FED−CURE: A Robust Federated Learning Algorithm with Cubic Regularized Newton

Avishek Ghosh [1] **Raj Kumar Maity** [2] **Arya Mazumdar** [3]

Abstract

In this paper, we analyze the cubic-regularized Newton method that avoids saddle points in non-convex optimization in the Federated Learning (FL) framework and simultaneously address several practical challenges that naturally arise in FL, like communication bottleneck and Byzantine attacks. We propose FEDerated CUbic REgularized Newton (FED−CURE) and obtain convergence guarantees under several settings. Being a second order algorithm, the iteration complexity of FED−CURE is much lower than its first order counterparts, and furthermore we can use compression (or sparsification) techniques like δ-approximate compression to achieve communication efficiency and norm-based thresholding for Byzantine resilience. We validate the performance of FED−CURE with experiments using standard datasets and several types of Byzantine attacks, and obtain an improvement of 25% with respect to first order methods in total iteration complexity.

1. Introduction

In FL, it is well-known that one of the major challenges is to tackle the behavior of the Byzantine machines (Lamport et al., 1982), which behave completely arbitrarily. This can happen owing to software or hardware crashes, poor communication link between the local machines and the center machine, stalled computations, and even coordinated or malicious attacks by a third party (see (Yin et al., 2018; Blanchard et al., 2017)). Another critical challenge in FL is the communication cost between the local machines and the center machine. The gains we obtain by parellelization of the task among several local machines often get bottle-necked by the this cost.

[1]Systems and Control Engg. and the Centre for Machine Intelligence and Data Sciences (CMInDS) at the Indian Institute of Technology, Bombay. [2]CS Department, University of Massachusetts, Amherst. [3]Halicioglu Data Science Institute, University of California, San Diego. Correspondence to: Avishek Ghosh <avishek_ghosh@iitb.ac.in>.

Workshop of Federated Learning and Analytics in Practice, colocated with 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA.

In order to fit complex machine learning models, one often requires to find local minima of a non-convex loss $f(.)$, instead of just critical points which may include several saddle points. Training deep neural networks and other high-capacity learning architectures (Soudry & Carmon, 2016; Ge et al., 2017) are some of the examples where finding local minima is crucial. (Ge et al., 2017; Kenji, 2016) show that the stationary points of these problems are in fact saddle points and far away from any local minimum, and hence designing efficient algorithm that escapes saddle points is of interest. Moreover, (Jain et al., 2017; Sun et al., 2016) argue that saddle points can lead to highly sub-optimal solutions in many problems of interest. This is amplified in high dimension as shown in (Dauphin et al., 2014), and becomes the main bottleneck in training deep neural nets. Furthermore, a line of recent work (Sun et al., 2016; Bhojanapalli et al., 2016; Sun et al., 2017), show that for many non-convex problems, it is sufficient to find a local minimum. In fact, in many problems of interest, all local minima are global minima (e.g., dictionary learning (Sun et al., 2017), phase retrieval (Sun et al., 2016), matrix sensing and completion (Bhojanapalli et al., 2016; Ge et al., 2017), and some of neural nets (Kenji, 2016)). Also, in (Choromanska et al., 2015), it is argued that for more general neural nets, the local minima are as good as global minima.

The issue of local minima convergence becomes non-trivial in the presence of Byzantine local machines as they can create *fake local minima* that are close to the saddle point of the loss function $f(.)$, and these are far away from the true local minima. This is popularly known as the *saddle-point attack* (see (Yin et al., 2019)), and it can arbitrarily destroy the performance of any non-robust learning algorithm.

The problem of saddle point avoidance in the context of non-convex optimization has received considerable attention in the past few years. In the seminal paper of (Jin et al., 2017), a (first order) gradient descent based approach is proposed. A few papers (Xu et al., 2017; Allen-Zhu & Li, 2017) following the above use various modifications. A Byzantine robust first order saddle point avoidance algorithm is proposed in (Yin et al., 2019), and probably is the closest to this work. Being a first order algorithm, the convergence rate is quite slow (the rate for gradient decay is $1/\sqrt{T}$, where T is the number of iterations). Moreover, implementation-wise, the algorithm presented in (Yin et al., 2019) is computation heavy, and takes potentially many iterations between the center and

Algorithm 1 FED−CURE

Input: Step size η_k, parameter $0 \le \alpha \le \beta, \gamma > 0, M > 0$ and δ-approximate compressor Q.
Initialize: Initial iterate $\mathbf{x}_0 \in \mathsf{R}^d$
for $k = 0,1,...,T-1$ **do**
  Central machine: broadcasts $\mathbf{x}_k$
  for $i \in [m]$ **do in parallel**
  i-th local machine:
  Non-Byzantine: Compute local gradient $\mathbf{g}_{i,k}$ and Hessian $\mathbf{H}_{i,k}$; locally solve the problem equation (1). Use the compressor Q and send $Q(\mathbf{s}_{i,k+1})$ to the center,
  Byzantine: Generate $\star$ (arbitrary), and send to center
  end for
  Center Machine:
  (i) Sort the local machines in a non decreasing order according to norm of updates $\{Q(\mathbf{s}_{i,k+1})\}_{i=1}^m$
  (ii) Return first $1-\beta$ fraction indices, $\mathcal{U}_k$,
  (iii) Update: $\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \frac{1}{|\mathcal{U}_k|} \sum_{i \in \mathcal{U}_k} Q(\mathbf{s}_{i,k+1})$
end for

the local machines (as we check in Section 4). Hence, this algorithm is not efficient in terms of the communication cost.

In this work, we consider a variant of the famous cubic-regularized Newton algorithm of Nesterov and Polyak (Nesterov & Polyak, 2006), namely FEDerated CUbic REgularized newton (FED−CURE), which efficiently escapes the saddle points of a non-convex function by appropriately choosing a regularization and thus pushing the Hessian towards a positive semi-definite matrix. The primary motivation behind this choice is the faster convergence rate compared to first order methods, which is crucial in terms of communication efficiency in applications like FL. Indeed, the rate of gradient decay is $\frac{1}{T^{2/3}}$.

FED−CURE simultaneously uses (i) a δ-approximate compressor (defined shortly) to compress the message send from local machines to center for communication reduction and (ii) a simple norm-based thresholding on the (compressed) solution sent by the local machines to defend adversarial (Byzantine) attacks. Norm based thresholding is also a standard trick for Byzantine resilience as featured in (Ghosh et al., 2021; 2020a). However, since the local optimization problem lacks a closed form solution, using norm-based trimming is also technical challenging in this case.

1.1. Our Contributions

Technical Novelty We propose FED−CURE that escapes saddle point efficiently and converges at a rate of $\frac{1}{T^{2/3}}$, which is faster than the first order methods (which converge at $1/\sqrt{T}$ rate, see (Yin et al., 2019)). Also, the convergence rate matches to that of the centralized scheme of (Nesterov & Polyak, 2006) and hence, we do not lose in terms of convergence rate while making the algorithm distributed.

In FED−CURE, the center machine aggregates the solution of the local machines. We emphasize that, unlike gradient aggregation, the aggregation of the solutions of the local optimization problems is a highly non-linear operation, as evidenced by even a much simpler second order optimization algorithm like GIANT ((Wang et al., 2019)). Hence, it is quite non-trivial to extend the centralized cubic regularized algorithm to a distributed one. The solution to the cubic regularization even lacks a closed form solution unlike the second order Hessian based update or the first order gradient based update. The analysis of FED−CURE is carried out by leveraging the first order and second order stationary conditions of the auxiliary function solved in each local machines.

Along with the saddle point avoidance, we simultaneously address the issues of (i) communication efficiency and (ii) Byzantine resilience by using a δ-approximate compressor and a norm-based thresholding scheme respectively. A major technical challenge here is to simultaneously address the above mentioned issues jointly.

Experiments In Section 4 (and in Appendix D), we verify our theoretical findings via experiments. We first show that FED−CURE indeed avoids saddle points via a simple example. Moreover, we use benchmark LIBSVM ((Chang & Lin, 2011)) datasets for logistic regression and non-convex robust regression and show convergence results for both non-Byzantine and several different Byzantine attacks. We observe that the algorithm of (Yin et al., 2019) requires 25% more total iterations than ours.

1.2. Problem Formulation

We minimize a loss function of the form: $f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m f_i(\mathbf{x})$, where the function $f : \mathsf{R}^d \to \mathsf{R}$ is twice differentiable and non-convex. We consider a standard FL framework with m local machines and one center machine where the local machines can only communicate to the center machine. Each local machine is associated with a local loss function f_i. We assume that the data distribution is non-iid across local machines, which is standard in FL. In addition to this, we also consider the case where α fraction of the local machines are Byzantine for some $\alpha < \frac{1}{2}$. The Byzantine machines can send arbitrary updates to the central machine which can disrupt the learning. Furthermore, the Byzantine machines can collude with each other, create *fake local minima* or attack maliciously by gaining information about the learning algorithm and other local machines. Furthermore, as stated in Section 1, we use a generic class of compressors from (Karimireddy et al., 2019):

Definition 1.1 (δ-Approximate Compressor)**.** An operator $Q(.) : \mathsf{R}^d \to \mathsf{R}^d$ is defined as δ approximate compressor on a set $\mathcal{S} \subseteq \mathsf{R}^d$ if, $\forall x \in \mathcal{S}$, $\|Q(x) - x\|^2 \le (1-\delta)\|x\|^2$, where $\delta \in (0,1]$ is the compression factor.

2. Algorithm: FED−CURE

In this section, we describe FED−CURE. Starting with initialization $\mathbf{x}_0$, the center machine broadcasts the parameter to

the local machines. At k-th iteration, the i-th local machine solves a cubic-regularized auxiliary loss function based on its local data:

$$\mathbf{s}_{i,k+1} = \operatorname*{argmin}_{\mathbf{s}} \mathbf{g}_{i,k}^T \mathbf{s} + \frac{\gamma}{2} \mathbf{s}^T \mathbf{H}_{i,k} \mathbf{s} + \frac{M}{6} \gamma^2 \|\mathbf{s}\|^3, \quad (1)$$

where $M > 0, \gamma > 0$ are parameter choose suitably and $\mathbf{g}_{i,k}, \mathbf{H}_{i,k}$ are the gradient and Hessian of the local loss function f_i computed on data (S_i) stored in the local machine given by $\mathbf{g}_{i,k} = \nabla f_i(x_k) = \frac{1}{|S_i|} \sum_{z_i \in S_i} \nabla f_i(x_k, z_i)$ and $\mathbf{H}_{i,k} = \nabla^2 f_i(x_k) = \frac{1}{|S_i|} \sum_{z_i \in S_i} \nabla^2 f_i(x_k, z_i)$. After solving the problem described in (1), each local machine applies compression operator Q as defined in Definition 1.1 on update $\mathbf{s}_{i,k+1}$. The application of the compression on the update is to minimize the communication cost.

Moreover, we also consider that $\alpha (< \frac{1}{2})$ fraction of the local machines are Byzantine in nature. We denote the set of Byzantine local machines by $\mathcal{B}$ and the set of the rest of the good machines as $\mathcal{M}$. In each iteration, the good machines send the compressed update of solution of the sub-cubic problem described in equation (1) and the Byzantine machines can send any arbitrary values or intentionally disrupt the learning algorithm with malicious updates.

After receiving all the updates from the local machines, the central machine outputs a set $\mathcal{U}$ which consists of the indices of the local machines with smallest norm. FED-CURE chooses the size of the set $\mathcal{U}$ to be $(1-\beta)m$. Hence, we 'trim' β fraction of the local machine so that we can control the iterated update by not letting the local machines with large norm participate and diverge the learning process. We denote the set of trimmed machine as $\mathcal{T}$. We choose $\beta > \alpha$ so that at least one of the good machines gets trimmed for theoretical tractability. The central machine updates the parameter, with step-size η_k as $\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \frac{1}{|U_t|} \sum_{i \in U_t} Q(\mathbf{s}_{i,k+1})$.

Remark 2.1 (Exact solution only for theory). We emphasize that the exact solution of the sub-problem is only required for theoretical tractability. In practice, it is not possible to obtain such solution. For that reason, in experiments (Section 4) we run the gradient based first order algorithm of (Tripuraneni et al., 2018) to achieve this. We expand on this in Section 3.1.

Remark 2.2. Note that, we introduce the parameter γ in the cubic regularized sub-problem, which was absent in the original formulation of (Nesterov & Polyak, 2006). γ emphasizes the effect of the second and third order terms in the sub-problem, and is important for convergence of FED-CURE.

3. Theoretical Guarantees

Assumption 3.1. $f(.)$ is twice continuously-differentiable and bounded below, i.e., $f^* = \inf_{\mathbf{x} \in \mathbb{R}^d} f(x) > -\infty$.

Assumption 3.2. The loss $f(.)$ is *L-Lipschitz continuous* $(\forall \mathbf{x}, \mathbf{y},\ |f(\mathbf{x}) - f(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|)$, has *$L_1$-Lipschitz gradients* $(\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L_1 \|\mathbf{x} - \mathbf{y}\|)$ and *L_2-Lipschitz Hessian* $(\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \leq L_2 \|\mathbf{x} - \mathbf{y}\|)$.

The above assumption states that the loss and the gradient and Hessian of the loss do not drastically change in the local neighborhood. These assumptions are standard in the analysis of the saddle point escape for cubic regularization (see (Tripuraneni et al., 2018; Kohler & Lucchi, 2017; Nesterov & Polyak, 2006; Carmon & Duchi, 2016)).

We assume the data distribution across local machines to be non-iid. However, we assume that the local gradient and Hessian computed at local machines (using local data) satisfies the following gradient and Hessian dissimilarity conditions. Note that these conditions are only applicable for non-Byzantine machines only. Byzantine machines can generate arbitrary gradients and Hessian.

Definition 3.3 (Bounded Heterogeneity)**.** For $\epsilon_g > 0$ and $\epsilon_H > 0$, we have $\|\nabla f(\mathbf{x}_k) - \mathbf{g}_{i,k}\| \leq \epsilon_g$ and $\|\nabla^2 f(\mathbf{x}_k) - \mathbf{H}_{i,k}\| \leq \epsilon_H$ for all k and i.

