\left\|w^0-w^*\right\|^2+\left(\rho\left(\frac{1-\rho^\tau}{1-\rho}\right)+1\right)\mathsf{M}^2\\ &=\rho^{\tau+1}\left\|w^0-w^*\right\|^2+\left(\frac{1-\rho^{\tau+1}}{1-\rho}\right)\mathsf{M}^2.\end{aligned} \tag{83}$$

Inequality (83) above shows that (22) holds true for $t=\tau$. Reasoning by induction, the above proves (22), i.e.,

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

B Proofs of Lemma 1 and Lemma 2

In this appendix, we present the proofs for Lemma 1 and 2, respectively.

The proofs rely on observation (43) made in Appendix A, according to which for an arbitrary deterministic multi-variate real-valued function Ψ, for each non-faulty agent i we have

$$\mathbb{E}_{\zeta^t} \Psi\left(g_i^t\right) = \mathbb{E}_{\zeta_i^t} \Psi\left(g_i^t\right). \tag{84}$$

We now present the proof of Lemma 1 restated below.

Lemma. *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. From the definition of Euclidean norm (5), note that for each iteration t,

$$\left\|g_i^t - \mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2 = \left\|g_i^t\right\|^2 - 2\left\langle g_i^t, \mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\rangle + \left\|\mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2. \tag{85}$$

As the expected value of a constant is the constant itself, upon taking expectations on both sides in (85) we obtain that

$$\mathbb{E}_{\zeta^t} \left\|g_i^t - \mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2 = \mathbb{E}_{\zeta^t} \left\|g_i^t\right\|^2 - \left\|\mathbb{E}_{\zeta^t}\left(g_i^t\right)\right\|^2. \tag{86}$$

Note, from (84), that

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

Substituting the above in (86) we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t - \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|^2 = \mathbb{E}_{\zeta^t} \left\|g_i^t\right\|^2 - \left\|\mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|^2. \tag{87}$$

Recall from (15) in Section 2.3 that $\mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right) = \nabla Q(w^t)$. Substituting this above we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\|g_i^t - \mathbb{E}_{\mathbf{z}_i^t}\left(g_i^t\right)\right\|^2 = \mathbb{E}_{\zeta^t} \left\|g_i^t\right\|^2 - \left\|\nabla Q(w^t)\right\|^2. \tag{88}$$

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 the above in (88) we obtain that

$$\mathbb{E}_{\zeta^t} \left\|g_i^t\right\|^2 \leq \sigma^2 + \left\|\nabla Q(w^t)\right\|^2, \quad \forall t \geq 0.$$

As i is an arbitrary non-faulty agent, the above proves the lemma. □

We now present the proof of Lemma 2 restated below.

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

$$\left\|g^t_{\nu_t}\right\| \geq \left\|g^t_i\right\|, \quad \forall i \in \mathcal{H}.$$

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

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

Proof. We begin the 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 owing to Arnold and Groeneveld [2],

$$\mathbb{E}(R_\nu) \leq \mathbb{E}(R) + \sqrt{\mathsf{Var}(R)\,(p-1)}. \tag{89}$$

Now, the proof presented below relies on the above result.

Consider an arbitrary iteration t. Recall that $\mathcal{H}$ comprises of only non-faulty agents, specifically $n-f$ non-faulty agents. Thus, from (84) we obtain that, for all $i \in \mathcal{H}$,

$$\mathbb{E}_{\zeta^t} \left\|g^t_i\right\| = \mathbb{E}_{\mathbf{z}^t_i} \left\|g^t_i\right\|, \tag{90}$$

and

$$\mathbb{E}_{\zeta^t} \left(\left\|g^t_i\right\| - \mathbb{E}_{\zeta^t} \left\|g^t_i\right\|\right)^2 = \mathbb{E}_{\mathbf{z}^t_i} \left(\left\|g^t_i\right\| - \mathbb{E}_{\mathbf{z}^t_i} \left\|g^t_i\right\|\right)^2. \tag{91}$$

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

$$\left\|g^t_{\nu_t}\right\| = \max\left\{\left\|g^t_i\right\|,\ i \in \mathcal{H}\right\}.$$