We emphasize that bounded gradient and hessian dissimilarity are quite common in FL, and are one major way to characterize the *degree of heterogeneity*. For example, see (Zhao et al., 2018; Sahu et al., 2018; Li et al., 2018; Sattler et al., 2019; Mohri et al., 2019; Karimireddy et al., 2020; Fallah et al., 2020) and the references therein. These papers use this *bounded heterogeneity* condition to motivate the need of obtaining a single model for all local machines.

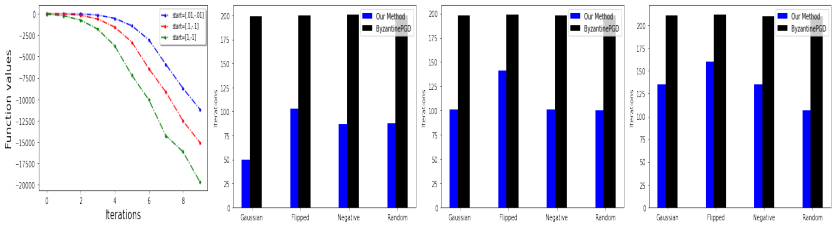
Values of ϵ_g and ϵ_H in special cases In (Kohler & Lucchi, 2017; Tripuraneni et al., 2018; Wang et al., 2020), the authors consider gradient and Hessian with sub-sampled data being drawn uniformly randomly from the data set leading to (ϵ_g, ϵ_H) diminishing at the rate $\propto 1/\sqrt{|S|}$ where $|S|$ is the size of the data sample in each local machine.

Remark 3.4 (Two rounds of communication $\epsilon_g = 0$, $\epsilon_H = 0$). We can make $\epsilon_g = 0$ via one more round of communication in each iteration. In the first iteration, all the local machines compute the local gradient, sends back and the center machine broadcasts the global gradient $\nabla f(\mathbf{x}_k) = \frac{1}{m} \sum_{i=1}^m \mathbf{g}_{i,k}$. In this manner, the local machines solve the sub-problem (1) with the actual gradient. Note that (Wang et al., 2019) does this exactly to avoid ϵ_g. Similarly, with more communication cost, we can make $\epsilon_H = 0$ by allowing local machines to send local Hessians and the center to aggregate and broadcast the aggregated Hessian. However, in standard FL, one typically avoids this additional round of communication and deal with gradient and Hessian dissimilarities.

Theorem 3.5 (Convergence of FED-CURE)**.** *Suppose $0 \leq \alpha < \beta \leq \frac{1}{2}$ and $m \geq 2$. Furthermore, we choose the problem parameters, $M = \mathcal{O}(m(1-\beta)(1+\sqrt{1-\delta})^3)$, and $\eta_k = \frac{c}{T m^\nu}; \gamma = \frac{c}{T m^\nu}$, for some constant $c > 0, \nu > 3$. Then, after T iterations of FED-CURE (Algorithm 1), the sequence $\{\mathbf{x}_i\}_{i=1}^T$ generated contains a point $\tilde{x}$ such that*

$$\|\nabla f(\tilde{x})\| \leq \frac{\chi_1}{T^{2/3}} + \epsilon_g + \chi_G,\ \lambda_{\min}(\nabla^2 f(\tilde{x})) \geq \frac{\chi_2}{T^{\frac{1}{3}}} - \epsilon_H - \chi_H,$$

where $(\chi_1, .., \chi_H)$ are T independent and depend on m, δ, β. The complete expressions can be found at Appendix B.



(a) (b) (c) (d)

Figure 1. (a) Plot of the function value with different initialization to show that the algorithm escapes the saddle point with functional value 0. (b,c,d) Comparison of our algorithm with ByzantinePGD (Yin et al., 2019) in terms of the total number of iterations.

Corollary 3.6 (Matching (Nesterov & Polyak, 2006))**.** *Suppose* $\alpha = 0, \beta = 0, \delta = 1$, $m = 1$ *(centralized),* $\eta = \gamma = 1$. *With this, we get* $\|\nabla f(\tilde{x})\| \leq \mathcal{O}(\frac{1}{T^{2/3}})$ *and* $\lambda_{\min}\left(\nabla^2 f(\tilde{x})\right) \geq -\mathcal{O}(\frac{1}{T^{1/3}})$*, which matches (Nesterov & Polyak, 2006).*

Discussion In Theorem 3.5, we have three terms. The first term implies rate of decay for gradient and the minimum eigenvalue of the Hessian are $O(1/T^{\frac{2}{3}})$ and $O(1/T^{\frac{1}{3}})$, respectively. We point out that both of these rates match with that of the centralized version of the cubic regularized Newton as shown in (Nesterov & Polyak, 2006).

The second term depends on ϵ_g and ϵ_H. This is owing to the *non-iid* nature of data. Note that in the centralized setup of (Nesterov & Polyak, 2006), this aspect of heterogeneity was absent. Furthermore, as mentioned above, in the special cases, both the terms ϵ_g and ϵ_H decrease at the rate of $1/\sqrt{|S|}$, where $|S|$ is the number of data in each of the local machines.

The third term is an error floor that decays with the number of machines, m, and can be made arbitrarily small. As shown in Corollary 3.6, this term vanishes when $m = 1$. This term originates from several sources. First, the center machine simply aggregates the solution of the local machines to obtain the next update. Unlike gradient aggregation, this yields a different solution from the global one, and hence one incurs a bias. This is the cost of going from centralized to a distributed setup. Second, our norm based thresholding also creates an error floor.

Comparison with (Yin et al., 2019) In a recent work, (Yin et al., 2019) provides a *perturbed gradient based algorithm* to escape the saddle point in non-convex optimization in the presence of Byzantine local machines. Also, in that paper, the Byzantine resilience is achieved using techniques such as trimmed mean, median and collaborative filtering. These methods require additional assumptions (coordinate of the gradient being sub-exponential etc.). In this work, we do not require such assumptions. Also (Yin et al., 2019) requires several rounds of communications between the central machine and the local machines whenever the norm of the gradient is small as this is an indication of either a local minima or a saddle point. In contrast to that, our method does not require any additional communication. Our method provides such ability by virtue of cubic regularization. Our algorithm achieves a superior rate of $O(1/T^{\frac{2}{3}})$ compared to the gradient based approach of rate $O(1/\sqrt{T})$. Our algorithm dominates ByzantinePGD (Yin et al., 2019) in terms of convergence, communication rounds and simplicity.

3.1. Solution of the cubic sub-problem

The cubic regularized sub-problem (1) needs to be solved to update the parameter. As this particular problem does not have a closed form solution, a solver is usually employed which yields a satisfactory solution (Cartis et al., 2011a; Agarwal et al., 2017; Carmon & Duchi, 2016). For the purpose of theoretical convergence analysis, similar to previous works (Wang et al., 2020; Zhou et al., 2018; Wang et al., 2019), we consider that local machines obtain the exact solution in each round. However, in experiments (Section 4), we apply the gradient based solver of (Tripuraneni et al., 2018) to solve the sub-problem. In Appendix C, we discuss this in detail.

4. Experimental results

First we show that `FED-CURE` *escapes saddle point* with a toy example with $(d = 2)$ $\min_{w \in \mathbb{R}^2}[f_1(w) + f_2(w)]$ where $f_1(w) = w_1^2 - w_2^2$ and $f_2(w) = 2w_1^2 - 2w_2^2$ (Here w_i denotes the i-th coordinate of w. In Figure 1 (a) we observe that our algorithm escapes the saddle point $(0,0)$, with random initialization. Note that, checking whether a point is a local minima or a saddle point is an NP-hard problem for non-convex losses (see (Jin et al., 2021), Sec. 2.2). So, for a simple toy problem, we may brute-force our way through to show *saddle points escape*, but this becomes intractable for real data examples. Also, in Figure 1 (b)-(d), we compare the total iteration complexity with PGD of (Yin et al., 2019).

More experimental details can be found in Appendix D. Here we provide a brief overview. We demonstrate the convergence of our algorithm, `FED-CURE` with benchmark dataset LIBSVM ((Chang & Lin, 2011)) with and without compression for both convex and non-convex losses. We choose four different Byzantine attacks, and show resilience against them. We compare our results with ByzantinePGD (Yin et al., 2019), a standard benchmark for robust saddle point avoidance algorithms. Furthermore, we compare `FED-CURE` with standard Federated Learning algorithms such as FEDGLOMO(Das et al., 2020) and FedAvg (McMahan et al., 2017).

References

Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., and Ma, T. Finding approximate local minima faster than gradient descent. In *49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1195–1199, 2017.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *NeurIPS*, pp. 1709–1720, 2017.

Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., and Renggli, C. The convergence of sparsified gradient methods. In *NeurIPS*, pp. 5973–5983, 2018.

Allen-Zhu, Z. and Li, Y. Neon2: Finding local minima via first-order oracles. *arXiv:1711.06673*, 2017.

Avdiukhin, D. and Yaroslavtsev, G. Escaping saddle points with compressed sgd. *In NeurIPS*, 2021.

Bernstein, J., Zhao, J., Azizzadenesheli, K., and Anandkumar, A. signsgd with majority vote is communication efficient and byzantine fault tolerant. *arXiv:1810.05291*, 2018.

Bhojanapalli, S., Neyshabur, B., and Srebro, N. Global optimality of local search for low rank matrix recovery, 2016.

Blanchard, P., Mhamdi, E. M. E., Guerraoui, R., and Stainer, J. Byzantine-tolerant machine learning. *arXiv:1703.02757*, 2017.

Carmon, Y. and Duchi, J. C. Gradient descent efficiently finds the cubic-regularized non-convex newton step. *arXiv:1612.00547*, 2016.

Cartis, C., Gould, N. I., and Toint, P. L. Adaptive cubic regularisation methods for unconstrained optimization. part i: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011a.

Cartis, C., Gould, N. I., and Toint, P. L. Adaptive cubic regularisation methods for unconstrained optimization. part ii: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011b.

Chang, C.-C. and Lin, C.-J. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

Chen, Y., Su, L., and Xu, J. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *ACM on Measurement and Analysis of Computing Systems*, 1(2):44, 2017.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks, 2015.

Das, R., Acharya, A., Hashemi, A., Sanghavi, S., Dhillon, I. S., and Topcu, U. Faster non-convex federated learning via global and local momentum. *arXiv preprint arXiv:2012.04061*, 2020.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *NeurIPS*, volume 27, pp. 2933–2941, 2014.

Du, S. S., Jin, C., Lee, J. D., Jordan, M. I., Poczos, B., and Singh, A. Gradient descent can take exponential time to escape saddle points. *arXiv:1705.10412*, 2017.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning: A meta-learning approach. *arXiv:2002.07948*, 2020.

Feng, J., Xu, H., and Mannor, S. Distributed robust learning. *arXiv:1409.5937*, 2014.

Gandikota, V., Kane, D., Maity, R. K., and Mazumdar, A. vqsgd: Vector quantized stochastic gradient descent. In *AISTATS*, pp. 2197–2205. PMLR, 2021.

Ge, R., Jin, C., and Zheng, Y. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *34th ICML*, volume 70, pp. 1233–1242. PMLR, 2017.

Ghosh, A., Maity, R. K., and Mazumdar, A. Distributed newton can communicate less and resist byzantine workers. In *NeurIPS December 6-12, 2020, virtual*, 2020a.

Ghosh, A., Maity, R. K., Mazumdar, A., and Ramchandran, K. Communication efficient distributed approximate newton method. In *ISIT*, pp. 2539–2544. IEEE, 2020b.

Ghosh, A., Maity, R. K., Kadhe, S., Mazumdar, A., and Ramchandran, K. Communication-efficient and byzantine-robust distributed learning with error feedback. *IEEE Journal on Selected Areas in Information Theory*, 2(3):942–953, 2021.

Jain, P., Jin, C., Kakade, S. M., and Netrapalli, P. Global convergence of non-convex gradient descent for computing matrix squareroot, 2017.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In *ICML*, pp. 1724–1732. PMLR, 2017.

Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29, 2021.

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error feedback fixes signsgd and other gradient compression schemes. In *ICML*, pp. 3252–3261. PMLR, 2019.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, pp. 5132–5143. PMLR, 2020.

Kenji, K. Deep learning without poor local minima. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *NeurIPS*, volume 29, pp. 586–594. Curran Associates, Inc., 2016.

Kohler, J. M. and Lucchi, A. Sub-sampled cubic regularization for non-convex optimization. In *ICML*, pp. 1895–1904. PMLR, 2017.

Lamport, L., Shostak, R., and Pease, M. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4 (3):382–401, July 1982. ISSN 0164-0925.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient descent converges to minimizers. *arXiv:1602.04915*, 2016.

Lee, J. D., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M. I., and Recht, B. First-order methods almost always avoid saddle points. *arXiv:1710.07406*, 2017.

Li, L., Xu, W., Chen, T., Giannakis, G. B., and Ling, Q. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. *arXiv:1811.03761*, 2018.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *20th AISTATS*, pp. 1273–1282, 2017.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *ICML*, pp. 4615–4625. PMLR, 2019.

Nesterov, Y. and Polyak, B. T. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Sahu, A. K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., and Smith, V. On the convergence of federated optimization in heterogeneous networks. *arXiv:1812.06127*, 3, 2018.

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2019.

Soudry, D. and Carmon, Y. No bad local minima: Data independent training error guarantees for multilayer neural networks, 2016.

Sun, J., Qu, Q., and Wright, J. A geometric analysis of phase retrieval. *CoRR*, abs/1602.06664, 2016.

Sun, J., Qu, Q., and Wright, J. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63 (2):853–884, Feb 2017.

Tripuraneni, N., Stern, M., Jin, C., Regier, J., and Jordan, M. Stochastic cubic regularization for fast nonconvex optimization. In *NeurIPS*, pp. 2899–2908, 2018.

Wang, H., Sievert, S., Liu, S., Charles, Z., Papailiopoulos, D., and Wright, S. Atomo: Communication-efficient learning via atomic sparsification. In *NeurIPS*, pp. 9850–9861, 2018.

Wang, Z., Zhou, Y., Liang, Y., and Lan, G. Stochastic variance-reduced cubic regularization for nonconvex optimization. In *The 22nd AISTATS*, pp. 2731–2740. PMLR, 2019.

Wang, Z., Zhou, Y., Liang, Y., and Lan, G. Cubic regularization with momentum for nonconvex optimization. In *Uncertainty in Artificial Intelligence*, pp. 313–322. PMLR, 2020.

Xu, Y., Jin, R., and Yang, T. First-order stochastic algorithms for escaping from saddle points in almost linear time. *arXiv:1711.01944*, 2017.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *35th ICML*, 2018.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Defending against saddle point attack in byzantine-robust distributed learning. In *ICML*, 2019.

Zhang, C. and Li, T. Escape saddle points by a simple gradient-descent based algorithm. *In NeurIPS*, 2021.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv:1806.00582*, 2018.

Zhou, D., Xu, P., and Gu, Q. Stochastic variance-reduced cubic regularized newton methods. In *ICML*, pp. 5990–5999. PMLR, 2018.

Supplementary Material

A. Detailed Related Work

Saddle Point avoidance algorithms: In the recent years, there are handful first order algorithms (Lee et al., 2016; 2017; Du et al., 2017) that focus on the escaping saddle points and convergence to local minima. The critical algorithmic aspect is running gradient based algorithm and adding perturbation to the iterates when the gradient is small. ByzantinePGD (Yin et al., 2019), PGD (Jin et al., 2017), Neon+GD(Xu et al., 2017), Neon2+GD (Allen-Zhu & Li, 2017) are examples of such algorithms. The work of Nesterov and Polyak (Nesterov & Polyak, 2006) first proposes the cubic regularized second order Newton method and provides analysis for the second order stationary condition. An algorithm called Adaptive Regularization with Cubics (ARC) was developed by (Cartis et al., 2011a;b) where cubic regularized Newton method with access to inexact Hessian was studied. Cubic regularization with both the gradient and Hessian being inexact was studied in (Tripuraneni et al., 2018). In (Kohler & Lucchi, 2017), a cubic regularized Newton with sub-sampled Hessian and gradient was proposed and analyzed. Momentum based cubic regularized algorithm was studied in (Wang et al., 2020). A variance reduced cubic regularized algorithm was proposed in (Zhou et al., 2018; Wang et al., 2019). In terms of solving the cubic sub-problem, (Carmon & Duchi, 2016) proposes a gradient based algorithm and (Agarwal et al., 2017) provides a Hessian-vector product technique. (Zhang & Li, 2021) employs a *a negative curvature finding algorithm* based on gradient descent and accelerated gradient descent method to improve the PGD algorithm (Jin et al., 2017). (Avdiukhin & Yaroslavtsev, 2021) proposes perturbed compressed SGD with error feedback.

Compression: In the recent years, several gradient quantization or sparsification schemes have been studied in (Gandikota et al., 2021; Alistarh et al., 2018; Wang et al., 2018; Alistarh et al., 2017). In (Karimireddy et al., 2019), the authors introduced the idea of δ-approximate compressor. In (Ghosh et al., 2020b), the authors use δ-approximate compressor to sparsify the second order update.

Byzantine resilience: In the distributed learning context, (Feng et al., 2014) proposes one shot median based robust learning. A median of mean based algorithm was proposed in (Chen et al., 2017) where the worker machines are grouped in batches and the Byzantine resilience is achieved by computing the median of the grouped machines. Later (Yin et al., 2018) proposes co-ordinate wise median, trimmed mean and iterative filtering based approaches. Communication-efficient and Byzantine robust algorithms were developed in (Bernstein et al., 2018; Ghosh et al., 2021). A norm based thresholding approach for Byzantine resilience for distributed Newton algorithm was also developed (Ghosh et al., 2020a). All these works provide only first order convergence guarantee (small gradient). The work (Yin et al., 2019) is the only one that provides second order guarantee (Hessian positive semi-definite) under Byzantine attack.

B. Theorem 3.5 with special cases

We here state the convergence guarantee of FED−CURE formally.

Theorem B.1 (Convergence of FED−CURE)**.** *Suppose* $0 \leq \alpha < \beta \leq \frac{1}{2}$ *and* $m \geq 2$*. Furthermore, we choose the problem parameters,* $M = \mathcal{O}(m(1-\beta)(1+\sqrt{1-\delta})^3)$*, and* $\eta_k = \frac{c}{Tm^\nu}; \gamma = \frac{c}{Tm^\nu}$*, for some constant* $c > 0, \nu > 3$*. Then, after* T *iterations of* FED-CURE *(Algorithm 1), the sequence* $\{\mathbf{x}_i\}_{i=1}^T$ *generated contains a point* $\tilde{x}$ *such that*

$$\| \nabla f(\tilde{x}) \| \leq \frac{\chi_1}{T^{2/3}} + \epsilon_g + \chi_G, \quad \lambda_{\min}\left(\nabla^2 f(\tilde{x}) \right) \geq -\frac{\chi_2}{T^{\frac{1}{3}}} - \epsilon_H - \chi_H,$$

where,

$$\chi_1 = O\left(\frac{(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)} + m(1-\beta)(1+\sqrt{1-\delta})^3\right)$$
$$\chi_2 = O\left(\frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)} + m(1-\beta)(1+\sqrt{1-\delta})^3\right)$$
$$\chi_G = O\Bigg(\left[\frac{(1-\alpha)(1+\sqrt{1-\delta})^2}{(1-\beta)} + m(1-\beta)(1+\sqrt{1-\delta})^3\right]\left(\frac{1}{m}\right)^{\left(\frac{2\nu}{3}\right)} + \frac{\alpha(1+\sqrt{1-\delta})^2}{(1-\beta)m^{2\nu}} + \frac{\alpha}{(1-\beta)m^{\nu}}(1+\sqrt{1-\delta})\Bigg)$$
$$\chi_H = O\Bigg(\left[m(1-\beta)(1+\sqrt{1-\delta})^3 + \frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)}\right]\left(\frac{1}{m}\right)^{\frac{\nu}{3}} + \frac{(1+\sqrt{1-\delta})\alpha}{(1-\beta)m^{\nu}}\Bigg)$$

Here, we state two corollaries. First, we relax the compression by choosing $\delta = 1$ and then the Byzantine resilience.

Corollary B.2 (No compression)**.** *Suppose $0 \le \alpha \le \beta \le \frac{1}{2}$, and we choose $M = \mathcal{O}(m(1-\beta))$, $\eta = \gamma = c/m^{\nu}T$ for some $c > 0, \nu > 3$. Then, after T iterations of* FED-CURE *for uncompressed update ($\delta = 1$), the sequence $\{\mathbf{x}_i\}_{i=1}^{T}$ generated contains a point $\tilde{x}$ such that*

$$\|\nabla f(\tilde{x})\| \le \frac{\chi_1}{T^{2/3}} + \epsilon_g + \chi_G, \lambda_{\min}\left(\nabla^2 f(\tilde{x})\right) \ge -\frac{\chi_2}{T^{\frac{1}{3}}} - \epsilon_H - \chi_H,$$

where,

$$\chi_1 = \mathcal{O}([\frac{(1-\alpha)}{(1-\beta)} + m(1-\beta)]), \quad \chi_2 = \mathcal{O}([\frac{(1-\alpha)}{(1-\beta)} + m(1-\beta)])$$
$$\chi_G = \mathcal{O}(\left[\frac{(1-\alpha)}{(1-\beta)} + m(1-\beta)\right](\frac{1}{m})^{(\frac{2\nu}{3})} + \frac{\alpha}{(1-\beta)m^{2\nu}} + \frac{\alpha}{(1-\beta)m^{\nu}})$$
$$\chi_H = \mathcal{O}(\left(m(1-\beta) + \frac{(1-\alpha)}{(1-\beta)}\right)(\frac{1}{m})^{\nu/3} + \frac{\alpha}{(1-\beta)m^{\nu}}).$$

Note that we still obtain similar three terms, but the expressions of $\chi_1, \chi_2 . \chi_G, \chi_H$ have reduced, which improves the convergence guarantees. The observations we made in Theorem 3.5 continues to hold here.

Next, we choose the non-Byzantine setup with $\alpha = \beta = 0$ in addition to the uncompressed update. This is just the distributed variant of the cubic regularized Newton method of (Nesterov & Polyak, 2006).

Corollary B.3 (Non Byzantine and no compression)**.** *Suppose we choose $M = \mathcal{O}(m)$, $\eta = \gamma = c/Tm^{\nu}$ for some $c > 0, \nu > 3$. Then, after T iterations of* FED-CURE *for uncompressed update ($\delta = 1$), the sequence $\{\mathbf{x}_i\}_{i=1}^{T}$ generated contains a point $\tilde{x}$ such that*

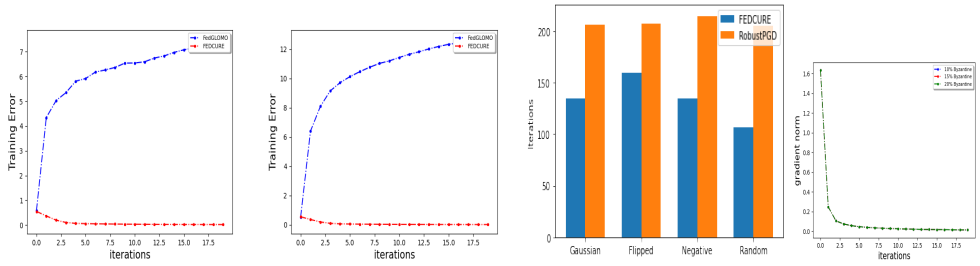
$$\|\nabla f(\tilde{x})\| \le \frac{\chi_1}{T^{2/3}} + \epsilon_g + \chi_G, \lambda_{\min}\left(\nabla^2 f(\tilde{x})\right) \ge -\frac{\chi_2}{T^{\frac{1}{3}}} - \epsilon_H - \chi_H,$$

where, $\chi_1 = \chi_2 = \mathcal{O}(m)$ and $\chi_G = \mathcal{O}(\frac{1}{m^{\frac{2\nu}{3}-1}}), \chi_H = \mathcal{O}(\frac{1}{m^{\frac{\nu}{3}-1}})$.

Note that the term χ_1, χ_2, χ_G and χ_H have further reduced, thus improving the performance. As $\nu > 3$, the parameter χ_G, χ_H are decreasing with the number of worker machines. Note that even in the simple distributed variant, the extra error terms (second and third terms) are present. As explained earlier, these are owing to the non-iid nature of data distribution and the simple (biased) aggregation of local solutions at the center respectively.

C. Solution of the cubic sub-problem

The cubic regularized sub-problem (1) needs to be solved to update the parameter. As this particular problem does not have a closed form solution, a solver is usually employed which yields a satisfactory solution. In previous works, different types of solvers have been used. (Cartis et al., 2011a;b) solve the sub-problem using Lanczos based method in Krylov subspace.



(a) (b) (c) (d)

Figure 2. Comparison of the FED−CURE with (a) FedGLOMO and FedAvg (b) for Gaussian attack. (c) Comparison of FED−CURE with robust PGD. (d) Plot of the gradient norm for 'a9a' data-set with Gaussian attack for robust linear regression.

In (Agarwal et al., 2017), the authors propose a solver based on Hessian-vector product and binary search. Gradient descent based solver is proposed in (Carmon & Duchi, 2016; Tripuraneni et al., 2018).

Previous works, (Wang et al., 2020; Zhou et al., 2018; Wang et al., 2019), consider the exact solution of the cubic sub-problem for theoretical analysis. Recently, inexact solutions to the sub-problem is also proposed in the centralized (non-distributed) framework. For instance, (Kohler & Lucchi, 2017) analyzes the cubic model with sub-sampled Hessian with approximate model minimization technique developed in (Cartis et al., 2011a). Moreover, (Tripuraneni et al., 2018) shows improved analysis with gradient based minimization which is a variant studied in (Carmon & Duchi, 2016). Both exact and inexact solutions to the sub-problem yields similar theoretical guarantees.

In our framework, each local machine is tasked with solving the sub-problem. For the purpose of theoretical convergence analysis, we consider that local machines obtain the exact solution in each round. However, in experiments (Section 4), we apply the gradient based solver of (Tripuraneni et al., 2018) to solve the sub-problem. Here, we let each local machines run the gradient based solver for 10 iterations and send the update to the center machine in each iteration.

D. Detailed Experimental results

We now validate on benchmark LIBSVM ((Chang & Lin, 2011)) data-set in both convex and non-convex problems. We choose the following loss functions:

- Logistic loss:$\min_{\mathbf{w}\in\mathsf{R}^d} \frac{1}{n}\sum_i \log\big(1+\exp(-y_i\mathbf{x}_i^T\mathbf{w})\big)+\frac{\lambda}{2n}\|\mathbf{w}\|^2$,
- Non-convex robust linear regression: $\min_{\mathbf{w}\in\mathsf{R}^d} \frac{1}{n}\sum_i \log(\frac{(y_i \quad \mathbf{w}^T\mathbf{x}_i)^2}{2}+1)$,

where $\mathbf{w}\in\mathsf{R}^d$ is the parameter, $\{\mathbf{x}_i\}_{i=1}^n \in \mathsf{R}^d$ are the feature vectors and $\{\mathbf{y}_i\}_{i=1}^n \in \{0,1\}$ are the corresponding labels. We choose 'a9a'($d=123, n\approx 32K$, and split the data into $70/30$ and use as training/testing purpose) and 'w8a'(training data $d=300, n\approx 50K$ and testing data $d=300, n\approx 15K$) classification datasets and partition the data in 20 machines.