Therefore, substituting from (90) and (91) in (89), we obtain that for an arbitrary agent $i \in \mathcal{H}$,

$$\mathbb{E}_{\zeta^t} \left\|g^t_{\nu_t}\right\| \leq \mathbb{E}_{\mathbf{z}^t_i} \left\|g^t_i\right\| + \sqrt{\mathbb{E}_{\mathbf{z}^t_i} \left(\left\|g^t_i\right\| - \mathbb{E}_{\mathbf{z}^t_i} \left\|g^t_i\right\|\right)^2 (n-f-1)}. \tag{92}$$

We now show below that

$$\mathbb{E}_{\mathbf{z}^t_i} \left\|g^t_i\right\| \leq \sigma + \left\|\nabla Q(w^t)\right\|, \quad \forall i \in \mathcal{H}. \tag{93}$$

Owing to Jensen's inequality [8], for all $i \in \mathcal{H}$,

$$\mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\| = \mathbb{E}_{\mathbf{z}_i^t} \sqrt{\left\| g_i^t \right\|^2} \leq \sqrt{\mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\|^2}. \tag{94}$$

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

$$\mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\|^2 \leq \sigma^2 + \left\| \nabla Q(w^t) \right\|^2, \quad \forall i \in \mathcal{H}. \tag{95}$$

Substituting from (95) in (94) we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\| \leq \sqrt{\sigma^2 + \left\| \nabla Q(w^t) \right\|^2}, \quad \forall i \in \mathcal{H}. \tag{96}$$

From triangle inequality,

$$\sqrt{\sigma^2 + \left\| \nabla Q(w^t) \right\|^2} \leq \sigma + \left\| \nabla Q(w^t) \right\|.$$

Substituting the above in (96) proves (93), i.e,

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

Next, we show that

$$\mathbb{E}_{\mathbf{z}_i^t} \left(\left\| g_i^t \right\| - \mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\| \right)^2 \leq \sigma^2, \quad \forall i \in \mathcal{H}. \tag{97}$$

Note that for all i,

$$\mathbb{E}_{\mathbf{z}_i^t} \left(\left\| g_i^t \right\| - \mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\| \right)^2 = \mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\|^2 - \left(\mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\| \right)^2. \tag{98}$$

Now, as Euclidean norm $\|\cdot\|$ is a convex function [8], using Jensen's inequality we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\| \geq \left\| \mathbb{E}_{\mathbf{z}_i^t} \left(g_i^t \right) \right\|, \quad \forall i \in \mathcal{H}. \tag{99}$$

As all agents in $\mathcal{H}$ are non-faulty, recall from (15) in Section 2.3 that

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

Upon substituting from above in (99) we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\| \geq \left\| \nabla Q(w^t) \right\|, \quad \forall i \in \mathcal{H}. \tag{100}$$

Substituting from (100) in (98) we obtain that

$$\mathbb{E}_{\mathbf{z}_i^t} \left(\left\| g_i^t \right\| - \mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\| \right)^2 \leq \mathbb{E}_{\mathbf{z}_i^t} \left\| g_i^t \right\|^2 - \left\| \nabla Q(w^t) \right\|^2, \quad \forall i \in \mathcal{H}. \tag{101}$$

Substituting from (95) in (101) we obtain that

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

The above proves (97), i.e,

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

Finally, substituting from (93) and (97) in (92) we obtain that

$$\begin{aligned} \mathbb{E}_{\zeta^t} \left\| g_{\nu_t}^t \right\| &\leq \sigma + \left\| \nabla Q(w^t) \right\| + \sqrt{\sigma^2 (n - f - 1)} \\ &= \sigma \left(1 + \sqrt{n - f - 1} \right) + \left\| \nabla Q(w^t) \right\|. \end{aligned}$$

As t above is an arbitrary iteration, the above proves the lemma.

C Proof of Corollary 1

From Markov's inequality [29], for an arbitrary positive real value ϵ,

$$Pr\left(\left\|w^{t+1}-w^*\right\|^2 \geq \epsilon\right) \leq \frac{\mathbb{E}_t\left\|w^{t+1}-w^*\right\|^2}{\epsilon}, \quad \forall t \geq 0. \tag{102}$$

If the conditions in Theorem 1 hold true then for all $t \geq 0$,

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

where $\rho \in (0,\, 1)$. Substituting from above in (102) we obtain, for all $t \geq 0$, that

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

Equivalently,

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

As $\rho < 1$, $\lim_{t\to\infty}\rho^t = 0$. Therefore, limiting t to infinity on both sides of the above inequality proves the corollary, i.e.,

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

□