We demonstrate FED−CURE in the presence of Byzantine machines and compressed update. For compression, each worker applies compression operator of QSGD (Alistarh et al., 2017). For a given vector $\mathbf{x}\in\mathsf{R}^d$, $[Q(\mathbf{x})]_i = \|\mathbf{x}\|_2 \text{sign}(\mathbf{x}_i)\times \texttt{Ber}(|\mathbf{x}_i|/\|x\|_2)$ for all $i\in[d]$. We consider the following four Byzantine attacks: 1. 'Gaussian Noise attack': where the Byzantine worker machines add Gaussian noise to the update. 2. 'Random label attack': where the Byzantine worker machines train and learn based on random labels instead of the proper labels. 3. 'Flipped label attack': where (for Binary classification) the Byzantine worker machines flip the labels of the data and learn based on wrong labels. 4. 'Negative update attack': where the Byzantine workers computes the update $\mathbf{s}$ (here solves the sub-problem in Eq. (1)) and communicates $-c*\mathbf{s}$ with $c\in(0,1)$ making the direction of the update opposite of the actual one.

Comparison with ByzantinePGD We compare our uncompressed version of FED−CURE ($\delta=1$) with ByzantinePGD of (Yin et al., 2019) here. We take the *total number of iterations* as a comparison metric. One outer iteration of Algorithm 1 corresponds to one round of communication between the center and the worker machines (and hence one parameter update). Note that in our algorithm the worker machines use 10 steps of gradient solver (see (Tripuraneni et al., 2018)) for the local sub

problem per iteration. So, the *total number of iterations* is given by 10 times the number of outer iterations. For both the algorithms, we choose ℓ_2 norm of the gradient as a stopping criteria. For ByzantinePGD, we choose $R=10, r=5, Q=10, T_{th}=10$ and 'co-ordinate wise Trimmed mean. In the Figure 1 (b-d), we plot the *total number of iterations* in all four types of attacks with different fraction of Byzantine machines. It is evident from the plot that our method requires less number of over all iterations (at least 48.4%, 29% and 25% less for 10%, 15% and 20% of Byzantine machines respectively).

Although FED−CURE uses Hessian (second order) information, the sub-problem actually uses gradient based first order algorithm, and hence we compare the total iteration complexity mentioned above. To the best of our knowledge, there is no saddle point avoidance second order algorithm in FL framework, and so we adhere to the comparison with first order methods.

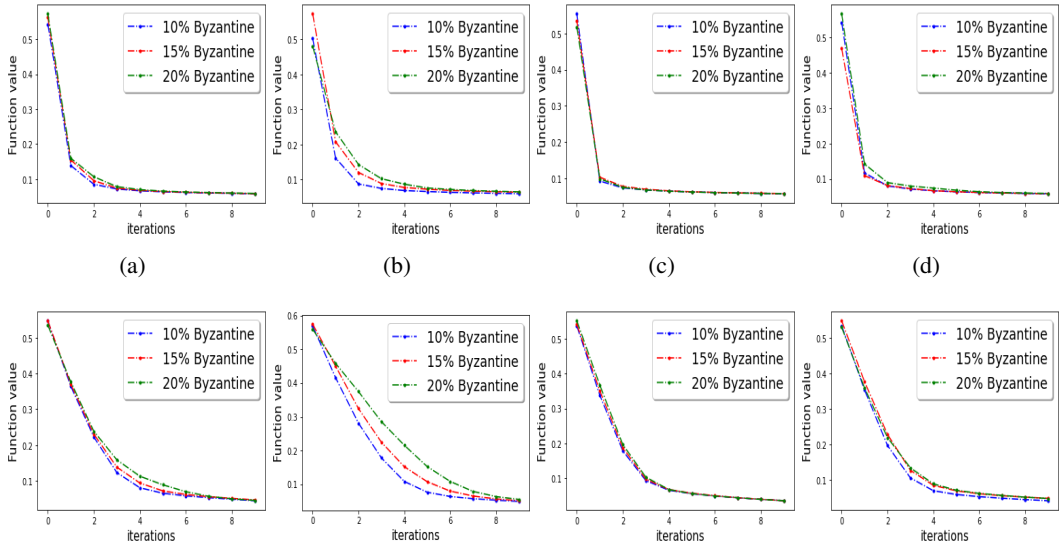


Figure 3. Non convex robust linear regression with 'a9a' (a,b,c,d) and 'w8a' (e,f,g,h) with 10%,15%,20% Byzantine worker machines for (a,e). Flipped label attack.(b,f). Negative Update attack. (c,g) Gaussian attack. (d,h) Random attack.

Comparison with standard FL algorithms We have implemented and compared the performance of standard FL algorithm like FedGLOMO (Das et al., 2020) (Federated Learning via Global and Local Momentum) and FedAvg (McMahan et al., 2017) with FED−CURE. The results are shown in Figure 2(a,b). Our method outperforms these standard baselines since they can tolerate Byzantine attacks (Gaussian attack in the experiment).

Training loss for compressed update In Figure 3, we plot the function value of the robust linear regression problem for 'flipped labels', 'negative update', 'Gaussian' and 'Random label' attacks with compressed update for both 'w8a' and 'a9a' datasets. We choose the parameters $\lambda=1, M=10$, learning rate $\eta_k=1$, $\alpha=\{.1,.15,.2\}$ and $\beta=\alpha+\frac{2}{m}$, where number of worker machines $m=20$. In Figure 2(d), we plot the gradient norm ($\|\mathbf{g}\|$) for Gaussian attack with 10%,15% and 20% of machines being Byzantine.

Classification accuracy We show the classification accuracy on testing data of 'a9a' and 'w8a' dataset for logistic regression problem in Figure 4 and training function loss of 'a9a' and 'w8a' dataset for robust linear regression problem in the Figure 4. It is evident from the plots that a simple *norm based thresholding* makes the learning algorithm robust.

Training loss for uncompressed update In Figure 5, we plot the function value of the robust linear regression with the similar attacks for the uncompressed update ($\delta=1$) for both 'w8a' and 'a9a' dataset.

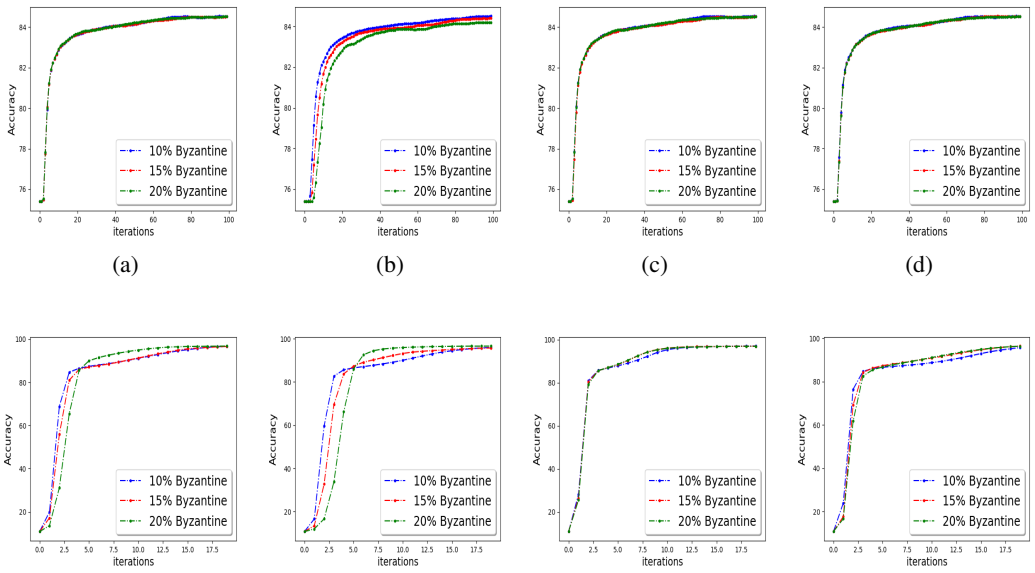


Figure 4. Classification accuracy of the testing data 'a9a' dataset (first row) and 'w8a' dataset (second row) with 10%,15%,20% Byzantine worker machines for (a,e). Flipped label.(b,f). Negative Update (c,g). Gaussian noise and (d,h). Random label attack for logistic regression problem.

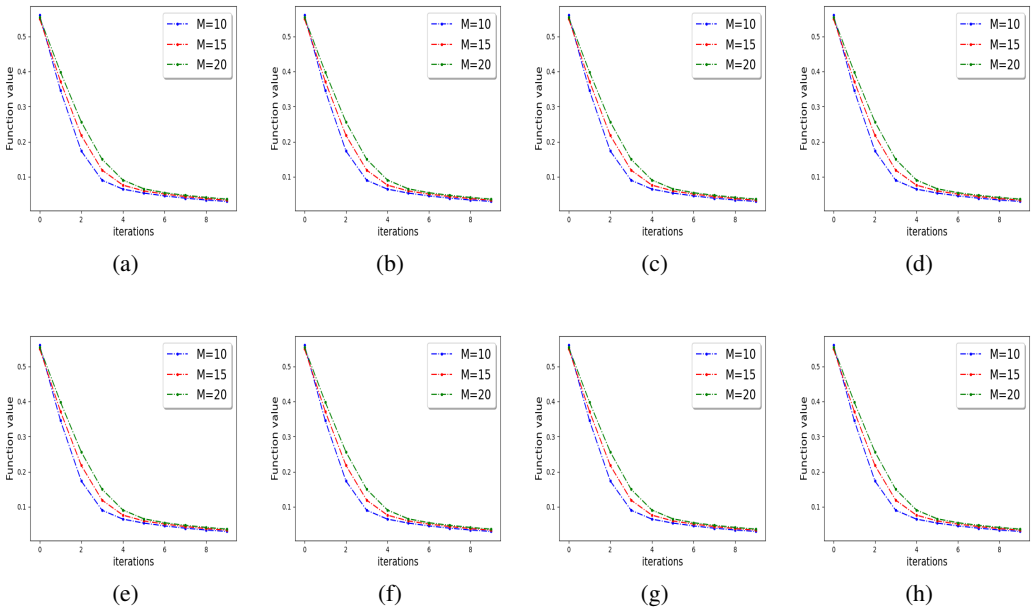


Figure 5. Function training loss for the training data 'a9a' (first row) and 'w8a' (second row) with 10%, 15%, 20% Byzantine worker machines. for (a,e). Gaussian attack.(b,f).Random attack (c,g) Flipped Attack and (d,h) Negative update attack for non-convex robust linear regression problem.

E. Proofs of main results

In this part, we establish some useful facts and lemmas. Next, we provide analysis of Theorems 3.5.

E.1. Some useful facts

For the purpose of analysis we use the following sets of inequalities.

Fact 1. For $a_1,...,a_n$ we have the following inequality

$$\|\left(\sum_{i=1}^{n} a_i\right)\|^3 \leq \left(\sum_{i=1}^{n}\|a_i\|\right)^3 \leq n^2\sum_{i=1}^{n}\|a_i\|^3 \tag{2}$$

$$\|\left(\sum_{i=1}^{n} a_i\right)\|^2 \leq \left(\sum_{i=1}^{n}\|a_i\|\right)^2 \leq n\sum_{i=1}^{n}\|a_i\|^2 \tag{3}$$

Fact 2. For $a_1,...,a_n > 0$ and $r < s$

$$\left(\frac{1}{n}\sum_{i=1}^{n} a_i^r\right)^{1/r} \leq \left(\frac{1}{n}\sum_{i=1}^{n} a_i^s\right)^{1/s} \tag{4}$$

Lemma E.1 ((Nesterov & Polyak, 2006)). *Under Assumption 3.2, i.e., the Hessian of the function is L_2-Lipschitz continuous, for any $\mathbf{x},\mathbf{y} \in \mathsf{R}^d$, we have*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) - \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x})\| \leq \frac{L_2}{2}\|\mathbf{y}-\mathbf{x}\|^2 \tag{5}$$

$$\left| f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) - \frac{1}{2}(\mathbf{y}-\mathbf{x})^T\nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x})\right| \leq \frac{L_2}{6}\|\mathbf{y}-\mathbf{x}\|^2 \tag{6}$$

Next, we establish the following Lemma that provides some nice properties of the cubic sub-problem.

Lemma E.2. *Let $M > 0, \gamma > 0, \mathbf{g} \in \mathsf{R}^d, \mathbf{H} \in \mathsf{R}^{d\ \ d}$, and*

$$\mathbf{s} = \operatorname{argmin}_{\mathbf{x}} \mathbf{g}^T\mathbf{x} + \frac{\gamma}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \frac{M\gamma^2}{6}\|\mathbf{x}\|^3. \tag{7}$$

The following holds

$$\mathbf{g} + \gamma\mathbf{H}\mathbf{s} + \frac{M\gamma^2}{2}\|\mathbf{s}\|\mathbf{s} = 0, \tag{8}$$

$$\mathbf{H} + \frac{M\gamma}{2}\|\mathbf{s}\|\mathbf{I} \succeq \mathbf{0}, \tag{9}$$

$$\mathbf{g}^T\mathbf{s} + \frac{\gamma}{2}\mathbf{s}^T\mathbf{H}\mathbf{s} \leq -\frac{M}{4}\gamma^2\|\mathbf{s}\|^3. \tag{10}$$

Proof. The equations (8) and (9) are from the first and second order optimal condition. We proof (10), by using the conditions of (8) and (9).

$$\mathbf{g}^T\mathbf{s} + \frac{\gamma}{2}\gamma\mathbf{s}^T\mathbf{H}\mathbf{s} = -\left(\gamma\mathbf{H}\mathbf{s} + \frac{M}{2}\gamma^2\|\mathbf{s}\|\mathbf{s}\right)^T\mathbf{s} + \frac{\gamma}{2}\gamma\mathbf{s}^T\mathbf{H}\mathbf{s} \tag{11}$$

$$= -\gamma\mathbf{s}^T\mathbf{H}\mathbf{s} - \frac{M}{2}\gamma^2\|\mathbf{s}\|^3 + \frac{\gamma}{2}\gamma\mathbf{s}^T\mathbf{H}\mathbf{s}$$

$$\leq \frac{M}{4}\gamma^2\|\mathbf{s}\|^3 - \frac{M}{2}\gamma^2\|\mathbf{s}\|^3 \tag{12}$$

$$= -\frac{M}{4}\gamma^2\|\mathbf{s}\|^3.$$

In (11), we substitute the expression $\mathbf{g}$ from the equation (8). In (12), we use the fact that $\mathbf{s}^T\mathbf{H}\mathbf{s} + \frac{M\gamma}{2}\|\mathbf{s}\|^3 > 0$ from the equation (9). □

E.2. Proof of Theorem 3.5

First we state the results of Lemma E.2 for each worker machine in iteration k,

$$\mathbf{g}_{i,k}+\gamma\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}+\frac{M}{2}\gamma^2\|\mathbf{s}_{i,k+1}\|\mathbf{s}_{i,k+1}=0 \tag{13}$$

$$\gamma\mathbf{H}_{i,k}+\frac{M}{2}\gamma^2\|\mathbf{s}_{i,k+1}\|\mathbf{I}\succeq\mathbf{0} \tag{14}$$

$$\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\frac{\gamma}{2}\mathbf{s}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\leq-\frac{M}{4}\gamma^2\|\mathbf{s}_{i,k+1}\|^3 \tag{15}$$

We also use the following fact form the setup and trimming set

$$|\mathcal{U}|=|\mathcal{U}\cap\mathcal{M}|+|\mathcal{U}\cap\mathcal{B}| \tag{16}$$

$$|\mathcal{M}|=|\mathcal{U}\cap\mathcal{M}|+|\mathcal{T}\cap\mathcal{M}| \tag{17}$$

Combining both the equations (16) and (17), we have

$$|\mathcal{U}|=|\mathcal{M}|-|\mathcal{T}\cap\mathcal{M}|+|\mathcal{U}\cap\mathcal{B}| \tag{18}$$

Now we state the following fact from the trimming set. as mentioned in the Algorithm 1, the norm of the update from any worker machines from the set $\mathcal{U}$ is less than the norm of the update from any worker machines in the set $\mathcal{T}$. Now as $\beta>\alpha$, at least one good machine (the largest norm) is in the set $\mathcal{T}$. So, we can claim the following,

$$\text{For all } i\in\mathcal{U}\cap\mathcal{B},\quad \|\mathbf{s}_i\|\leq\max_{i\in M}\|\mathbf{s}_i\|$$

Summing over all the Byzantine machines in the untrimmed set which is $\mathcal{U}\cap\mathcal{B}$, we get

$$\sum_{i\in U\setminus B}\|.\|\leq\alpha m\max_{i\in M}\|.\|$$

as $|\mathcal{U}\cap\mathcal{B}|\leq\alpha m$. Also,

$$\sum_{i\in M\setminus T}\|.\|\leq\sum_{i\in M}\|.\|$$

Combining the above two equations, we get

$$\sum_{i\in U\setminus B}\|.\|+\sum_{i\in M\setminus T}\|.\|\leq\sum_{i\in M}\|.\|+\alpha m\max_{i\in M}\|.\| \tag{19}$$

For the rest of the calculation, we use the following notation

$$\Gamma=\max_{i\in M,k}\|\mathbf{s}_{i,k}\|. \tag{20}$$

If the optimization sub-problem domain is bounded, Γ can be upper-bounded by the diameter of the parameter space. Note that in the definition of Γ, the maximum is taken over good machines only.

Characterization of Γ : For any good worker machine $i\in\mathcal{M}$, we have the following

$$\mathbf{s}_{i,k+1}=\operatorname{argmin}_{\mathbf{s}}\mathbf{g}_{i,k}^T\mathbf{s}+\frac{\gamma}{2}\mathbf{s}^T\mathbf{H}_{i,k}\mathbf{s}+\gamma^2\frac{M}{6}\|\mathbf{s}\|^3$$

for some $M>0$ and $\gamma=\frac{c}{T}$. Next, we consider $\mathbf{u}_{i,k+1}=\gamma\mathbf{s}_{i,k+1}$ and we get the following expression

$$\mathbf{u}_{i,k+1}=\operatorname{argmin}_{\mathbf{u}}\mathbf{g}_{i,k}^T\mathbf{u}+\frac{1}{2}\mathbf{u}^T\mathbf{H}_{i,k}\mathbf{u}+\frac{M}{6}\|\mathbf{u}\|^3$$

Following the similar results of the E.2, we have the following result from the second order condition,

$$\mathbf{g}_{i,k}^T\mathbf{u}_{i,k+1}+\frac{1}{2}\mathbf{u}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{u}_{i,k+1}\leq-\frac{M}{4}\|\mathbf{u}_{i,k+1}\|^3.$$

Therefore,

$$\begin{aligned}
&\frac{M}{4}\|\mathbf{u}_{i,k+1}\|^3\\
\leq&\|\mathbf{g}_{i,k}\|\mathbf{u}_{i,k+1}\|+\frac{1}{2}\|\mathbf{H}_{i,k}\|\|\mathbf{u}_{i,k+1}\|^2\\
=&\|\mathbf{g}_{i,k}-\nabla f(\mathbf{s}_{i,k+1})+\nabla f(\mathbf{s}_{i,k+1})\|\mathbf{u}_{i,k+1}\|+\frac{1}{2}\|\mathbf{H}_{i,k}-\nabla^2 f(\mathbf{s}_{i,k+1})+\nabla^2 f(\mathbf{s}_{i,k+1})\|\|\mathbf{u}_{i,k+1}\|^2\\
\leq&(\|\mathbf{g}_{i,k}-\nabla f(\mathbf{s}_{i,k+1})\|+\|\nabla f(\mathbf{s}_{i,k+1})\|\mathbf{u}_{i,k+1}\|+\frac{1}{2}(\|\mathbf{H}_{i,k}-\nabla^2 f(\mathbf{s}_{i,k+1})\|+\|\nabla^2 f(\mathbf{s}_{i,k+1})\|)\|\mathbf{u}_{i,k+1}\|^2\\
\leq&(\epsilon_g+L)\|\mathbf{u}_{i,k+1}\|+(\epsilon_H+L_1)\|\mathbf{u}_{i,k+1}\|^2
\end{aligned}$$

In the above expression, we have ϵ_g, ϵ_H are gradient and Hessian dissimilarity respectively and $\|\nabla f(\mathbf{s}_{i,k+1}\| \leq L, \|\nabla^2 f(\mathbf{s}_{i,k+1}\| \leq L_1$ which are constants. This shows that $\|\mathbf{u}_{i,k+1}\|$ to be bounded and hence $\max_{i\in M}\|\mathbf{u}_{i,k+1}\|$ to be bounded. For $\gamma=\frac{c}{T}$, we have

$$\begin{aligned}
\|\mathbf{s}_{i,k+1}\|=\|\mathbf{u}_{i,k+1}/\gamma\|=O(T)\\
\Rightarrow\Gamma=O(T)
\end{aligned}\tag{21}$$

From the definition of the δ-approximate compressor in Definition 1.1, we use the following simple fact

$$\|Q(\mathbf{x})\|\leq(1+\sqrt{1-\delta})\|\mathbf{x}\|\tag{22}$$

At any iteration k, we have (with Taylor's expansion)

$$\begin{aligned}
&f(\mathbf{x}_{k+1})-f(\mathbf{x}_k)\\
\leq&\nabla f(\mathbf{x}_k)^T(\mathbf{x}_{k+1}-\mathbf{x}_k)+\frac{1}{2}(\mathbf{x}_{k+1}-\mathbf{x}_k)^T\nabla^2 f(\mathbf{x}_k)(\mathbf{x}_{k+1}-\mathbf{x}_k)+\frac{L_2}{6}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^3\\
=&\underbrace{\frac{\eta_k}{|\mathcal{U}|}\nabla f(\mathbf{x}_k)^T\sum_{i\in U}Q(\mathbf{s}_{i,k+1})}_{Term1}+\underbrace{\frac{\eta_k^2}{2|\mathcal{U}|^2}\left(\sum_{i\in U}Q(\mathbf{s}_{i,k+1})\right)^T\nabla^2 f(\mathbf{x}_k)\left(\sum_{i\in U}Q(\mathbf{s}_{i,k+1})\right)}_{Term2}\\
&+\underbrace{\frac{L_2}{6}\left\|\frac{\eta_k}{|\mathcal{U}|}\sum_{i\in U}Q(\mathbf{s}_{i,k+1})\right\|^3}_{Term3}
\end{aligned}\tag{23}$$

In 23, we use the update of the parameter in the center machine $\mathbf{x}_{k+1}-\mathbf{x}_k=\frac{\eta_k}{|U|}\sum_{i\in U}Q(\mathbf{s}_{i,k+1})$, as expressed in the Algorithm 1.

First we choose the Term 1 in (23) and expand it using (18)

$$\begin{aligned}
&\frac{\eta_k}{|\mathcal{U}|}\nabla f(\mathbf{x}_k)^T\sum_{i\in U} Q(\mathbf{s}_{i,k+1}) \\
=&\frac{\eta_k}{(1-\beta)m}\nabla f(\mathbf{x}_k)^T\left[\sum_{i\in M} Q(\mathbf{s}_{i,k+1})-\sum_{i\in M\setminus T} Q(\mathbf{s}_{i,k+1})+\sum_{i\in U\setminus B} Q(\mathbf{s}_{i,k+1})\right] \\
=&\underbrace{\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\left[\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\nabla f(\mathbf{x}_k)^T Q(\mathbf{s}_{i,k+1})-\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}\right]}_{Term 1.1} \\
&+\underbrace{\frac{\eta_k}{(1-\beta)m}\nabla f(\mathbf{x}_k)^T\left[-\sum_{i\in M\setminus T} Q(\mathbf{s}_{i,k+1})+\sum_{i\in U\setminus B} Q(\mathbf{s}_{i,k+1})\right]}_{Term 1.2} \qquad (24)
\end{aligned}$$

First we consider Term 1.1 in (24) (notice that the sum is over only good machines),

$$\begin{aligned}
&\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\left[\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\nabla f(\mathbf{x}_k)^T Q(\mathbf{s}_{i,k+1})-\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}\right] \\
=&\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\left[\nabla f(\mathbf{x}_k)^T Q(\mathbf{s}_{i,k+1})-\nabla f(\mathbf{x}_k)^T\mathbf{s}_{i,k+1}+\nabla f(\mathbf{x}_k)^T\mathbf{s}_{i,k+1}-\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}\right] \\
=&\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\left[\nabla f(\mathbf{x}_k)^T(Q(\mathbf{s}_{i,k+1})-\mathbf{s}_{i,k+1})+(\nabla f(\mathbf{x}_k)-\mathbf{g}_{i,k})^T\mathbf{s}_{i,k+1}\right] \\
\le&\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\left[\|\nabla f(\mathbf{x}_k)\|\|Q(\mathbf{s}_{i,k+1})-\mathbf{s}_{i,k+1}\|+\|\nabla f(\mathbf{x}_k)-\mathbf{g}_{i,k}\|\|\mathbf{s}_{i,k+1}\|\right] \\
\le&\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\left[L\sqrt{1-\delta}\|\mathbf{s}_{i,k+1}\|+\epsilon_g\|\mathbf{s}_{i,k+1}\|\right] \qquad (25) \\
\le&\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\frac{\eta_k(1-\alpha)}{(1-\beta)}(L\sqrt{1-\delta}+\epsilon_g)\Gamma \qquad (26)
\end{aligned}$$

In (25), we use the following facts: 1. $\|\nabla f(\mathbf{x}_k)\| \le L$ as the function f is L- Lipschitz. 2. $\|Q(\mathbf{s}_{i,k+1}) - \mathbf{s}_{i,k+1}\| \le \sqrt{1-\delta}\|\mathbf{s}_{i,k+1}\|$ by definition of the δ-compressor. 3. $\|\nabla f(\mathbf{x}_k)-\mathbf{g}_{i,k}\| \le \epsilon_g$ (gradient dissimilarity). In (26), we use the bound stated in (20).

Next we consider Term1.2 in (24),

$$\begin{aligned}
&\frac{\eta_k}{(1-\beta)m}\nabla f(\mathbf{x}_k)^T\left[-\sum_{i\in M\setminus T} Q(\mathbf{s}_{i,k+1})+\sum_{i\in U\setminus B} Q(\mathbf{s}_{i,k+1})\right] \\
\le&\frac{\eta_k}{(1-\beta)m}\left[\sum_{i\in M\setminus T}\|\nabla f(\mathbf{x}_k)\|\|Q(\mathbf{s}_{i,k+1})\|+\sum_{i\in U\setminus B}\|\nabla f(\mathbf{x}_k)\|\|Q(\mathbf{s}_{i,k+1})\|\right] \\
\le&\frac{\eta_k L}{(1-\beta)m}\left[\sum_{i\in M\setminus T}\|Q(\mathbf{s}_{i,k+1})\|+\sum_{i\in U\setminus B}\|Q(\mathbf{s}_{i,k+1})\|\right] \qquad (27) \\
\le&\frac{\eta_k L}{(1-\beta)m}\left[\sum_{i\in T}\max_{i\in M}\|Q(\mathbf{s}_{i,k+1})\|+\sum_{i\in B}\max_{i\in M}\|Q(\mathbf{s}_{i,k+1})\|\right] \qquad (28) \\
\le&\frac{\eta_k L}{(1-\beta)m}\left[\beta m\max_{i\in M}\|Q(\mathbf{s}_{i,k+1})\|+\alpha m\max_{i\in M}\|Q(\mathbf{s}_{i,k+1})\|\right] \qquad (29) \\
\le&\frac{\eta_k(\alpha+\beta)L}{(1-\beta)}(1+\sqrt{1-\delta})\left[\max_{i\in M}\|\mathbf{s}_{i,k+1}\|)\right] \qquad (30) \\
\le&\frac{\eta_k(\alpha+\beta)L}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma \qquad (31)
\end{aligned}$$

We use the fact $\|\nabla f(\mathbf{x}_k)\| \leq L$ in (27), ,the fact stated in (19), in (28). We use the definition of δ-compressor in (30) and the bound of update as described in (20) in (31).

We apply the bound derived for Term1.1 in (26) and for Term1.2 in (31) in the bound for Term1 in (24) and derive the following,

$$
\begin{aligned}
&Term1\\
\leq&\frac{\eta_k}{(1-\beta)m}\sum_{i2\,M}\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\frac{\eta_k}{(1-\beta)}(L\sqrt{1-\delta}+\epsilon_g)\Gamma+\frac{\eta_k\alpha L}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma\\
=&\frac{\eta_k}{(1-\beta)m}\sum_{i2\,M}\left[\mathbf{g}_{i,k}^T\mathbf{s}_{i,k+1}+\frac{\gamma}{2}\mathbf{s}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\right]-\frac{\eta_k}{(1-\beta)m}\sum_{i2\,M}\frac{\gamma}{2}\mathbf{s}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\\
&+\frac{\eta_k}{(1-\beta)}(L\sqrt{1-\delta}+\epsilon_g)\Gamma+\frac{\eta_k\alpha L}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma\\
\leq&-\frac{\gamma^2M\eta_k}{4(1-\beta)m}\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\|^3-\frac{\eta_k}{(1-\beta)m}\sum_{i2\,M}\frac{\gamma}{2}\mathbf{s}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\\
&+\frac{\eta_k(1-\alpha)}{(1-\beta)}(L\sqrt{1-\delta}+\epsilon_g)\Gamma+\frac{\eta_k(\alpha+\beta)L}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma
\end{aligned}
\tag{32}
$$

In line (32), we use the bound stated in (15).

Now we consider the Term 3 in equation (23),

$$
\begin{aligned}
&\frac{L_2}{6}\left\|\frac{\eta_k}{|\mathcal{U}|}\sum_{i2U}Q(\mathbf{s}_{i,k+1})\right\|^3\\
\leq&\frac{L_2\eta_k^3}{6|\mathcal{U}|}\sum_{i2U}\|Q(\mathbf{s}_{i,k+1})\|^3 &(33)\\
\leq&\frac{L_2\eta_k^3}{6|\mathcal{U}|}\left[\sum_{i2\,M}\|Q(\mathbf{s}_{i,k+1})\|^3-\sum_{i2\,M\setminus T}\|Q(\mathbf{s}_{i,k+1})\|^3+\sum_{i2U\setminus B}\|Q(\mathbf{s}_{i,k+1})\|^3\right] &(34)\\
\leq&\frac{L_2\eta_k^3}{6|\mathcal{U}|}\left[\sum_{i2\,M}\|Q(\mathbf{s}_{i,k+1})\|^3+\sum_{i2U\setminus B}\|Q(\mathbf{s}_{i,k+1})\|^3\right] &(35)\\
\leq&\frac{L_2\eta_k^3}{6(1-\beta)m}\left[\sum_{i2\,M}\|Q(\mathbf{s}_{i,k+1})\|^3+\alpha m\max_{i2\,M}\|Q(\mathbf{s}_{i,k+1})\|^3\right] &(36)\\
\leq&\frac{L_2\eta_k^3}{6(1-\beta)m}\left[\sum_{i2\,M}(1+\sqrt{1-\delta})^3\|\mathbf{s}_{i,k+1}\|^3+\alpha m\max_{i2\,M}(1+\sqrt{1-\delta})^3\|\mathbf{s}_{i,k+1}\|^3\right] &(37)\\
\leq&\frac{L_2\eta_k^3}{6(1-\beta)m}(1+\sqrt{1-\delta})^3\left[\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\|^3+\alpha m\Gamma^3\right] &(38)
\end{aligned}
$$

In (33), we use the fact stated in (2). Next in (34), we expand the trimmed set $\mathcal{U}$ using (18) and in (36), we use the bound of (19). Finally, in (37), we use the definition of the δ-compressor and the bound stated in (20) in (38).

Now we consider the Term 2 in (23)

$$\begin{aligned}
&\frac{\eta_k^2}{2|\mathcal{U}|^2}\left(\sum_{i\in U} Q(\mathbf{s}_{i,k+1})\right)^T \nabla^2 f(\mathbf{x}_k)\left(\sum_{i\in U} Q(\mathbf{s}_{i,k+1})\right)\\
&= \frac{\eta_k^2}{2(1-\beta)^2 m^2}\underbrace{\sum_{i\in U} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1})}_{Term 2.1}\\
&+ \frac{\eta_k^2}{2(1-\beta)^2 m^2}\underbrace{\sum_{i\neq j\in U} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{j,k+1})}_{Term 2.2}
\end{aligned} \tag{39}$$

Now we consider Term2.1 in (39) and expand it using (18)

$$\begin{aligned}
&\sum_{i\in U} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1})\\
&= \underbrace{\sum_{i\in M} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1})}_{Term 2.1.1} - \underbrace{\sum_{i\in M\setminus T} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1}) + \sum_{i\in B\setminus U} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1})}_{Term 2.1.2}
\end{aligned}$$

We consider Term2.1.1

$$\begin{aligned}
&\sum_{i\in M} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1})\\
&= \sum_{i\in M} \mathbf{s}_{i,k+1}^T \mathbf{H}_{i,k} \mathbf{s}_{i,k+1} - \sum_{i\in M} \mathbf{s}_{i,k+1}^T \mathbf{H}_{i,k} \mathbf{s}_{i,k+1} + \sum_{i\in M} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1})\\
&= \sum_{i\in M} \mathbf{s}_{i,k+1}^T \mathbf{H}_{i,k} \mathbf{s}_{i,k+1} - \sum_{i\in M} \mathbf{s}_{i,k+1}^T (\mathbf{H}_{i,k} - \nabla^2 f(\mathbf{x}_k)\mathbf{s}_{i,k+1} - \sum_{i\in M} \mathbf{s}_{i,k+1}^T \nabla^2 f(\mathbf{x}_k)\mathbf{s}_{i,k+1} +\\
&\quad + \sum_{i\in M} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1}) - \sum_{i\in M} \mathbf{s}_{i,k+1}^T \nabla^2 f(\mathbf{x}_k) Q(\mathbf{s}_{i,k+1}) + \sum_{i\in M} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k)\mathbf{s}_{i,k+1}\\
&= \sum_{i\in M} \mathbf{s}_{i,k+1}^T \mathbf{H}_{i,k} \mathbf{s}_{i,k+1} - \sum_{i\in M} \mathbf{s}_{i,k+1}^T (\mathbf{H}_{i,k} - \nabla^2 f(\mathbf{x}_k)\mathbf{s}_{i,k+1} - \sum_{i\in M} (Q(\mathbf{s}_{i,k+1}) - \mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k)\mathbf{s}_{i,k+1} +\\
&\quad + \sum_{i\in M} Q(\mathbf{s}_{i,k+1})^T \nabla^2 f(\mathbf{x}_k)(Q(\mathbf{s}_{i,k+1}) - \mathbf{s}_{i,k+1})\\
&\leq \sum_{i\in M} \mathbf{s}_{i,k+1}^T \mathbf{H}_{i,k} \mathbf{s}_{i,k+1} + \sum_{i\in M} \|(\nabla^2 f(\mathbf{x}_k) - \mathbf{H}_{i,k})\|\|\mathbf{s}_{i,k+1}\|^2 + 2\sum_{i\in M} \|\nabla^2 f(\mathbf{x}_k)\|\|\mathbf{s}_{i,k+1}\|\|Q(\mathbf{s}_{i,k+1}) - \mathbf{s}_{i,k+1}\|\\
&\leq \sum_{i\in M} \mathbf{s}_{i,k+1}^T \mathbf{H}_{i,k} \mathbf{s}_{i,k+1} + \sum_{i\in M} \epsilon_H \|\mathbf{s}_{i,k+1}\|^2 + 2\sum_{i\in M} L_1\sqrt{1-\delta}\|\mathbf{s}_{i,k+1}\|^2 \qquad (40)\\
&\leq \sum_{i\in M} \mathbf{s}_{i,k+1}^T \mathbf{H}_{i,k} \mathbf{s}_{i,k+1} + \sum_{i\in M} (\epsilon_H + 2L_1\sqrt{1-\delta})\|\mathbf{s}_{i,k+1}\|^2 \qquad (41)\\
&\leq \sum_{i\in M} \mathbf{s}_{i,k+1}^T \mathbf{H}_{i,k} \mathbf{s}_{i,k+1} + (1-\alpha)m(\epsilon_H + 2L_1\sqrt{1-\delta})\Gamma^2 \qquad (42)
\end{aligned}$$

In 40, we use the Hessian dissimilarity bound of $\|(\nabla^2 f(\mathbf{x}_k) - \mathbf{H}_{i,k})\| \leq \epsilon_H$ and the Hessian bound of $\|(\nabla^2 f(\mathbf{x}_k)\| \leq L_1$. And in 41, we apply the definition of the δ-compressor.

Next, we consider the Term2.1.2,

$$\begin{aligned}
&\sum_{i\in M\setminus T} Q(\mathbf{s}_{i,k+1})^T\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})+\sum_{i\in B\setminus U} Q(\mathbf{s}_{i,k+1})^T\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})\\
&\leq \sum_{i\in M\setminus T} L_1\|Q(\mathbf{s}_{i,k+1})\|^2+\sum_{i\in B\setminus U} L_1\|Q(\mathbf{s}_{i,k+1})\|^2 \qquad (43)\\
&\leq \sum_{i\in B}\max_{i\in T} L_1(1+\sqrt{1-\delta})^2\|\mathbf{s}_{i,k+1}\|^2+\sum_{i\in B}\max_{i\in M} L_1(1+\sqrt{1-\delta})^2\|\mathbf{s}_{i,k+1}\|^2 \qquad (44)\\
&\leq \beta m L_1(1+\sqrt{1-\delta})^2\Gamma^2+\alpha m L_1(1+\sqrt{1-\delta})^2\Gamma^2\\
&= (\alpha+\beta)mL_1(1+\sqrt{1-\delta})^2\Gamma^2 \qquad (45)
\end{aligned}$$

Combining (42) and (45), we bound the Term2.1,

$$\begin{aligned}
&\text{Term2.1}\\
&\leq \sum_{i\in M}\mathbf{s}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}+(1-\alpha)m(\epsilon_H+2L_1\sqrt{1-\delta})\Gamma^2+(\alpha+\beta)mL_1(1+\sqrt{1-\delta})^2\Gamma^2 \qquad (46)
\end{aligned}$$

Now we consider the Term 2.2 in equation (39)

$$\begin{aligned}
&\sum_{i\neq j\in U} Q(\mathbf{s}_{i,k+1})^T\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{j,k+1})\\
&\leq \sum_{i\neq j\in U} L_1(1+\sqrt{1-\delta})^2\|\mathbf{s}_{i,k+1}\|\|\mathbf{s}_{j,k+1}\| \qquad (47)\\
&= L_1(1+\sqrt{1-\delta})^2\left[\|\sum_{i\in U}\mathbf{s}_{i,k+1}\|^2-\sum_{i\in U}\|\mathbf{s}_{i,k+1}\|^2\right]\\
&\leq L_1(1+\sqrt{1-\delta})^2\left[|\mathcal{U}|\sum_{i\in U}\|\mathbf{s}_{i,k+1}\|^2-\sum_{i\in U}\|\mathbf{s}_{i,k+1}\|^2\right]\\
&= L_1(1+\sqrt{1-\delta})^2((1-\beta)m-1)\left[\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^2-\sum_{i\in M\setminus T}\|\mathbf{s}_{i,k+1}\|^2+\sum_{i\in B\setminus U}\|\mathbf{s}_{i,k+1}\|^2\right] \qquad (48)\\
&\leq L_1(1+\sqrt{1-\delta})^2((1-\beta)m-1)\left[\sum_{i\in U}\|\mathbf{s}_{i,k+1}\|^2\right] \qquad (49)\\
&= L_1(1+\sqrt{1-\delta})^2((1-\beta)m-1)(1-\beta)m\Gamma^2 \qquad (50)
\end{aligned}$$

In (47), we apply the definition of δ compressor. We use the expansion described in (18) in (48).

Now combining the results in (50) and (39) we get,

$$\begin{aligned}
&Term2\\
&\leq \frac{\eta_k^2}{2(1-\beta)^2m^2}\left[\sum_{i\in M}\mathbf{s}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}+(1-\alpha)m(\epsilon_H+2L_1\sqrt{1-\delta})\Gamma^2+(\alpha+\beta)mL_1(1+\sqrt{1-\delta})^2\Gamma^2\right]\\
&+\frac{\eta_k^2}{2(1-\beta)^2m^2}L_1(1+\sqrt{1-\delta})^2((1-\beta)m-1)(1-\beta)m\Gamma^2
\end{aligned}$$

Now we combine all the upper bound of the Term 1, Term 2 and Term 3

$$
\begin{aligned}
&f(\mathbf{x}_{k+1})-f(\mathbf{x}_k)\\
\leq&-\frac{\gamma^2 M\eta_k}{4(1-\beta)m}\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^3-\frac{\eta_k}{(1-\beta)m}\sum_{i\in M}\frac{\gamma}{2}\mathbf{s}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\\
&+\frac{\eta_k(1-\alpha)}{(1-\beta)}(L\sqrt{1-\delta}+\epsilon_g)\Gamma+\frac{\eta_k(\alpha+\beta)L}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma\\
&+\frac{\eta_k^2}{2(1-\beta)^2m}\Big[(1-\alpha)(\epsilon_H+L_1+L_1(1+\sqrt{1-\delta})^2)+(\alpha+\beta)L_1(1+\sqrt{1-\delta})^2\Big]\Gamma^2\\
&+\frac{\eta_k^2}{2}L_1(1+\sqrt{1-\delta})^2\Gamma^2+\frac{L_2\eta_k^3}{6(1-\beta)m}(1+\sqrt{1-\delta})^3\left[\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^3+\alpha m\Gamma^3\right]\\
=&\left(-\frac{\gamma^2 M\eta_k}{4(1-\beta)m}+\frac{L_2\eta_k^3}{6(1-\beta)m}(1+\sqrt{1-\delta})^3\right)\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^3-\frac{\eta_k}{2(1-\beta)m}\left(\gamma-\frac{\eta_k}{(1-\beta)m}\right)\mathbf{s}_{i,k+1}^T\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\\
&+\left(\frac{\eta_k(1-\alpha)}{(1-\beta)}(L\sqrt{1-\delta}+\epsilon_g)+\frac{\eta_k(\alpha+\beta)L}{(1-\beta)}(1+\sqrt{1-\delta})\right)\Gamma+\frac{L_2\eta_k^3}{6(1-\beta)}(1+\sqrt{1-\delta})^3\alpha\Gamma^3\\
&+\frac{\eta_k^2}{2(1-\beta)^2m}\Big((1-\alpha)(\epsilon_H+2L_1\sqrt{1-\delta})+(\alpha+\beta)L_1(1+\sqrt{1-\delta})^2)+L_1(1+\sqrt{1-\delta})^2((1-\beta)m-1)(1-\beta)m\Big)\Gamma^2
\end{aligned}
$$

Also we assume that $\gamma\geq\frac{\eta_k}{(1-\beta)m}$ and use the fact $-\mathbf{s}_{i,k+1}\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\leq\frac{M\gamma}{2}\|\mathbf{s}_{i,k+1}\|^3$. We also choose that

$$
\begin{aligned}
\lambda\ =&\left(\frac{\eta_k(1-\alpha)}{(1-\beta)}(L\sqrt{1-\delta}+\epsilon_g)+\frac{\eta_k(\alpha+\beta)L}{(1-\beta)}(1+\sqrt{1-\delta})\right)\Gamma+\frac{L_2\eta_k^3}{6(1-\beta)}(1+\sqrt{1-\delta})^3\alpha\Gamma^3\\
&+\frac{\eta_k^2}{2(1-\beta)^2m}\Big((1-\alpha)(\epsilon_H+2L_1\sqrt{1-\delta})+(\alpha+\beta)L_1(1+\sqrt{1-\delta})^2)+L_1(1+\sqrt{1-\delta})^2((1-\beta)m-1)(1-\beta)m\Big)\Gamma^2
\end{aligned}
\tag{51}
$$

Using the fact step-size $\eta_k=\frac{c}{m^\nu T}$ for some $\nu\geq 3$ and the bound of Γ as described in (21), we have λ to be upper bounded by $\mathcal{O}(\frac{1}{m^\nu})$ Now we have,

$$
\begin{aligned}
&f(\mathbf{x}_{k+1})-f(\mathbf{x}_k) &(52)\\
\leq&\left(-\frac{\gamma^2 M\eta_k}{4(1-\beta)m}+\frac{L_2\eta_k^3}{6(1-\beta)m}(1+\sqrt{1-\delta})^3\right)\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^3+\frac{\eta_k}{2(1-\beta)m}\left(\gamma-\frac{\eta_k}{(1-\beta)m}\right)\sum_{i\in M}\frac{M\gamma}{2}\|\mathbf{s}_{i,k+1}\|^3+\lambda\\
=&\left(-\frac{\gamma M\eta_k^2}{4(1-\beta)^2m^2}+\frac{L_2\eta_k^3}{6(1-\beta)m}(1+\sqrt{1-\delta})^3\right)\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^3+\lambda\\
=&-\lambda_{comp}\frac{1}{(1-\alpha)m}\sum_{i\in M}\|\eta_k\mathbf{s}_{i,k+1}\|^3+\lambda &(53)
\end{aligned}
$$

where

$$\lambda_{comp}=[\frac{\gamma M}{4(1-\beta)^2\eta_k m^2}-\frac{L_2}{6(1-\beta)m}(1+\sqrt{1-\delta})^3](1-\alpha)m$$

To ensure $\lambda_{comp}>0$, we need

$$M>\frac{4\eta_k m(1-\beta)}{\gamma}\frac{L_2}{6}(1+\sqrt{1-\delta})^3 \tag{54}$$

Now for the choice of $\eta_k=\frac{c}{Tm^\nu}$ and $\gamma=\frac{c_1}{Tm^\nu}$ for some constant $c_1>0$. We have $M=\mathcal{O}(m(1-\beta)(1+\sqrt{1-\delta})^3)$. Thus we have $\lambda_{comp}=\mathcal{O}(1)$ and $\lambda\ =\mathcal{O}(\frac{1}{m^\nu})$. Now we have

$$\frac{1}{(1-\alpha)m}\sum_{i\in M}\|\eta_k\mathbf{s}_{i,k+1}\|^3\leq\frac{1}{\lambda_{comp}}[f(\mathbf{x}_k)-f(\mathbf{x}_{k+1})+\lambda\ ]$$

At any iteration k, we have

$$\begin{aligned}
\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^3 &= \|\eta_k\, Q(s_{k+1})\|^3 \\
&\leq \frac{1}{(1-\beta)m}\sum_{i\in U}\|\eta_{k_0} Q(\mathbf{s}_{i,k+1})\|^3 \\
&\leq \frac{(1+\sqrt{1-\delta})^3}{(1-\beta)m}\sum_{i\in U}\|\eta_k \mathbf{s}_{i,k+1}\|^3 \\
&= \frac{(1+\sqrt{1-\delta})^3}{(1-\beta)m}\left[\sum_{i\in M}\|\eta_k \mathbf{s}_{i,k+1}\|^3 - \sum_{i\in M\setminus T}\|\eta_k \mathbf{s}_{i,k+1}\|^3 + \sum_{i\in U\setminus B}\|\eta_k \mathbf{s}_{i,k+1}\|^3\right] \\
&\leq \frac{(1+\sqrt{1-\delta})^3}{(1-\beta)m}\left[\sum_{i\in M}\|\eta_k \mathbf{s}_{i,k+1}\|^3 + \alpha m\eta_k^3\Gamma^3\right]
\end{aligned}$$

Now we consider the step k_0, where $k_0 = \operatorname{argmin}_{0\ \leq k\ \leq T-1}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|$.

$$\begin{aligned}
&\min_{0\ \leq k\ \leq T-1}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^3 \\
&\leq \min_{0\ \leq k\ \leq T-1}\frac{(1+\sqrt{1-\delta})^3}{(1-\beta)m}\left[\sum_{i\in M}\|\eta_k \mathbf{s}_{i,k+1}\|^3 + \alpha m\eta_k^3\Gamma^3\right] \\
&\leq \frac{1}{T}\sum_{k=0}^{T-1}(1+\sqrt{1-\delta})^3\frac{(1-\alpha)}{(1-\beta)}\left[\frac{1}{(1-\alpha)m}\sum_{i\in M}\|\eta_k \mathbf{s}_{i,k+1}\|^3 + \frac{\alpha}{1-\alpha}\eta_{k_0}^3\Gamma^3\right] \\
&\leq \frac{1}{T}\sum_{k=0}^{T-1}(1+\sqrt{1-\delta})^3\frac{(1-\alpha)}{(1-\beta)}\left[\frac{f(\mathbf{x}_k)-f(\mathbf{x}_{k+1})}{\lambda_{comp}} + \frac{\lambda}{\lambda_{comp}} + \frac{\alpha}{1-\alpha}\frac{\eta_{k_0}^3\Gamma^3}{\lambda_{comp}}\right] \\
&\leq \frac{1}{T}(1+\sqrt{1-\delta})^3\frac{(1-\alpha)}{(1-\beta)}\left[\frac{f(\mathbf{x}_0)-f}{\lambda_{comp}} + \sum_{k=0}^{T-1}\frac{\lambda}{\lambda_{comp}} + \sum_{k=0}^{T-1}\frac{\alpha}{1-\alpha}\frac{\eta_{k_0}^3\Gamma^3}{\lambda_{comp}}\right] \\
&\leq (1+\sqrt{1-\delta})^3\frac{(1-\alpha)}{(1-\beta)}\left[\frac{f(\mathbf{x}_0)-f}{T\lambda_{comp}} + \frac{\lambda}{\lambda_{comp}} + \frac{\alpha}{1-\alpha}\frac{\eta_{k_0}^3\Gamma^3}{\lambda_{comp}}\right]
\end{aligned}$$

With the choice of η_k, γ we have the terms $\frac{\lambda_\Gamma}{\lambda_{comp}}$ and $\frac{\alpha}{1-\alpha}\frac{\eta_{k_0}^3}{\lambda_{comp}}^3$ are upper bounded by $\mathcal{O}(\frac{1}{m^\nu})$ and higher order of $\mathcal{O}(\frac{1}{m^\nu})$.

We have

$$\begin{aligned}
&\frac{1}{(1-\beta)m}\left[\sum_{i\in M}\|\eta_{k_0}\mathbf{s}_{i,k_0+1}\|^3 + \alpha m\eta_{k_0}^3\Gamma^3\right] \leq \frac{(1-\alpha)}{(1-\beta)}\left[\frac{f(\mathbf{x}_0)-f}{T\lambda_{comp}} + \frac{\lambda}{\lambda_{comp}} + \frac{\alpha}{1-\alpha}\frac{\eta_{k_0}^3\Gamma^3}{\lambda_{comp}}\right] \\
\Rightarrow &\frac{1}{(1-\alpha)m}\left[\sum_{i\in M}\|\eta_{k_0}\mathbf{s}_{i,k_0+1}\|^3 + \alpha m\eta_{k_0}^3\Gamma^3\right] \leq \left[\frac{f(\mathbf{x}_0)-f}{T\lambda_{comp}} + \frac{\lambda}{\lambda_{comp}} + \frac{\alpha}{1-\alpha}\frac{\eta_{k_0}^3\Gamma^3}{\lambda_{comp}}\right] \\
&\Rightarrow \frac{1}{(1-\alpha)m}\sum_{i\in M}\|\eta_{k_0}\mathbf{s}_{i,k_0+1}\|^3 \leq \left[\frac{f(\mathbf{x}_0)-f}{T\lambda_{comp}}\right] = \frac{\psi_{comp}}{T} + C
\end{aligned}$$

where $\psi_{comp} = \frac{f(\mathbf{x}_0)\ \ f}{\lambda_{comp}}$ where C is $\mathcal{O}(1/m)$.

So, we have the term ψ_{comp} is of the order $\mathcal{O}(1)$.

The gradient condition is (using (13))

$$
\begin{aligned}
&\|\nabla f(\mathbf{x}_{k+1})\| \\
=&\left\|\nabla f(\mathbf{x}_{k+1})-\frac{1}{|\mathcal{M}|}\sum_{i2\,M}\mathbf{g}_{i,k}-\frac{1}{|\mathcal{M}|}\sum_{i2\,M}\gamma\mathbf{H}_{i,k+1}\mathbf{s}_{i,k+1}-\frac{1}{|\mathcal{M}|}\sum_{i2\,M}\frac{M\gamma^2}{2}\|\mathbf{s}_{i,k+1}\|\mathbf{s}_{i,k+1}\right\| \\
\leq&\left\|\nabla f(\mathbf{x}_{k+1})-\nabla f(\mathbf{x}_k)-\nabla^2 f(\mathbf{x}_k)(x_{k+1}-x_k)\right\|+\left\|\frac{1}{|\mathcal{M}|}\sum_{i2\,M}(\mathbf{g}_{i,k}-\nabla f(\mathbf{x}_k))\right\| \\
&\quad+\left\|\nabla^2 f(\mathbf{x}_k)(x_{k+1}-x_k)-\gamma\frac{1}{|\mathcal{M}|}\sum_{i2\,M}\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\right\|+\left\|\frac{1}{|\mathcal{M}|}\sum_{i2\,M}\frac{M\gamma^2}{2}\|\mathbf{s}_{i,k+1}\|\mathbf{s}_{i,k+1}\right\| \\
\leq&\frac{L_2\eta_k^2}{2}\left\|\frac{1}{|\mathcal{U}|}\sum_{i2U}Q(\mathbf{s}_{i,k+1})\right\|^2+\epsilon_g+\frac{M\gamma^2}{2}\frac{1}{|\mathcal{M}|}\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\|^2+\left\|\frac{\eta_k}{|\mathcal{U}|}\sum_{i2U}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})-\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\right\|
\end{aligned}
\tag{55}
$$

Now consider the term in (55)

$$
\begin{aligned}
&\left\|\frac{\eta_k}{|\mathcal{U}|}\sum_{i2U}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})-\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\right\| \\
\leq&\left\|\frac{\eta_k}{|\mathcal{U}|}\left[\sum_{i2\,M}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})-\sum_{i2\,M\setminus T}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})+\sum_{i2B\setminus U}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})\right]-\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\right\| \\
\leq&\left\|\frac{\eta_k}{|\mathcal{U}|}\sum_{i2\,M}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})-\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\right\|+\frac{\eta_k}{|\mathcal{U}|}\sum_{i2\,M\setminus T}\|\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})\| \\
&+\frac{\eta_k}{|\mathcal{U}|}\sum_{i2B\setminus U}\|\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})\| \\
\leq&\left\|\frac{\eta_k}{|\mathcal{U}|}\sum_{i2\,M}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})-\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})\right\| \\
&+\left\|\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\nabla^2 f(\mathbf{x}_k)Q(\mathbf{s}_{i,k+1})-\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\nabla^2 f(\mathbf{x}_k)\mathbf{s}_{i,k+1}\right\| \\
&+\left\|\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\nabla^2 f(\mathbf{x}_k)\mathbf{s}_{i,k+1}-\frac{\gamma}{|\mathcal{M}|}\sum_{i2\,M}\mathbf{H}_{i,k}\mathbf{s}_{i,k+1}\right\|+\frac{\eta_k}{(1-\beta)m}(1+\sqrt{1-\delta})L_1[\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\|+\alpha m\Gamma] \\
\leq&(\frac{\eta_k}{(1-\beta)m}-\frac{\gamma}{(1-\alpha)m})L_1(1+\sqrt{1-\delta})\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\|+\frac{\gamma}{(1-\alpha)m}L_1\sqrt{1-\delta}\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\| \\
&+\frac{\gamma\epsilon_H}{(1-\alpha)m}\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\|+\frac{\eta_k}{(1-\beta)m}(1+\sqrt{1-\delta})L_1[\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\|+\alpha m\Gamma] \\
&\leq\left(\frac{\eta_k}{(1-\beta)m}L_1(1+\sqrt{1-\delta})(2+\alpha m)-\frac{\gamma L_1}{(1-\alpha)m}\right)\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\|+\frac{\gamma\epsilon_H}{(1-\alpha)m}\sum_{i2\,M}\|\mathbf{s}_{i,k+1}\| \\
&=\left(\frac{(1-\alpha)}{(1-\beta)}2L_1(1+\sqrt{1-\delta})-\frac{\gamma}{\eta_k}(L_1-\epsilon_H)\right)\frac{1}{(1-\alpha)m}\sum_{i2\,M}\|\eta_k\mathbf{s}_{i,k+1}\|+\frac{\eta_k\alpha}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma
\end{aligned}
$$

Next we consider the term

$$
\begin{aligned}
&\frac{L_2\eta_k^2}{2}\left\|\frac{1}{|\mathcal{U}|}\sum_{i\in U}Q(\mathbf{s}_{i,k+1})\right\|^2\\
\leq&\frac{L_2(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)m}\sum_{i\in U}\|\mathbf{s}_{i,k+1}\|^2\\
\leq&\frac{L_2(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)m}\left[\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^2+\sum_{i\in U\setminus B}\|\mathbf{s}_{i,k+1}\|^2\right]\\
=&\frac{L_2(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)m}\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^2+\frac{L_2\alpha(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)}\Gamma^2
\end{aligned}
\tag{56}
$$

So finally we have

$$
\begin{aligned}
&\|\nabla f(\mathbf{x}_{k+1})\|\\
\leq&\frac{L_2(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)m}\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^2+\epsilon_g+\frac{M\gamma^2}{2(1-\alpha)m}\sum_{i\in M}\|\mathbf{s}_{i,k+1}\|^2\\
&+\left(\frac{(1-\alpha)}{(1-\beta)}2L_1(1+\sqrt{1-\delta})-\frac{\gamma}{\eta_k}(L_1-\epsilon_H)\right)\frac{1}{(1-\alpha)m}\sum_{i\in M}\|\eta_k\mathbf{s}_{i,k+1}\|\\
&+\frac{L_2\alpha(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)}\Gamma^2+\frac{\eta_k\alpha}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma
\end{aligned}
$$

Now we choose $\gamma>\frac{(1-\alpha)}{(1-\beta)}2L_1(1+\sqrt{1-\delta})\frac{\eta_k}{L_1-\epsilon_H}$.

$$
\begin{aligned}
&\|\nabla f(\mathbf{x}_{k+1})\|\\
\leq&\left[\frac{L_2(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)}+\frac{M\gamma^2}{2\eta_k^2}\right]\frac{1}{(1-\alpha)m}\sum_{i\in M}\|\eta_k\mathbf{s}_{i,k+1}\|^2\\
&+\frac{L_2\alpha(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)}\Gamma^2+\frac{\eta_k\alpha}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma+\epsilon_g\\
\leq&\left[\frac{L_2(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)}+\frac{M\gamma^2}{2\eta_k^2}\right]\left[\frac{1}{(1-\alpha)m}\sum_{i\in M}\|\eta_k\mathbf{s}_{i,k+1}\|^3\right]^{2/3}\\
&+\frac{L_2\alpha(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)}\Gamma^2+\frac{\eta_k\alpha}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma+\epsilon_g
\end{aligned}
$$

At step $k=k_0$,

$$
\begin{aligned}
&\|\nabla f(\mathbf{x}_{k_0+1})\|\\
&\leq\left[\frac{L_2(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)}+\frac{M\gamma^2}{2\eta_k^2}\right]\left(\frac{\psi_{comp}}{T}+C\right)^{2/3}+\epsilon_g+\frac{L_2\alpha(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)}\Gamma^2+\frac{\eta_k\alpha}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma\\
&\leq\left[\frac{L_2(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)}+\frac{M\gamma^2}{2\eta_k^2}\right]\left(\frac{\psi_{comp}}{T}\right)^{2/3}+\epsilon_g\\
&+\left[\frac{L_2(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)}+\frac{M\gamma^2}{2\eta_k^2}\right]C^{2/3}+\frac{L_2\alpha(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)}\Gamma^2+\frac{\eta_k\alpha}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma\\
&\leq\frac{\chi_1}{T^{2/3}}+\epsilon_g+\chi_G
\end{aligned}
$$

where $\chi_1 = \left[\frac{L_2(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)} + \frac{M\gamma^2}{2\eta_k^2}\right](\psi_{comp})^{2/3}$. And as $C = \mathcal{O}(\frac{1}{m^\nu})$, we have $\chi_G = \mathcal{O}(\frac{1}{m^{2\nu/3-1}}) + \mathcal{O}(\frac{\alpha}{m^\nu})$. As $\nu \geq 3$, χ_G is always decreasing with m.

The Hessian bound is

$$
\begin{aligned}
&\lambda_{\min}(\nabla^2 f(\mathbf{x}_{k+1})) \\
&= \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \lambda_{\min}\left[\nabla^2 f(\mathbf{x}_{k+1})\right] \\
&= \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \lambda_{\min}\left[\mathbf{H}_{i,k} - (\mathbf{H}_{i,k} - \nabla^2 f(\mathbf{x}_{k+1}))\right] \\
&\geq \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \left[\lambda_{\min}(\mathbf{H}_{i,k}) - \|\mathbf{H}_{i,k} - \nabla^2 f(\mathbf{x}_{k+1})\|\right] \\
&\geq \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \lambda_{\min}(\mathbf{H}_{i,k}) - \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \|\mathbf{H}_{i,k} - \nabla^2 f(\mathbf{x}_{k+1})\| \\
&\geq \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} -\frac{M\gamma}{2}\|\mathbf{s}_{i,k+1}\| - \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \|\mathbf{H}_{i,k} - \nabla^2 f(\mathbf{x}_k)\| \\
&\qquad - \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \|\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_{k+1})\| \\
&\geq \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} -\frac{M\gamma}{2\eta_k}\|\eta_k \mathbf{s}_{i,k+1}\| - \epsilon_H - \frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} L_2\|\mathbf{x}_k - \mathbf{x}_{k+1}\| \\
&\geq -\frac{M\gamma}{2\eta_k}\left[\frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \|\eta_k \mathbf{s}_{i,k+1}\|^3\right]^{1/3} - L_2\|\mathbf{x}_k - \mathbf{x}_{k+1}\| - \epsilon_H \\
&\geq -\frac{M\gamma}{2\eta_k}\left[\frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \|\eta_k \mathbf{s}_{i,k+1}\|^3\right]^{1/3} - L_2\left[\frac{(1+\sqrt{1-\delta})}{(1-\beta)m}\sum_{i\in\mathcal{U}}\|\eta_k \mathbf{s}_{i,k+1}\|\right] - \epsilon_H \\
&\geq -\frac{M\gamma}{2\eta_k}\left[\frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \|\eta_k \mathbf{s}_{i,k+1}\|^3\right]^{1/3} - L_2\frac{(1+\sqrt{1-\delta})}{(1-\beta)m}\left[\sum_{i\in\mathcal{M}}\|\eta_k \mathbf{s}_{i,k+1}\| + \sum_{i\in\mathcal{B}\setminus\mathcal{U}}\|\eta_k \mathbf{s}_{i,k+1}\|\right] - \epsilon_H \\
&\geq -\frac{M\gamma}{2\eta_k}\left[\frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}} \|\eta_k \mathbf{s}_{i,k+1}\|^3\right]^{1/3} - L_2\frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)}\left[\frac{1}{(1-\alpha)m}\sum_{i\in\mathcal{M}}\|\eta_k \mathbf{s}_{i,k+1}\|\right] - \epsilon_H \\
&- L_2\frac{(1+\sqrt{1-\delta})\alpha}{(1-\beta)}\eta_k\Gamma
\end{aligned} \tag{57}
$$

At $k = k_0$ we have

$$
\begin{aligned}
&\lambda_{\min}(\nabla^2 f(\mathbf{x}_{k_0+1})) \qquad (58)\\
&\geq -\frac{M\gamma}{2\eta_k}\left[\frac{\psi_{comp}}{T} + C\right]^{1/3} - L_2\frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)}\left[\frac{\psi_{comp}}{T} + C\right]^{1/3} - \epsilon_H - L_2\frac{(1+\sqrt{1-\delta})\alpha}{(1-\beta)}\eta_k\Gamma \\
&\geq -\left[\frac{M\gamma}{2\eta_k} + L_2\frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)}\right]\psi_{comp}^{1/3}\left(\frac{1}{T}\right)^{1/3} - \epsilon_H - \left(\frac{M\gamma}{2\eta_k}C^{1/3} + L_2\frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)}C^{1/3}\right) \\
&\quad - L_2\frac{(1+\sqrt{1-\delta})\alpha}{(1-\beta)}\eta_k\Gamma \\
&\geq -\frac{\chi_2}{T^{1/3}} - \epsilon_H - \chi_H \qquad (59)
\end{aligned}
$$

where $\chi_2 = \left[\frac{M\gamma}{2\eta_k} + L_2\frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)}\right]\psi_{comp}^{1/3}$. And , we have $\chi_H = \mathcal{O}(\frac{1}{m^{\nu/3-1}}) + \mathcal{O}(\frac{1}{m^\nu})$. As $\nu \geq 3$, we χ_H to be strictly decreasing with m.

Finally, we restate the Theorem 3.5,

Theorem 3.5 (Convergence of FED-CURE). *Suppose $0 \leq \alpha < \beta \leq \frac{1}{2}$. Furthermore, we choose the problem parameters, $M = \mathcal{O}(m(1-\beta)(1+\sqrt{1-\delta})^3)$, and $\eta = \gamma = \frac{c}{Tm^\nu}$ for some constant $c > 0, \nu > 3$. Then, after T iterations of FED-CURE (Algorithm 1), the sequence $\{\mathbf{x}_i\}_{i=1}^T$ generated contains a point $\tilde{x}$ such that*

$$\|\nabla f(\tilde{x})\| \leq \frac{\chi_1}{T^{2/3}} + \epsilon_g + \chi_G, \quad \lambda_{\min}\left(\nabla^2 f(\tilde{x})\right) \geq -\frac{\chi_2}{T^{\frac{1}{3}}} - \epsilon_H - \chi_H, \quad \textit{where,}$$

$$\chi_1 = \left[\frac{L_2(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)} + \frac{M\gamma^2}{2\eta_k^2}\right](\psi_{comp})^{2/3}$$

$$\chi_2 = \left[\frac{M\gamma}{2\eta_k} + L_2\frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)}\right]\psi_{comp}^{1/3}$$

$$\chi_G = \left[\frac{L_2(1-\alpha)(1+\sqrt{1-\delta})^2}{2(1-\beta)} + \frac{M\gamma^2}{2\eta_k^2}\right]C^{2/3} + \frac{L_2\alpha(1+\sqrt{1-\delta})^2\eta_k^2}{2(1-\beta)}\Gamma^2 + \frac{\eta_k\alpha}{(1-\beta)}(1+\sqrt{1-\delta})\Gamma$$

$$\chi_H = \left(\frac{M\gamma}{2\eta_k} + L_2\frac{(1+\sqrt{1-\delta})(1-\alpha)}{(1-\beta)}\right)C^{1/3} + L_2\frac{(1+\sqrt{1-\delta})\alpha}{(1-\beta)}\eta_k\Gamma$$

$$\psi_{comp} = \frac{f(\mathbf{x}_0) - f}{\lambda_{comp}} \textit{ and } C = \frac{\lambda}{\lambda_{comp}}$$

$$\lambda_{comp} = [\frac{\gamma M}{4(1-\beta)\eta_k m^2} - \frac{L_2}{6(1-\beta)m}(1+\sqrt{1-\delta})^3](1-\alpha)m$$

$$\lambda = \left(\frac{\eta_k(1-\alpha)}{(1-\beta)}(L\sqrt{1-\delta}+\epsilon_g) + \frac{\eta_k(\alpha+\beta)L}{(1-\beta)}(1+\sqrt{1-\delta})\right)\Gamma + \frac{L_2\eta_k^3}{6(1-\beta)}(1+\sqrt{1-\delta})^3\alpha\Gamma^3$$
$$+ \frac{\eta_k^2}{2(1-\beta)^2 m}\Big((1-\alpha)(\epsilon_H + 2L_1\sqrt{1-\delta}) + (\alpha+\beta)L_1(1+\sqrt{1-\delta})^2) + L_1(1+\sqrt{1-\delta})^2((1-\beta)m-1)(1-\beta)m\Big)\Gamma^2.$$

For the choice of $\eta = \frac{c}{Tm^\nu}$ and $\gamma = \frac{c}{Tm^\nu}$ and $M = \mathcal{O}(m(1-\beta)(1+\sqrt{1-\delta})^3)$, we have $\lambda = \mathcal{O}(\frac{1}{m^\nu})$ and λ_{comp} to be $\mathcal{O}(1)$.

E.3. Proof of Corollary 3.6

In this Corollary statement we consider centralized ($m = 1$), uncompressed ($\delta = 1$) and non-Byzantine setup ($\alpha = \beta = 0$). With these parameters, we have the value of λ from equation (51) to be 0. Consequently, we have $C = 0$. With $\gamma = 1$, we have

$$\lambda_{comp} = \frac{M}{4\eta_k} - \frac{L_2}{6}$$

So in order for $\lambda_{comp} > 0$, for constant step-size ($\eta_k = 1$), we need $M > \frac{2L_2}{3}$. With $C = 0, \alpha = 0$, we have $\chi_G = \chi_H = 0$. Moreover we have $\chi_1 = \left[\frac{L_2+M}{2}\right](\psi_{comp})^{2/3}$ and $\chi_2 = \left[\frac{2M+L_2}{2}\right](\psi_{comp})^{1/3}$. As it is a centralized setup, there are no gradient and Hessian dissimilarities $\epsilon_g = \epsilon_H = 0$. So we have

$$\|\nabla f(\tilde{x})\| \leq \left[\frac{L_2+M}{2}\right](\psi_{comp})^{2/3}\frac{1}{T^{2/3}}, \quad \lambda_{\min}\left(\nabla^2 f(\tilde{x})\right) \geq -\left[\frac{2M+L_2}{2}\right](\psi_{comp})^{1/3}\frac{1}{T^{1/3}},$$

where $M > \frac{2L_2}{3}$. Thus, the convergence rate of FED-CURE reduces to that of (Nesterov & Polyak, 2006).